

DHML

HTML CSS JavaScript DOM

**referência e dicionário
por ARIANE STOLFI**

Trabalho Final de Graduação
Faculdade de Arquitetura e Urbanismo
da Universidade de São Paulo FAU-USP

Professor orientador
Vicente Gil Filho

São Paulo, julho de 2006

Editado a partir de documentação da W3Schools, Microsoft MSDN, Mozilla Developer Center, W3COM e Wikipedia. Todos direitos (e esquadros) reservados (ou revertidos). Trabalho acadêmico, uso comercial proibido.

As informações providas são apenas para aprendizado e o conteúdo não pode ser garantido. Qualquer erro decorrente da utilização deste guia é de responsabilidade do usuário

composto em Helvetica Neue 45, 47, 65 e 67
capa em tipografia sobre tergal

aos sites que produziremos

Índice

5	Sobre este guia
6	Sobre o projeto gráfico
7	Introdução
9	Hypertext Markup Language
14	Elementos estruturais
15	General reference
16	Color names
17	Entities (special caracteres)
21	HTML Tags
23	Tags by Purpose
25	CSS Reference
33	HTML DOM Objects
34	Core Attributes
35	Events
36	JavaScript Referenc
40	Statements, reserved words
41	Operatos
42	Dictionary Index
43	Convenções sintáticas
44	Dicionário
379	Bibliografia
380	Sites consultados

Sobre este guia

A função deste livro não é ensinar HTML dinâmico. Para isso é melhor procurar um tutorial. O site w3schools.com oferece alguns bons tutoriais que podem ser feitos por quem quiser aprender.

Esse guia contém todos os vocábulos pertencentes às linguagens HTML, CSS, JavaScript, JScript e ao DOM. Foi organizado para ser uma referência rápida, tanto para quem já sabe programar e procura se lembrar da forma correta de determinada palavra, quanto para quem procura descobrir o significado de certa palavra em algum código. Com este guia, é possível compreender códigos alheios e dissecá-los para utilizar somente aquilo que se deseja.

O guia é dividido em duas partes: A primeira contém uma referência geral separada por linguagem e tipo, bem como referência de cores, entidades e tabelas de conversão. A segunda é um dicionário, organizado alfabeticamente, contendo todos os métodos, propriedades, objetos, eventos e operadores de todas as linguagens-alvo (tokens fixos).

Nesta segunda parte, cada palavra está seguida de uma breve descrição, seguida de exemplo e/ou sintaxe de utilização e valores possíveis para os parâmetros requeridos. No caso dos objetos, também estão listadas todas as propriedades e atributos padrão e de estilos aplicáveis, as coleções, objetos, behaviors (comportamentos) e métodos aplicáveis a eles. No caso dos métodos, propriedades, eventos e operadores, estão listados os objetos aos quais eles se aplicam.

Assim, a forma de consulta é, ou diretamente por verbete, em ordem alfabética, ou, quando não se sabe ao certo o que buscar, primeiro no guia geral e depois no verbete relacionado.

Metodologia de Trabalho

Este trabalho é o desenvolvimento de um trabalho anterior: O HTML and CSS reference guide, que posteriormente foi traduzido. Esse trabalho consistia em uma compilação da referência do site w3schools.com.

Para a elaboração do guia de referência de HTML dinâmico, foi feita uma pesquisa no Microsoft developers center, que contém referência separada por tipo de elemento. Foi compilada a informação mais relevante de todos itens, como descrição, sintaxe e/ou evento. Itens que não tinham padrão definido em algum lugar ou extensões propostas pela microsoft foram marcados com um [ie]. Posteriormente, foi adicionada informação relativas às tags retirada do [w3schools](http://w3schools.com) de onde também foram compiladas as informações sobre os métodos da linguagem JavaScript—sinalizados com um [JavaScript]. Informações sobre operandos e declarações (statements) foram compilados do mozilla developers center. As informações foram homogenizadas já na fase de compilação. Depois foi feita organização alfabética dos elementos sintáticos e a revisão de padrão. Elementos exclusivos do Internet explorer não foram, em sua maioria, incluídos na referência geral

Finalmente, foram feitas as adequações para o formato de impressão em A4 frente-e-verso.

Por enquanto, este é um trabalho puramente acadêmico e as informações contidas são protegidas por copyright de terceiros. Sendo assim, não pode ser reproduzida com finalidades comerciais.

Sobre o projeto gráfico

Quem já folheou algum livro de programação vai notar que este guia rompeu com uma velha tradição que é a das letras mono-espaçadas na transcrição de linhas de código. O uso desse tipo de fonte acarretaria num aumento considerável da área de impressão, e se há alguma relação entre familiaridade e facilidade de compreensão, ela não é suficiente para compensar os prejuízos práticos a que se expôs a tradição até agora, que produziu grandes trambolhos.

O formato — A4 cortado na vertical — foi escolhido para permitir fácil reprodução por meio de impressão digital ou xérox. A versão final contém uma mesma página repetida duas vezes, que impresso frente e verso gera duas cópias do dicionário. Impresso só as pares ou ímpares gera uma cópia com folhas dobradas. Além disso, o formato fino ocupa pouco espaço próximo ao computador, um espaço que muitas vezes fica entulhado de coisas e normalmente não é muito abundante.

A encadernação wire-o foi escolhida para propiciar abertura total aliada a um sistema semi-industrial (ao contrário da espiral normal que requer um sistema trabalhoso de colocação)

Outro ponto a se notar é que procurou-se não utilizar elementos puramente decorativos. Homenagem à essa escola que forjou o que hoje podemos chamar de um projeto estético político revolucionário — transitório — que não vai ter uma forma final acabada, enquanto houver luta de classes.

As poucas ilustrações espalhadas ao longo do livro além de serem informativas, não num sentido estrito, são um descanso merecido para o leitor e para a autora. Elas dialogam com o estado de barbárie geral e estética da internet e do trabalho de programação/diagramação.

De uma maneira geral, este guia é um contraponto estético ao que em geral se produz nos últimos tempos no mercado. Embora não tenha me concentrado em passar noções de design e composição para os leitores, espero que o próprio projeto gráfico seja de certa forma inspirador.

Introdução

A internet começou a ser explorada pela iniciativa privada no Brasil em 1995. De lá pra cá, passou de uma ferramenta restrita a um dos meios de comunicação mais importantes, e é certamente o meio com maior potencial de crescimento e desenvolvimento.

De repente, a cultura digital começa a já tomar conta de uma geração, que passa dias e dias no mns, orkut, youtube, fotolog, flickr, myspace, wikipedia, google ou em qualquer que seja o fenômeno internetesco momentâneo¹. A maioria das pessoas, no entanto, tem um contato apenas com a forma final, aparente desses fenômenos culturais.

Por trás disso, uma legião de webmasters, designers e programadores mantendo tudo atualizado, funcionando e desenvolvendo meios de manter satisfeitas as vontades das legiões (BÁRBARAS?!) de empresários, consumidores e agências sedentos por interfaces amigas e cartões de crédito.

Mais uma vez na história fica demonstrado que todo desenvolvimento é contraditório. Enquanto uns enxergam no comércio eletrônico a mais nova tentativa de salvar o capitalismo, diversos grupos políticos de ideologias diferentes vêem nessa nova cultura um instrumento potencial para a transformação (radical ou não) da sociedade. É mesmo possível que haja um potencial revolucionário nesse novo modo de comunicação, mas para isso, os agentes dessa transformação possível não poderão ficar passivos em relação às novas tecnologias. É preciso avançar e conhecê-las a fundo.

A forma oculta

Basicamente, cada página da internet é um arquivo separado, que fica hospedado em um servidor, em um determinado endereço IP, que pode ser acessado via um nome, conhecido como URL. O arquivo contém informações que são interpretadas pelos programas conhecidos como navegadores. Esses programas reconhecem os tokens² e transformam uma informação codificada na informação visual, juntando imagens e texto que é apresentada ao internauta.

Neste processo, quem manda é o navegador. Ele que processa, avalia e executa as ordens dadas pelos programadores. Na sociedade capitalista agonizante, cada conglomerado decide segundo seus interesses. Apesar de existirem consórcios como o W3C, que determina os padrões para o HTML, isso não significa necessariamente que as empresas fabricantes de software implementem as linguagens como estabelecido.

Isto na prática quer dizer:

Nem tudo funciona como o previsto.

Basicamente, existem duas implementações diferentes de DHTML: Uma windows/IE, que é semelhante à da netscape (referência on-line: <http://msdn.microsoft.com>) e outra, baseada num sistema chamado Geko, implementada pelo Mozilla, Firefox e Opera (referência on-line: <http://developer.mozilla.org/>). Existem vários sites que expõem problemas de incompatibili-

dade que podem surgir na visualização de estilos em browsers diferentes. Alguns conselhos para se evitar problemas com CSS são: Procurar sempre definir margens — os navegadores têm valores-padrão diferente para margens. Os valores de padding costumam ser interpretados de maneira diferente se usados em conjunto com dimensionamento, existem vários meios de contornar esse problema. Um relativamente simples é usar um contêiner para a dimensão, aonde se aplicam valores de width e height e outro dentro com o conteúdo propriamente dito, aonde se aplicam os valores de padding.

Outra sugestão que fica é procurar validar a sintaxe de seus arquivos. alguns endereços de validadores on-line:

validador de HTML da WDG

<http://www.htmlhelp.com/tools/validator/>

validador de CSS da W3Co

<http://jigsaw.w3.org/css-validator/>

As páginas da internet podem ser feitas de maneiras diferentes, ou diretamente, escrevendo as linhas de código em um programa de edição de texto comum ou especializado. Neste caso, o programador-designer tem que ter um planejamento espacial prévio de posicionamento dos elementos na página.

Outro modo de se fazer uma página é simplesmente copiar o código-fonte de um site que se deseje imitar, alterando os conteúdos desejados. O código fonte da maioria dos sites é acessível a qualquer usuário via um clique com o botão direito do mouse.

Uma terceira maneira de se fazer uma página é utilizando um software especial, como o Dreamweaver, o Fireworks e o goLive, que possuem uma interface gráfica e constroem automaticamente o arquivo HTML a partir do posicionamento dos elementos na página. Arquivos gerados dessa maneira possuem alguns inconvenientes, como o de serem muitas vezes mais pesados ou terem a estrutura organizada de uma forma que não é necessariamente a mais lógica.

Um sintoma colateral muito comum é o excesso de tabelas, que são criadas automaticamente para o posicionamento dos elementos.

Uma tendência que tem crescido nos últimos tempos é Tableless, um princípio de criação que usa o CSS para posicionamento dos elementos, a organização do conteúdo em divs, que podem mudar de posição, cor, tamanho e estilo de uma maneira mais flexível do que as tabelas. Assim, as tabelas são usadas quando o significado semântico pede. Grandes sites, como o youtube, já seguem essa nova lógica de projeto.

Robots

Um passo fundamental no crescimento e difusão do uso da internet foi a criação dos mecanismos de busca automáticos, que indexam e organizam o conteúdo dessa rede dispersa, tornando acessíveis as páginas para aqueles que não sabem exatamente aonde ir. Cada mecanismo diferente (google, yahoo, msn, acoona ...) tem um algoritmo diferente de indexação, mas parece que a organização semântica da página tem influência nos resultados das pesquisas feitas³.

Rôbos reconhecem a estrutura semântico-hierárquica da página, priorizando informação contida em títulos e subtítulos, por exemplo, quando esses estão indicados com <h1> <h2> etc.

1. Alguns desses sites recebem cerca de 250 milhões de visitas por dia (fonte www.alexaholic.com)

2. token é uma sequência de letras que possui um conteúdo semântico (significado) numa determinada linguagem

3. Ver Diego Eis, www.tableless.com

HyperText

Apesar de o termo Hipertexto ter sido cunhado em 1965 (por Ted Nelson), só o desenvolvimento da internet, principalmente ao longo dos anos 90 deu a ele uma significação social e uma utilização massiva.

Trata-se de texto encadeado de maneira não linear, através de links para outros textos por referências-cruzadas, permitindo o usuário navegar entre conteúdos determinados, encadeados de maneira não linear.

Markup Language

Uma markup language, ou linguagem de marcação, combina texto com informação extra, que entre outras coisas pode ser utilizada para estruturar o documento e indicar recursos de apresentação.

O termo deriva da antiga prática de marcar um manuscrito para transmitir instruções aos tipógrafos, como tamanho de fonte, posição, justificação etc. De uma certa maneira, a diagramação via linguagem de programação ainda se assemelha em muitos aspectos com a antiga tipografia em metal. Há uma necessidade de planejamento antecipado espacial de dimensões concretas como a altura das linhas, tamanhos das fontes, posicionamento de elementos em relação a outros no espaço da página, sem contar uma relativa constrição em relação à quantidade de fontes disponíveis.

Principais exemplos

TeX

Criada nos anos 70 por Donald Knuth, a linguagem permite uma diagramação de qualidade para fórmulas matemáticas e em conjunto com um sistema de preparação de documentos chamado LaTeX é ainda hoje o padrão mais utilizado para publicações acadêmicas em áreas como matemática, física e engenharia. A metafont, desenvolvida pelo matemático para ser utilizada em conjunto com a linguagem, é uma fonte baseada em curvas bézier, que pode ser parametrizável, e tem uma certa mobilidade de caracteres, ajustáveis a fórmulas, por exemplo.

AMS Euler

$$\int_0^3 9x^2 + 2x + 4 \, dx = 3x^3 + x^2 + 4x + C$$
$$e^{x+iy} = e^x (\cos y + i \sin y)$$
$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
From Wikipedia, Mathematical text typeset using TeX and the AMS Euler font
the source code for this is:
\input gkpmac

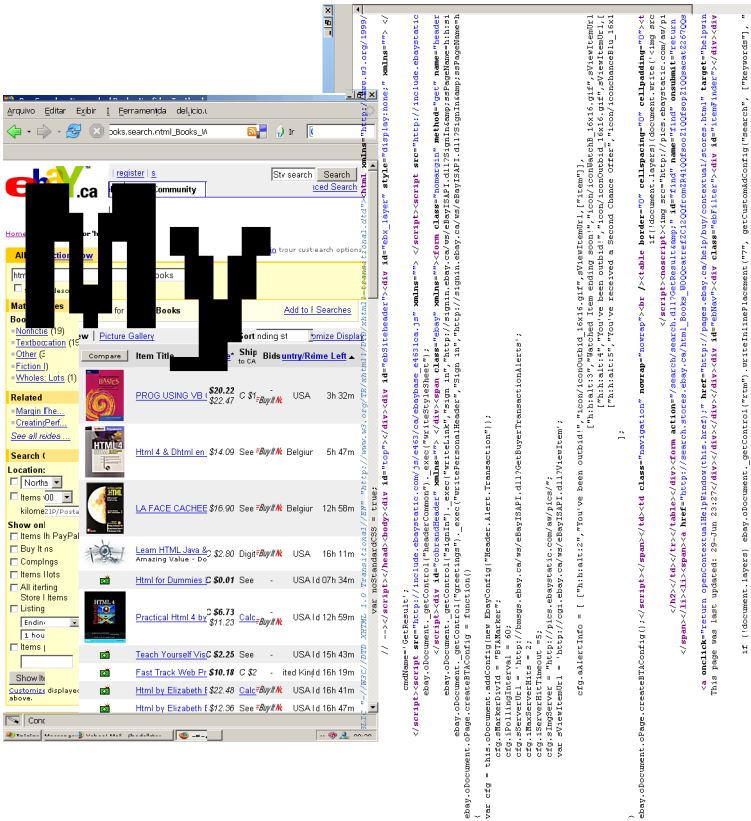
$$AMS\ Euler$$
$$\int_0^3 9x^2 + 2x + 4\, dx = 3x^3 + x^2 + 4x + C$$
$$e^{x+iy} = e^x(\cos(y) + i\sin(y))$$
$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
```

SGML

O SGML (Standard Generalized Markup Language), foi desenvolvido nos anos 80 a partir da idéia de que a marcação — feita a partir de tags definidas entre um < (menor que) e > (maior que) — deveria se concentrar em aspectos estruturais do documento, deixando a apresentação visual para o interpretante. Foi desenhado para permitir a leitura de documentos por máquinas em grandes projetos governamentais.

HTML

O HyperText Markup Language foi desenvolvido em 1991 a partir da sintaxe do SGML e utiliza alguns tags específicos (listados adiante) para definir a estrutura semântica e de apresentação de um texto. Assim, através de certas marcas, é possível definir o comportamento de um certo trecho de texto, a inserção de imagens, botões, formulários e scripts, a relação entre dois documentos, cabeçalhos, citações, divisões e tabelas entre outros. Essas marcas, chamadas tags vêm normalmente em pares como <div> e </div> uma marcando o início e outra o fim de um determinado elemento ou seção. O HTML aceita que algumas tags não tenham fecho, as ditas tags vazias, como
, que indica uma quebra de linha ou , que indica uma imagem. O HTML foi essencial para o desenvolvimento da Internet, e até hoje é a linguagem mais utilizada na rede, e se padrão de linguagem e sintaxe é determinado pelo World Wide Web Consortium (W3C). A versão atual é 4.01. Embora existam vários programas específicos para a formatação de páginas de internet, a estrutura do documento é um arquivo comum de texto, e pode ser feita em qualquer editor de texto salvando-o como *.html. A vantagem de utilizar um editor específico para HTML é que eles podem conter recursos úteis como a separação do texto e da estrutura por cores, ou reconhecer certos erros. O navegador reconhece as marcas e cuida de dispor as informações segundo o determinado pela marcação.



XML

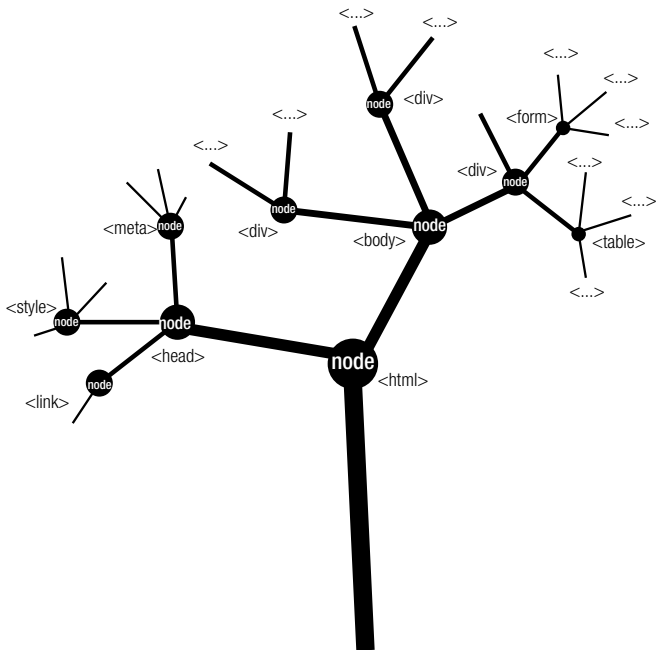
Uma linguagem de marcação que tem ganhado importância recentemente é a eXtensible Markup Language, que tem a sintaxe muito parecida à do HTML, um pouco simplificada e que permite a criação de Tags personalizadas. Pela simplicidade e flexibilidade, tem se tornado uma das linguagens mais importantes para formação de banco de dados e transmissão de dados entre servidores.

XHTML

Pode ser tratado como sucessor do HTML, mas é na verdade uma recomendação diferente do W3C. Combina as tags do HTML com as regras do XML e consegue ser interpretado por qualquer dispositivo, pois suas marcações possuem sentido semântico para as máquinas. Diferente da HTML, a XHTML, assim como o CSS não permite tags vazias, ou seja, toda tag tem de ser fechada adequadamente como por exemplo: `<br></br>`, ou `<br />`. Além disso a estrutura é um pouco mais rígida, e as tags devem ser fechadas na ordem em que foram abertas. Quando existirem diferenças de uso entre, as linguagens, serão salientadas neste guia de referência

Estrutura em árvore

```
<html>
<head>
<title>Titulo da pagina</title>
</head>
<body>
um texto qualquer <b>aqui fica bold</b>
</body>
</html>
```



Os documentos HTML, XHTML ou XML têm em comum uma estrutura de árvore, aonde elementos são colocados em relação uns com os outros de uma maneira encadeada.

A tag `<html>` indica que o documento começou; `</html>` é o fechamento dessa seção e indica que a partir desse ponto o browser não reconhece mais o texto como html.

Dentro desse primeiro nó, são inseridos os demais elementos, que primeiro se dividem entre as tags `<head>`/`</head>`, aonde entram informações sobre o documento, como título, meta-

informação, links entre outras e as tags `<body></body>`, que representam o início e o fim do corpo do arquivo, aquilo que é exibido na janela do navegador. Dentro desse espaço podem ser colocadas diversas subdivisões e objetos.

CSS

Através da definição de estilos, pode-se controlar a apresentação da informação escrita em uma linguagem de marcação. A forma mais utilizada em conjunto com o HTML é o CSS, ou Cascading Style Sheets.

Através dessa linguagem pode-se definir fonte, cor, posição, tamanho, margens, bordas, alinhamento do texto, entrelinhamento, corpo, entre outros atributos.

A informação CSS pode ser usada de três maneiras: inline, dentro de um tag, através do atributo style (ex. `<span style="color:red">`), embebida no documento (através do tag `<style>`), ou através de um arquivo .css linkado ao html. Nos dois últimos casos, utiliza-se a sintaxe específica para o CSS. A grande vantagem da utilização de um arquivo CSS separado é a possibilidade de aplicar a mesma folha de estilos em vários arquivos HTML, e alterá-los facilmente mexendo em apenas um arquivo. Um mesmo arquivo HTML pode ter sua aparência radicalmente modificada através da aplicação de folhas de estilos diferentes.

Através do CSS é possível também especificar estilos diferentes para mídias diversas, como fonte e tamanho diferente para impressão e tela.

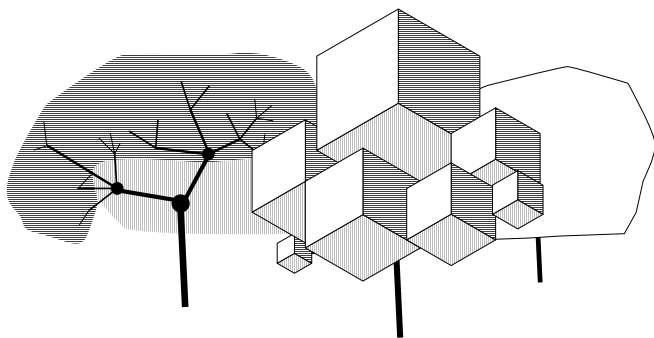
Exemplo de definição de estilo:

definição de classe:

```
.manchete {
border:0px;
text-decoration: none;
font-size: 60px;
font-weight: bold;
line-height: 55px;
color:#000000;
font-family: arial, helvetica, sans-serif;
}
```

definição de id:

```
#corpo {
float:left;
position:absolute;
top:150px;
left:30%;
width:60%;
border-bottom:3px solid #000;
border-top:3px solid #000;
padding:0 0 20px;
}
```



JavaScript

A linguagem foi criada a partir da necessidade de validação de formulários on-line e interação com páginas HTML. É uma linguagem interpretada, de script que tem tipos de variáveis indefinidos, com ótimo suporte a expressões regulares.

O padrão ECMA-262 também conhecido como ECMAScript, é um padrão que também é seguido pela linguagem ActionScript do Flash, da macromedia, e também pelo Jscript, da Microsoft.

JavaScript + CSS + HTML = DHTML

Através do DOM que é um modelo de objeto de documentos, é possível utilizar Javascript para fazer modificações em arquivos HTML e CSS. O DOM é uma interface de programação de aplicativos (API) que permite o acesso à estrutura em raiz de um documento HTML ou XML e a operação de seu conteúdo.

Através do HTML dinâmico é possível, por exemplo, a criação de blogs, a personalização de websites, o movimento de itens e elementos e até a criação de códigos maliciosos por hackers.

AJAX

O último estágio de desenvolvimento da tecnologia de conteúdo dinâmico na web é chamada de AJAX, uma abreviação de Asynchronous JavaScript and XML, que utiliza o XMLHttpRequest para a transmissão de dados entre o servidor e o usuário sem ter que dar reload na página. Isto tem se desenvolvido recentemente e algumas falhas de segurança foram descobertas em alguns sites que funcionam por este método.

Elementos estruturais

objetos

Objetos são entidades que possuem atributos e propriedades, podendo ser manipulados e utilizados. Cada utilização de um determinado objeto é uma instância. Cada tag HTML representa um objeto pelo DOM.

propriedades

Ou atributos, podem ser utilizados para controlar, definir ou observar estados dos objetos.

métodos

Métodos são ações que podem ser feitas pelos ou sobre os objetos. Funções, algoritmos ou procedimentos são métodos. Neste guia, estão sinalizados com () após o nome. Ex: parse()

eventos

São a sinalização de determinada interação do usuário, ou de ação ocorrida devido ao tempo, como por exemplo, a passagem do mouse pelo objeto, ou então o carregamento do arquivo.

variáveis

São objetos que contêm um valor que pode ser adulterado

Valores possíveis

String

Um trecho de texto. Quando o valor de um determinado parâmetro pode ser um string, seu nome pode ser indicado neste guia com um "s" no início, como por exemplo *sColor*.

Integer

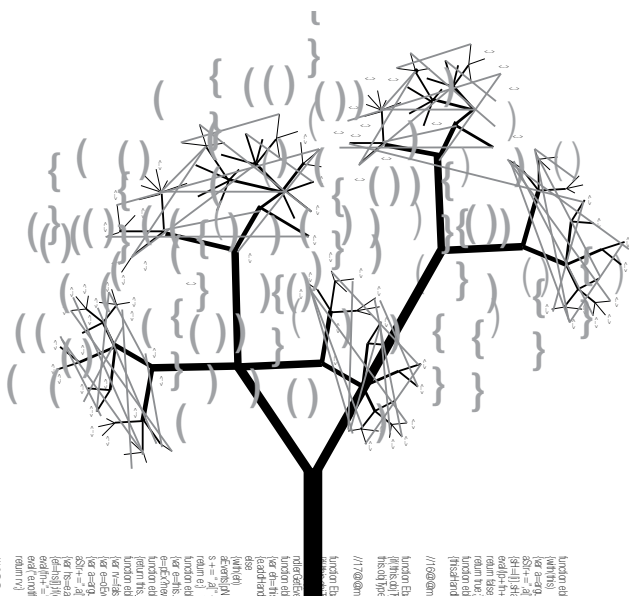
Número inteiro. Quando o valor de um parâmetro deve ser um número inteiro, seu nome pode ser indicado neste guia com um "i" no início como por exemplo *iIndex*

Floating-point number

Número real, pode ter decimais

Boolean

Valor lógico correspondente a verdadeiro e falso. Pode ser 0 e 1 ou true or false. Quando o valor de um parâmetro deve ser um número valor booleano, seu nome pode ser indicado neste guia com um "b" no início como por exemplo *bClear*.



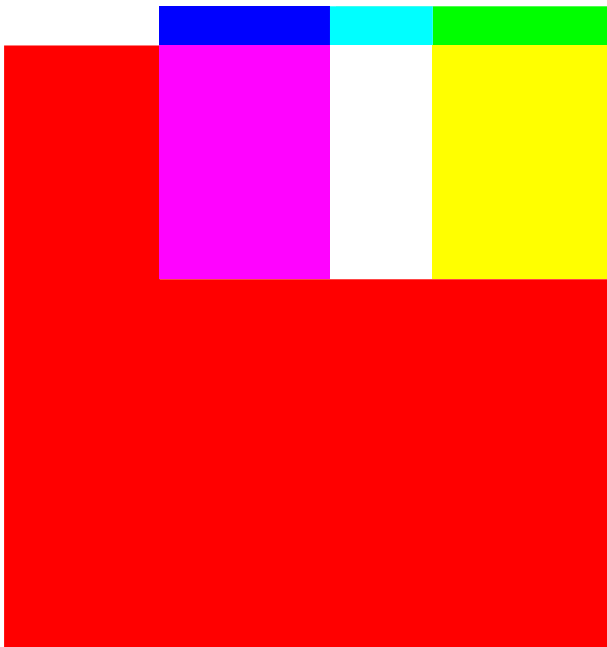
General Reference

[illegible]

Conversion table from Hexadecimal to Decimal

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	000	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015
1	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
2	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047
3	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063
4	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079
5	080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095
6	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

The computer monitors produce color by an additive model, based on the sum of red, green and blue luminocent points (R,G,B). The black is the ausence of color



Color Names

not supported by all browsers

printed values do not correspond exactly with the colors at the screen

Name	HEX	LightGray	#D3D3D3
		LightGreen	#90EE90
		LightPink	#FFB6C1
		LightSalmon	#FFA07A
		LightSeaGreen	#20B2AA
		LightSkyBlue	#87CEFA
		LightSlateBlue	#8470FF
		LightSlateGray	#778899
		LightSteelBlue	#B0C4DE
		LightYellow	#FFFFE0
		Lime	#00FF00
		LimeGreen	#32CD32
		Linen	#FAF0E6
		Magenta	#FF00FF
		Maroon	#800000
		MediumAquaMarine	#66CDAA
		MediumBlue	#0000CD
		MediumOrchid	#BA55D3
		MediumPurple	#9370D8
		MediumSeaGreen	#3CB371
		MediumSlateBlue	#7B68EE
		MediumSpringGreen	#00FA9A
		MediumTurquoise	#48D1CC
		MediumVioletRed	#C71585
		MidnightBlue	#191970
		MintCream	#F5FFFA
		MistyRose	#FFE4E1
		Moccasin	#FFE4B5
		NavajoWhite	#FFDEAD
		Navy	#000080
		OldLace	#FDF5E6
		Olive	#808000
		OliveDrab	#6B8E23
		Orange	#FFA500
		OrangeRed	#FF4500
		Orchid	#DA70D6
		PaleGoldenRod	#EEE8AA
		PaleGreen	#98FB98
		PaleTurquoise	#AFEEEE
		PaleVioletRed	#87CEEB
		PapayaWhip	#FFEFD5
		PeachPuff	#FFDAB9
		Peru	#CD853F
		Pink	#FFC0CB
		Plum	#DDA0DD
		PowderBlue	#B0E0E6
		Purple	#800080
		Red	#FF0000
		RosyBrown	#BC8F8F
		RoyalBlue	#4169E1
		SaddleBrown	#8B4513
		Salmon	#FA8072
		SandyBrown	#F4A460
		SeaGreen	#2E8B57
		SeaShell	#FFF5EE
		Sienna	#A0522D
		Silver	#C0C0C0
		SkyBlue	#87CEEB
		SlateBlue	#6A5ACD
		SlateGray	#708090
		Snow	#FFFAFA
		SpringGreen	#00FF7F
		SteelBlue	#4682B4
		Tan	#D2B48C
		Teal	#008080
		Thistle	#D8BFD8
		Tomato	#FF6347
		Turquoise	#40E0D0
		Violet	#EE82EE
		VioletRed	#D02090
		Wheat	#F5DEB3
		White	#FFFFFF
		WhiteSmoke	#F5F5F5
		Yellow	#FFFF00
		YellowGreen	#9ACD32

7-BIT Printable ASCII Characters

Result	Description	Entity Number			
	space	 	P	uppercase P	P
!	exclamation mark	!	Q	uppercase Q	Q
"	quotation mark	"	R	uppercase R	R
#	number sign	#	S	uppercase S	S
\$	dollar sign	$	T	uppercase T	T
%	percent sign	%	U	uppercase U	U
&	ampersand	&	V	uppercase V	V
'	apostrophe	'	W	uppercase W	W
(	left parenthesis	(	X	uppercase X	X
)	right parenthesis	)	Y	uppercase Y	Y
*	asterisk	*	Z	uppercase Z	Z
+	plus sign	+	[	left square bracket	[
,	comma	,	\	backslash	\
-	hyphen	-	]	right square bracket	]
.	period	.	^	caret	^
/	slash	/	_	underscore	_
0	digit 0	0	`	grave accent	`
1	digit 1	1	a	lowercase a	a
2	digit 2	2	b	lowercase b	b
3	digit 3	3	c	lowercase c	c
4	digit 4	4	d	lowercase d	d
5	digit 5	5	e	lowercase e	e
6	digit 6	6	f	lowercase f	f
7	digit 7	7	g	lowercase g	g
8	digit 8	8	h	lowercase h	h
9	digit 9	9	i	lowercase i	i
:	colon	:	j	lowercase j	j
;	semicolon	;	k	lowercase k	k
<	less-than	<	l	lowercase l	l
=	equals-to	=	m	lowercase m	m
>	greater-than	>	n	lowercase n	n
?	question mark	?	o	lowercase o	o
@	at sign	@	p	lowercase p	p
A	uppercase A	A	q	lowercase q	q
B	uppercase B	B	r	lowercase r	r
C	uppercase C	C	s	lowercase s	s
D	uppercase D	D	t	lowercase t	t
E	uppercase E	E	u	lowercase u	u
F	uppercase F	F	v	lowercase v	v
G	uppercase G	G	w	lowercase w	w
H	uppercase H	H	x	lowercase x	x
I	uppercase I	I	y	lowercase y	y
J	uppercase J	J	z	lowercase z	z
K	uppercase K	K	{	left curly brace	{
L	uppercase L	L		vertical bar	|
M	uppercase M	M	}	right curly brace	}
N	uppercase N	N	~	tilde	~
O	uppercase O	O			

7-BIT ASCII Device Control Characters

designed to control hardware devices like printers and tape drives, have nothing to do inside any HTML document.

Result	Description	Entity Number			
NUL	null character	�	DLE	data link escape	
SOH	start of header		DC1	device control 1	
STX	start of text		DC2	device control 2	
ETX	end of text		DC3	device control 3	
EOT	end of transmission		DC4	device control 4	
ENQ	enquiry		NAK	negative acknowledge	
ACK	acknowledge		SYN	synchronize	
BEL	bell (ring)		ETB	end transmission block	
BS	backspace		CAN	cancel	
HT	horizontal tab			EM	end of medium	
LF	line feed	
	SUB	substitute	
VT	vertical tab		ESC	escape	
FF	form feed		FS	file separator	
CR	carriage return		GS	group separator	
SO	shift out		RS	record separator	
SI	shift in		US	unit separator	
			DEL	delete (rubout)	

ISO 8859-1 Symbol Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
¡	inverted exclamation mark	¡	¡
¢	currency	¢	¢
£	pound	£	£
¥	yen	¥	¥
¦	broken vertical bar	¦	¦
§	section	§	§
¨	spacing diaeresis	¨	¨
©	copyright	©	©
ª	feminine		
	ordinal indicator	ª	ª
«	angle quotation mark (left)	&lquo;	«
¬	negation	¬	¬
–	soft hyphen	­	­
®	registered trademark	®	®
™	trademark	™	™
ˆ	spacing macron	¯	¯
°	degree	°	°
±	plus-or-minus	±	±
²	superscript 2	²	²
³	superscript 3	³	³
´	spacing acute	´	´
µ	micro	µ	µ
¶	paragraph	¶	¶
·	middle dot	·	·
¸	spacing cedilla	¸	¸
¹	superscript 1	¹	¹
º	masculine ordinal indicator	º	º
»	angle quotation mark (right)	&rquo;	»
¼	fraction 1/4	¼	¼
½	fraction 1/2	½	½
¾	fraction 3/4	¾	¾
¿	inverted question mark	¿	¿
×	multiplication	×	×
÷	division	÷	÷

¶ × ¿ ± ®
● ° ¼ §

ISO 8859-1 Character Entities

Some Other Entities supported by HTML

Description	Name	Number
À	À	À
Á	&Acute;	Á
Â	Â	Â
Ã	Ã	Ã
Ä	Ä	Ä
Å	Å	Å
Æ	&Aelig;	Æ
Ç	Ç	Ç
È	È	È
É	É	É
Ê	Ê	Ê
Ë	Ë	Ë
Ì	Ì	Ì
Í	Í	Í
Î	Î	Î
Ï	Ï	Ï
Ð	Ð	Ð
Ñ	Ñ	Ñ
Ò	Ò	Ò
Ó	Ó	Ó
Ô	Ô	Ô
Õ	Õ	Õ
Ö	Ö	Ö
Ø	Ø	Ø
Ù	Ù	Ù
Ú	Ú	Ú
Û	Û	Û
Ü	Ü	Ü
Ý	Ý	Ý
Þ	Þ	Þ
ß	ß	ß
à	à	à
á	á	á
â	â	â
ã	ã	ã
ä	ä	ä
å	å	å
æ	æ	æ
ç	ç	ç
è	è	è
é	é	é
ê	ê	ê
ë	ë	ë
ì	ì	ì
í	í	í
î	î	î
ï	ï	ï
ð	ð	ð
ñ	ñ	ñ
ò	ò	ò
ó	ó	ó
ô	ô	ô
õ	õ	õ
ö	ö	ö
ø	ø	ø
ù	ù	ù
ú	ú	ú
û	û	û
ü	ü	ü
ý	ý	ý
þ	þ	þ
ÿ	ÿ	ÿ

		Name	Number
Œ	capital ligature OE	Œ	Œ
œ	small ligature oe	œ	œ
Š	capital S with caron	Š	Š
š	small S with caron	š	š
Ÿ	capital Y with diaeresis	Ÿ	Ÿ
^	modifier letter circumflex accent	ˆ	ˆ
˜	small tilde	˜	˜
	en space	 	 
	em space	 	 
	thin space	 	 
	zero width non-joiner	‍	‌
†	zero width joiner	‍	‍
↵	left-to-right mark	‎	‎
↶	right-to-left mark	‏	‏
—	en dash	–	–
—	em dash	—	—
‘	left single quotation mark	‘	‘
’	right single quotation mark	’	’
‚	single low-9 quotation mark	‚	‚
“	left double quotation mark	“	“
”	right double quotation mark	”	”
„	double low-9 quotation mark	„	„
†	dagger	†	†
‡	double dagger	‡	‡
...	horizontal ellipsis	…	…
‰	per mille	‰	‰
◀	single left-pointing angle quotation	‹	‹
▶	single right-pointing angle quotation	›	›
€	euro	€	€

[illegible]

Html tags

Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines an anchor
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<applet>	Deprecated. Defines an applet
<area>	Defines an area inside an image map
	Defines bold text
<base>	Defines a base URL for all the links in a page
<basefont>	Deprecated. Defines a base font
<bdo>	Defines the direction of text display
<bgsound>	Defines background sound. IE only
<big>	Defines big text
<blockquote>	Defines a long quotation
<body>	Defines the body element
 	Inserts a single line break
<button>	Defines a push button
<caption>	Defines a table caption
<center>	Deprecated. Defines centered text
<cite>	Defines a citation
<code>	Defines computer code text
<col>	Defines attributes for table columns
<colgroup>	Defines groups of table columns
<custom>	Defines a user-defined element. IE only
<dd>	Defines a definition description
	Defines deleted text
<dir>	Deprecated. Defines a directory list
<div>	Defines a section in a document
<dfn>	Defines a definition term
<dl>	Defines a definition list
<dt>	Defines a definition term
	Defines emphasized text
<embed>	Defines an embed document of any type. IE only
<fieldset>	Defines a fieldset
	Deprecated. Defines text font, size, and color
<form>	Defines a form
<frame>	Defines a sub window (a frame)
<frameset>	Defines a set of frames
<h1> to <h6>	Defines header 1 to header 6
<head>	Defines information about the document
<hr>	Defines a horizontal rule
<html>	Defines an html document
<i>	Defines italic text
<iframe>	Defines an inline sub window (frame)

	Defines an image
<input>	Defines an input field
<ins>	Defines inserted text
<isindex>	Deprecated. Defines a single-line input field
<kbd>	Defines keyboard text
<label>	Defines a label for a form control
<legend>	Defines a title in a fieldset
	Defines a list item
<link>	Defines a resource reference
<listing>	Deprecated. Defines keyboard text. IE only
<map>	Defines an image map
<marquee>	Defines a scrolling text marquee. IE only
<menu>	Deprecated. Defines a menu list
<meta>	Defines meta information
<nobr>	Reders text whith no line break. IE only
<noframes>	Defines a noframe section
<noscript>	Defines a noscript section
<object>	Defines an embedded object
	Defines an ordered list
<optgroup>	Defines an option group
<option>	Defines an option in a drop-down list
<p>	Defines a paragraph
<param>	Defines a parameter for an object
<pre>	Defines preformatted text
<q>	Defines a short quotation
<rt>	Defines the ruby text. IE only
<ruby>	Defines an inline annotation. IE only
<s>	Deprecated. Defines strikethrough text
<samp>	Defines sample computer code
<script>	Defines a script
<select>	Defines a selectable list
<small>	Defines small text
	Defines a section in a document
<strike>	Deprecated. Defines strikethrough text
	Defines strong text
<style>	Defines a style definition
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<table>	Defines a table
<tbody>	Defines a table body
<td>	Defines a table cell
<textarea>	Defines a text area
<tfoot>	Defines a table footer
<th>	Defines a table header
<thead>	Defines a table header
<title>	Defines the document title
<tr>	Defines a table row
<tt>	Defines teletype text
<u>	Deprecated. Defines underlined text
	Defines an unordered list
<var>	Defines a variable
<wbr>	Defines a line break in a nobr section. IE only
<xml>	Defines a Xml data island in a htm page. IE only
<xmp>	Deprecated. Defines preformatted text

tags by purpose

Start tag	Purpose
Basic Tags	
<!DOCTYPE>	Defines the document type
<html>	Defines an html document
<body>	Defines the body element
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!--...-->	Defines a comment
Char Format	
	Defines bold text
	Deprecated. Defines text font, size, and color
<i>	Defines italic text
	Defines emphasized text
<big>	Defines big text
	Defines strong text
<small>	Defines small text
<sup>	Defines superscripted text
<sub>	Defines subscripted text
<bdo>	Defines the direction of text display
<u>	Deprecated. Defines underlined text
Output	
<pre>	Defines preformatted text
<code>	Defines computer code text
<tt>	Defines teletype text
<kbd>	Defines keyboard text
<var>	Defines a variable
<dfn>	Defines a definition term
<samp>	Defines sample computer code
<xmp>	Deprecated. Defines preformatted text
Blocks	
<acronym>	Defines an acronym
<abbr>	Defines an abbreviation
<address>	Defines an address element
<blockquote>	Defines a long quotation
<center>	Deprecated. Defines centered text
<q>	Defines a short quotation
<cite>	Defines a citation
<ins>	Defines inserted text
	Defines deleted text
<s>	Deprecated. Defines strikethrough text
<strike>	Deprecated. Defines strikethrough text
Links	
<a>	Defines an anchor
<link>	Defines a resource reference
Frames	
<frame>	Defines a sub window (a frame)
<frameset>	Defines a set of frames
<noframes>	Defines a noframe section
<iframe>	Defines an inline sub window (frame)

Input	
<form>	Defines a form
<input>	Defines an input field
<textarea>	Defines a text area
<button>	Defines a push button
<select>	Defines a selectable list
<optgroup>	Defines an option group
<option>	Defines an item in a list box
<label>	Defines a label for a form control
<fieldset>	Defines a fieldset
<legend>	Defines a title in a fieldset
<isindex>	Deprecated. Defines a single-line input field
Lists	
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dir>	Deprecated. Defines a directory list
<dl>	Defines a definition list
<dt>	Defines a definition term
<dd>	Defines a definition description
<menu>	Deprecated. Defines a menu list
Images	
	Defines an image
<map>	Defines an image map
<area>	Defines an area inside an image map
Tables	
<table>	Defines a table
<caption>	Defines a table caption
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<thead>	Defines a table header
<tbody>	Defines a table body
<tfoot>	Defines a table footer
<col>	Defines attributes for table columns
<colgroup>	Defines groups of table columns
Styles	
<style>	Defines a style definition
<div>	Defines a section in a document
	Defines a section in a document
Meta Info	
<head>	Defines information about the document
<title>	Defines the document title
<meta>	Defines meta information
<base>	Defines a base URL for all the links in a page
<basefont>	Deprecated. Defines a base font
Programming	
<script>	Defines a script
<noscript>	Defines a noscript section
<applet>	Deprecated. Defines an applet
<object>	Defines an embedded object
<param>	Defines a parameter for an object

CSS Reference

Units

Measurements

Unit	Description
%	percentage
in	inch
cm	centimeter
mm	millimeter
em	one em is equal to the current font size of the current element
ex	one ex is the x-height of a font
pt	point (1 pt is the same as 1/72 inch)
pc	pica (1 pc is the same as 12 points)
px	pixels (a dot on the computer screen)

Colors

Unit	Description
color_name	A color name (e.g. red)
rgb(x,x,x)	An RGB value (e.g. rgb(255,0,0))
rgb(x%, x%, x%)	An RGB percentage value (e.g. rgb(100%,0%,0%))
#rrggbb	A HEX number (e.g. #ff0000)

Pseudo-classes

Pseudo-class	Purpose
:active	Adds special style to a selected link
:hover	Adds special style to a link when you mouse over it
:link	Adds special style to an unvisited link
:visited	Adds special style to a visited link
:first-child	Adds special style to an element that is the first child of some other element
:lang	Allows the author to specify a language to use in a specified element

Pseudo-elements

Pseudo-element	Purpose
:first-letter	Adds special style to the first letter of a text
:first-line	Adds special style to the first line of a text
:before	Inserts some content before an element
:after	Inserts some content after an element

Properties

Are listed in the guide as the scripting notation. For the CSS use, broke the word at the capitalize letter whith an hyphen

Ex

backgroundColor (script)

background-color (CSS)

borderRightStyle (script)

border-right-style (CSS)

For the shorthand properties, if four values are specified they apply to top, right, bottom and left respectively. If there is only one value, it applies to all sides, if there are two or three, the missing values are taken from the opposite side.

Background

Property	Description	Values
background	A shorthand property for setting all background properties in one declaration	background-color background-image background-repeat background-attachment background-position
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed
background-color	Sets the background color of an element	color-rgb color-hex color-name transparent
background-image	Sets an image as the background	url none
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat

Border

Property	Description	Values
border	A shorthand property for setting all of the properties for the four borders in one declaration	border-width border-style border-color
border-bottom	A shorthand property for setting all of the properties for the bottom border in one declaration	border-bottom-width border-style border-color
border-bottom-color	Sets the color of the bottom border	border-color
border-bottom-style	Sets the style of the bottom border	border-style
border-bottom-width	Sets the width of the bottom border	thin medium thick length
border-color	Sets the color of the four borders, can have from one to four colors	color
border-left	A shorthand property for setting all of the properties for the left border in one declaration	border-left-width border-style border-color
border-left-color	Sets the color of the left border	border-color
border-left-style	Sets the style of the left border	border-style
border-left-width	Sets the width of the left border	thin thick medium length
border-right	A shorthand property for setting all of the properties for the right border in one declaration	border-right-width border-style border-color
border-right-color	Sets the color of the right border	border-color
border-right-style	Sets the style of the right border	border-style

border-right-width	Sets the width of the right border	thin medium thick length	
border-style	Sets the style of the four borders, can have from one to four styles	none hidden dotted dashed solid	double groove ridge inset outset
border-top	A shorthand property for setting all of the properties for the top border in one declaration	border-top-width border-style border-color	
border-top-color	Sets the color of the top border	border-color	
border-top-style	Sets the style of the top border	border-style	
border-top-width	Sets the width of the top border	thin thick	medium length
border-width	A shorthand property for setting the width of the four borders in one declaration, can have from one to four values	thin medium thick length	

Classification

Property	Description	Values	
clear	Sets the sides of an element where other floating elements are not allowed	left right both none	
cursor	Specifies the type of cursor to be displayed	url auto crosshair default pointer move e-resize ne-resize nw-resize	n-resize se-resize sw-resize s-resize w-resize text wait help
display	Sets how/if an element is displayed	none inline block list-item run-in compact marker table	inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption
float	Sets where an image or a text will appear in another element	left right none	
position	Places an element in a static, relative, absolute or fixed position	static relative	absolute fixed (does not work at IE)
visibility	Sets if an element should be visible or invisible	visible hidden collapse	

Dimension

Property	Description	Values
height	Sets the height of an element	auto length %
line-height	Sets the distance between lines	normal number length %
max-height	Sets the maximum height of an element	none length %
max-width	Sets the maximum width of an element	none length %

min-height	Sets the minimum height of an element	length %
min-width	Sets the minimum width of an element	length %
width	Sets the width of an element	auto % length

Font

Property	Description	Values	
font	A shorthand property for setting all of the properties for a font in one declaration	font-style font-variant font-weight font-size/line-height font-family caption icon menu message-box small-caption status-bar	
font-family	A prioritized list of font family names and/or generic family names for an element	family-name generic-family	
font-size	Sets the size of a font	xx-small x-small small medium large x-large	xx-large smaller larger length %
font-size-adjust	Specifies an aspect value for an element that will preserve the x-height of the first-choice font	none number	
font-stretch	Condenses or expands the current font-family	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	
font-style	Sets the style of the font	normal italic oblique	
font-variant	Displays text in a small-caps font or a normal font	normal small-caps	
font-weight	Sets the weight of a font	normal bold bolder lighter 100 200 300	400 500 600 700 800 900

```
.text {
font-size: 12px;
font-weight: normal;
color: black;
line-height: 16px;
font-family: verdana, arial, helvetica, sans-serif;
text-decoration: none}
```

it's better to define many alternative fonts because different operational systems use different fonts by default.

Generated Content

Property	Description	Values
content	Generates content in a document. Used with the :before and :after pseudo-elements (does not work at IE)	string url counter(name) counter(name, list-style-type) counters(name, string) counters(name, string, list-style-type) attr(X) open-quote close-quote no-open-quote no-close-quote
counter-increment	Sets how much the counter increments on each occurrence of a selector	none <i>identifier number</i>
counter-reset	Sets the value the counter is set to on each occurrence of a selector	none <i>identifier number</i>
quotes	Sets the type of quotation marks	none <i>string string</i>

List and Marker

Property	Description	Values
list-style	A shorthand property for setting all of the properties for a list in one declaration	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>
list-style-image	Sets an image as the list-item marker	none <i>url</i>
list-style-position	Sets where the list-item marker is placed in the list	inside outside
list-style-type	Sets the type of the list-item marker	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha
marker-offset		auto length

Margin

Property	Description	Values
margin	A shorthand property for setting the margin properties in one declaration	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>
margin-bottom	Sets the bottom margin of an element	auto length %
margin-left	Sets the left margin of an element	auto length %

margin-right	Sets the right margin of an element	auto length %
margin-top	Sets the top margin of an element	auto length %

Outlines

Property	Description	Values
outline	A shorthand property for setting all the outline properties in one declaration	outline-color outline-style outline-width
not supported by IE and netscape		
outline-color	Sets the color of the outline around an element	color invert
not supported by IE and netscape		
outline-style	Sets the style of the outline around an element	none dotted dashed solid double groove ridge inset outset
not supported by IE and netscape		
outline-width	Sets the width of the outline around an element	thin medium thick length
not supported by IE and netscape		

Padding

Property	Description	Values
padding	A shorthand property for setting all the outline properties in one declaration	padding-top padding-right padding-bottom padding-left
padding-bottom	Sets the bottom padding of an element	length %
padding-left	Sets the left padding of an element	length %
padding-right	Sets the right padding of an element	length %
padding-top	Sets the top padding of an element	length %

The padding property works by different ways in different browsers when used with the dimensioning properties at the same element

Positioning

Property	Description	Values
bottom	Sets how far the bottom edge of an element is above/below the bottom edge of the parent element	auto <i>length</i> %
clip	Sets the shape of an element. The element is clipped into this shape, and displayed	<i>shape</i> auto
left	Sets how far the left edge of an element is to the right/left of the left edge of the parent element	auto % <i>length</i>
overflow	Sets what happens if the content of an element overflow its area	visible hidden scroll auto
position	Places an element in a static, relative, absolute or fixed position	static relative absolute fixed

right	Sets how far the right edge of an element is to the left/right of the right edge of the parent element	auto % <i>length</i>
top	Sets how far the top edge of an element is above/below the top edge of the parent element	auto % <i>length</i>
vertical-align	Sets the vertical alignment of an element	baseline sub super top text-top middle bottom text-bottom <i>length</i> %
z-index	Sets the stack order of an element	auto <i>number</i>

Table

Property	Description	Values
border-collapse	Sets the border model of a table	collapse separate
border-spacing	Sets the distance between the borders of adjacent cells (only for the "separated borders" model)	<i>length length</i>
caption-side	Sets the position of the caption according to the table	top bottom left right
empty-cells	Sets whether cells with no visible content should have borders or not (only for the "separated borders" model)	show hide
table-layout	Sets the algorithm used to lay out the table	auto fixed

Text

Property	Description	Values
color	Sets the color of a text	color
direction	Sets the text direction	ltr rtl
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>
text-align	Aligns the text in an element	left right center justify
text-decoration	Adds decoration to text	<i>none</i> <i>underline</i> <i>overline</i> <i>line-through</i> <i>blink</i>
text-indent	Indents the first line of text in an element	<i>length</i> %
text-shadow		<i>none</i> <i>color</i> <i>length</i>
text-transform	Controls the letters in an element	<i>none</i> capitalize uppercase lowercase
unicode-bidi		normal embed bidi-override

white-space	Sets how white space inside an element is handled	normal pre nowrap
word-spacing	Increase or decrease the space between words	normal <i>length</i>

Print

Property	Description	Values
orphans	Sets the minimum number of lines for a paragraph that must be left at the bottom of a page	<i>number</i>
marks	Sets what sort of marks should be rendered outside the page box	none crop cross
page	Sets a page type to use when displaying an element	<i>auto</i> <i>identifier</i>
page-break-after	Sets the page-breaking behavior after an element	auto always avoid left right
page-break-before	Sets the page-breaking behavior before an element	auto always avoid left right
page-break-inside	Sets the page-breaking behavior inside an element	<i>auto</i> <i>avoid</i>
size	Sets the orientation and size of a page	<i>auto</i> <i>portrait</i> <i>landscape</i>
widows	Sets the minimum number of lines for a paragraph that must be left at the top of a page	<i>number</i>

HTML DOM Objects

Object	Description
Anchor	Represents an HTML a element (a hyperlink)
Applet	Represents an HTML applet element. The applet element is used to place executable content on a page
Area	Represents an area of an image-map. An image-map is an image with clickable regions
Base	Represents an HTML base element
Basefont	Represents an HTML basefont element
Body	Represents the body of the document (the HTML body)
Button	Represents a push button on an HTML form. For each instance of an HTML <input type="button"> tag on an HTML form, a Button object is created
Checkbox	Represents a checkbox on an HTML form. For each instance of an HTML <input type="checkbox"> tag on an HTML form, a Checkbox object is created
Document	Used to access all elements in a page
Event	Represents the state of an event, such as the element in which the event occurred, the state of the keyboard keys, the location of the mouse, and the state of the mouse buttons
FileUpload	For each instance of an HTML <input type="file"> tag on a form, a FileUpload object is created
Form	Forms are used to prompt users for input. Represents an HTML form element
Frame	Represents an HTML frame
Frameset	Represents an HTML frameset
Hidden	Represents a hidden field on an HTML form. For each instance of an HTML <input type="hidden"> tag on a form, a Hidden object is created
History	A predefined object which can be accessed through the history property of the Window object. This object consists of an array of URLs. These URLs are all the URLs the user has visited within a browser window
Iframe	Represents an HTML inline-frame
Image	Represents an HTML img element
Link	Represents an HTML link element. The link element can only be used within the <head> tag
Location	Contains information about the current URL
Meta	Represents an HTML meta element
Navigator	Contains information about the client browser
Option	Represents an option in a selection list on an HTML form. For each instance of an HTML <option> tag in a selection list on a form, an Option object is created
Password	Represents a password field on an HTML form. For each instance of an HTML <input type="password"> tag on a form, a Password object is created
Radio	Represents radio buttons on an HTML form. For each instance of an HTML <input type="radio"> tag on a form, a Radio object is created
Reset	Represents a reset button on an HTML form. For each instance of an HTML <input type="reset"> tag on a form, a Reset object is created
Screen	Automatically created by the JavaScript runtime engine and it contains information about the client's display screen
Select	Represents a selection list on an HTML form. For each instance of an HTML <select> tag on a form, a Select object is created
Style	Represents an individual style statement. This object can be accessed from the document or from the elements to which that style is applied

Submit	Represents a submit button on an HTML form. For each instance of an HTML <input type="submit"> tag on a form, a Submit object is created
Table	Represents an HTML table element
TableData	Represents an HTML td element
TableHeader	Represents an HTML th element
TableRow	Represents an HTML tr element
Text	Represents a text field on an HTML form. For each instance of an HTML <input type="text"> tag on a form, a Text object is created
Textarea	Represents an HTML textarea element
Window	Corresponds to the browser window. A Window object is created automatically with every instance of a <body> or <frameset> tag

Core Attributes

Not valid in base, head, html, meta, param, script, style, and title elements.

Attribute	Value	Description
class	class_rule or style_rule	The class of the element
id	id_name	A unique id for the element
style	style_definition	An inline style definition
title	tooltip_text	A text to display in a tool tip

Language Attributes

Not valid in base, br, frame, frameset, hr, iframe, param, and script elements.

Attribute	Value	Description
dir	ltr rtl	Sets the text direction
lang	language_code	Sets the language code

Keyboard Attributes

Attribute	Value	Description
accesskey	character	Sets a keyboard shortcut to access an element
tabindex	number	Sets the tab order of an element

Events

Window Events

Only valid in body and frameset elements

Attribute	Value	Description
onload	script	Script to be run when a document loads
onunload	script	Script to be run when a document unloads

Form Element Events

Only valid in form elements.

Attribute	Value	Description
onchange	script	Script to be run when the element changes
onsubmit	script	Script to be run when the form is submitted
onreset	script	Script to be run when the form is reset
onselect	script	Script to be run when the element is selected
onblur	script	Script to be run when the element loses focus
onfocus	script	Script to be run when the element gets focus

Keyboard Events

Not valid in base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title elements.

Attribute	Value	Description
onkeydown	script	What to do when key is pressed
onkeypress	script	What to do when key is pressed and released
onkeyup	script	What to do when key is released

Mouse Events

Not valid in base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title elements.

Attribute	Value	Description
onclick	script	What to do on a mouse click
ondblclick	script	What to do on a mouse doubleclick
onmousedown	script	What to do when mouse button is pressed
onmousemove	script	What to do when mouse pointer moves
onmouseover	script	What to do when mouse pointer moves over an element
onmouseout	script	What to do when mouse pointer moves out of an element
onmouseup	script	What to do when mouse button is released

JavaScript Reference

JavaScript Function

The top-level properties and functions can be used on all of the built-in JavaScript objects.

Top-level Functions

decodeURI()	Decodes an encoded URI
decodeURIComponent()	Decodes an encoded URI component
encodeURI()	Encodes a string as a URI
encodeURIComponent()	Encodes a string as a URI component
escape()	Encodes a string
eval()	Evaluates a string and executes it as if it was script code
isFinite()	Checks if a value is a finite number
isNaN()	Checks if a value is not a number
Number()	Converts an object's value to a number
parseFloat()	Parses a string and returns a floating point number
parseInt()	Parses a string and returns an integer
String()	Converts an object's value to a string
unescape()	Decodes a string encoded by escape()

Top-level Properties

Infinity, NaN, undefined

Array object

Array Object Methods

concat()	Joins two or more arrays and returns the result
join()	Puts all the elements of an array into a string. The elements are separated by a specified delimiter
pop()	Removes and returns the last element of an array
push()	Adds one or more elements to the end of an array and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes and returns the first element of an array
slice()	Returns selected elements from an existing array
sort()	Sorts the elements of an array
splice()	Removes and adds new elements to an array
toSource()	Represents the source code of an object
toString()	Converts an array to a string and returns the result
unshift()	Adds one or more elements to the beginning of an array and returns the new length
valueOf()	Returns the primitive value of an Array object

Array Object Properties

constructor, index, input, length, prototype

Boolean object

Boolean Object Methods

toSource()	Represents the source code of an object . Does not work in IE
toString()	Converts a Boolean value to a string and returns the result
valueOf()	Returns the primitive value of a Boolean object

Boolean Object Properties

constructor, object, prototype

Date object

Date Object Methods

Date()	Returns today's date and time
getDate()	Returns the day of the month from a Date object (from 1-31)
getDay()	Returns the day of the week from a Date object (from 0-6)
getMonth()	Returns the month from a Date object (from 0-11)
getFullYear()	Returns the year, as a four-digit number, from a Date object
getYear()	Returns the year, as a two-digit or a four-digit number, from a Date object. Use <code>getFullYear()</code> instead !!
getHours()	Returns the hour of a Date object (from 0-23)
getMinutes()	Returns the minutes of a Date object (from 0-59)
getSeconds()	Returns the seconds of a Date object (from 0-59)
getMilliseconds()	Returns the milliseconds of a Date object (from 0-999)
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970
getTimezoneOffset()	Returns the difference in minutes between local time and Greenwich Mean Time (GMT)
getUTCDate()	Returns the day of the month from a Date object according to universal time (from 1-31)
getUTCDay()	Returns the day of the week from a Date object according to universal time (from 0-6)
getUTCMonth()	Returns the month from a Date object according to universal time (from 0-11)
getUTCFullYear()	Returns the four-digit year from a Date object according to universal time
getUTCHours()	Returns the hour of a Date object according to universal time (from 0-23)
getUTCMinutes()	Returns the minutes of a Date object according to universal time (from 0-59)
getUTCSeconds()	Returns the seconds of a Date object according to universal time (from 0-59)
getUTCMilliseconds()	Returns the milliseconds of a Date object according to universal time (from 0-999)
parse()	Takes a date string and returns the number of milliseconds since midnight of January 1, 1970
setDate()	Sets the day of the month in a Date object (from 1-31)
setMonth()	Sets the month in a Date object (from 0-11)
setFullYear()	Sets the year in a Date object (four digits)
setYear()	Sets the year in the Date object (two or four digits). Use <code>setFullYear()</code> instead !!
setHours()	Sets the hour in a Date object (from 0-23)
setMinutes()	Set the minutes in a Date object (from 0-59)
setSeconds()	Sets the seconds in a Date object (from 0-59)
setMilliseconds()	Sets the milliseconds in a Date object (from 0-999)
setTime()	Calculates a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970
setUTCDate()	Sets the day of the month in a Date object according to universal time (from 1-31)
setUTCMonth()	Sets the month in a Date object according to universal time (from 0-11)
setUTCFullYear()	Sets the year in a Date object according to universal time (four digits)
setUTCHours()	Sets the hour in a Date object according to universal time (from 0-23)
setUTCMinutes()	Set the minutes in a Date object according to universal time (from 0-59)

setUTCSeconds()	Set the seconds in a Date object according to universal time (from 0-59)
setUTCMilliseconds()	Sets the milliseconds in a Date object according to universal time (from 0-999)
toSource()	Represents the source code of an object
toString()	Converts a Date object to a string
toGMTString()	Converts a Date object, according to Greenwich time, to a string. Use <code>toUTCString()</code> instead !!
toUTCString()	Converts a Date object, according to universal time, to a string
toLocaleString()	Converts a Date object, according to local time, to a string
UTC()	Takes a date and returns the number of milliseconds since midnight of January 1, 1970 according to universal time
valueOf()	Returns the primitive value of a Date object

Date Object Properties

constructor, prototype

Math object

Math Object Methods

abs(x)	Returns the absolute value of a number
acos(x)	Returns the arccosine of a number
asin(x)	Returns the arcsine of a number
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y,x)	Returns the angle theta of an (x,y) point as a numeric value between -PI and PI radians
ceil(x)	Returns the value of a number rounded upwards to the nearest integer
cos(x)	Returns the cosine of a number
exp(x)	Returns the value of E^x
floor(x)	Returns the value of a number rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of a number
max(x,y)	Returns the number with the highest value of x and y
min(x,y)	Returns the number with the lowest value of x and y
pow(x,y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Rounds a number to the nearest integer
sin(x)	Returns the sine of a number
sqrt(x)	Returns the square root of a number
tan(x)	Returns the tangent of an angle
toSource()	Represents the source code of an object
valueOf()	Returns the primitive value of a Math object

Math Object Properties

Property, constructor, E, LN2, LN10, LOG2E, LOG10E, PI, prototype, SQRT1_2, SQRT2

String object

String Object Methods

anchor()	Creates an HTML anchor
big()	Displays a string in a big font
blink()	Displays a blinking string. Not at IE
bold()	Displays a string in bold
charAt()	Returns the character at a specified position
charCodeAt()	Returns the Unicode of the character at a specified position
concat()	Joins two or more strings
fixed()	Displays a string as teletype text
fontcolor()	Displays a string in a specified color
fontSize()	Displays a string in a specified size
fromCharCode()	Takes the specified Unicode values and returns a string
indexOf()	Returns the position of the first occurrence of a specified string value in a string
italics()	Displays a string in italic
lastIndexOf()	Returns the position of the last occurrence of a specified string value, searching backwards from the specified position in a string
link()	Displays a string as a hyperlink
match()	Searches for a specified value in a string
replace()	Replaces some characters with some other characters in a string
search()	Searches a string for a specified value
slice()	Extracts a part of a string and returns the extracted part in a new string
small()	Displays a string in a small font
split()	Splits a string into an array of strings
strike()	Displays a string with a strikethrough
sub()	Displays a string as subscript
substr()	Extracts a specified number of characters in a string, from a start index
substring()	Extracts the characters in a string between two specified indices
sup()	Displays a string as superscript
toLowerCase()	Displays a string in lowercase letters
toUpperCase()	Displays a string in uppercase letters
toSource()	Represents the source code of an object
valueOf()	Returns the primitive value of a String object

String Object Properties

constructor, length, prototype

Statements

JavaScript statements consist of keywords used with the appropriate syntax. A single statement may span multiple lines. Multiple statements may occur on a single line if each statement is separated by a semicolon.

Statement	Descrição
block { }	A block statement is used to group zero or more statements. The block is delimited by a pair of curly brackets
break	Terminates the current loop, switch, or label statement and transfers program control to the statement following the terminated statement
const	Declares a read-only, named constant
continue	Terminates execution of the statements in the current iteration of the current or labelled loop, and continues execution of the loop with the next iteration
do...while	Creates a loop that executes a specified statement until the test condition evaluates to false. The condition is evaluated after executing the statement, resulting in the specified statement executing at least once
export	Allows a signed script to provide properties, functions, and objects to other signed or unsigned scripts
for	Creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement executed in the loop
for...in	Iterates a specified variable over all the properties of an object. For each distinct property, a specified statement is executed
function	The function statement declares a function with the specified parameters
if...else	Executes a statement if a specified condition is true. If the condition is false, another statement can be executed
import	Allows a script to import properties, functions, and objects from a signed script that has exported the information
label	Provides an identifier that can be used with break or continue to indicate where the program should continue execution
return	Specifies the value to be returned by a function
switch	Evaluates an expression, matching the expression's value to a case label, and executes statements associated with that case
throw	Throws a user-defined exception
try...catch	Marks a block of statements to try, and specifies a response, should an exception be thrown
var	Declares a variable, optionally initializing it to a value
while	Creates a loop that executes a specified statement as long as the test condition evaluates to true. The condition is evaluated before executing the statement
with	Extends the scope chain for a statement

JavaScript reserved words

Cannot be used as JavaScript variables, functions, methods, or object names. Some of these words are keywords used in JavaScript; others are reserved for future use.

abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Operators

Arithmetic Operators + - * / % ++ -- unary -	Arithmetic operators take numerical values (either literals or variables) as their operands and return a single numerical value.
Assignment Operators = += -= *= /= >>= <<= >>>= &= = ^=	An assignment operator assigns a value to its left operand based on the value of its right operand.
Bitwise Operators & ^ ~ << >> >>>	Bitwise operators treat their operands as a set of 32 bits (zeros and ones), rather than as decimal, hexadecimal, or octal numbers. For example, the decimal number nine has a binary representation of 1001. Bitwise operators perform their operations on such binary representations, but they return standard JavaScript numerical values.
Comparison Operators == != === !== > >= < <=	A comparison operator compares its operands and returns a logical value based on whether the comparison is true. The operands can be numerical or string values. Strings are compared based on standard lexicographical ordering, using Unicode values. A boolean value is returned as the result of the comparison.
Logical Operators && !	Logical operators are typically used with boolean (logical) values, and when they are, they return a boolean value. The AND (&&) and OR () operators actually return the value of one of the specified operands, so if these are used with non-boolean values, they may return a non-boolean value.
Special Operators	There are several special operators that do not fit into any other category.
String Operators + +=	In addition to the comparison operators, which can be used on string values, the concatenation operator (+) concatenates two string values together, returning another string that is the union of the two operand strings. For example, "my" + "string" returns "my string".
Member Operators	Member operators provide access to a property or method of an object.

Operator Precedence

Operator precedence determines the order in which operators are evaluated. Operators with higher precedence are evaluated first.

A common example:

3 + 4 * 5 // returns 23

The multiplication operator ("*") has higher precedence than the addition operator ("+") and thus will be evaluated first.

Associativity

Associativity determines the order in which operators of the same precedence are processed. For example, consider an expression:

a OP b OP c

Left-associativity (left-to-right) means that it is processed as (a OP b) OP c, while right-associativity (right-to-left) means it is interpreted as a OP (b OP c). Assignment operators are right-associative, so you can write:

a = b = 5;

with the expected result that a and b get the value 5. This is because the assignment operator returns the value that it assigned. First, b is set to 5. Then the a is set to the value of b.

Operators ordered from highest (1) to lowest (17) precedence.

Precedence	Operator type	Associativity	Individual operators
1	member	left-to-right	. []
	new	right-to-left	new
2	function call	left-to-right	()
3	increment	n/a	++
	decrement	n/a	--
4	logical-not	right-to-left	!
	bitwise not	right-to-left	~
	unary +	right-to-left	+
	unary negation	right-to-left	-
	typeof	right-to-left	typeof
	void	right-to-left	void
	delete	right-to-left	delete
	multiplication	left-to-right	*
	division	left-to-right	/
	modulus	left-to-right	%
6	addition	left-to-right	+
	subtraction	left-to-right	-
7	bitwise shift	left-to-right	<< >> >>>
8	relational	left-to-right	< <= > >=
	in	left-to-right	in
	instanceof	left-to-right	instanceof
9	equality	left-to-right	== != === !==
10	bitwise-and	left-to-right	&
11	bitwise-xor	left-to-right	^
12	bitwise-or	left-to-right	
13	logical-and	left-to-right	&&
14	logical-or	left-to-right	
15	conditional	right-to-left	?:
16	assignment	right-to-left	= += -= *= /=
			%=
			<<=
			>>=
			>>>=
			&=
			^=
			=
17	comma	left-to-right	,

dictionary

[illegible]

* [REDACTED]

a

b [REDACTED]

C

d

f

g [REDACTED]

i

11 of 11

m

0 [REDACTED]

p [REDACTED]

r

S

† [REDACTED]

11

Convenções sintáticas

[ie] indica que determinado verbete foi extraído da documentação da Microsoft e não se encontra entre os padrões estabelecidos pelo W3com ou ECMA script. Alguns podem funcionar em plataformas baseadas em Gecko, como Mozilla e Firefox, mas só é garantido que funcionem em plataforma Windows/Netscape.

[javaScript]

indica um método ou operando documentado pela linguagem
javaScript

itálico

Palavras em *itálico* são partes a serem mudadas nas indicações de sintaxe.

fontFamily — font-family

A letra capitalizada indica uma palavra composta. Separando as palavras com um hífen, têm-se a propriedade CSS correspondente à determinada propriedade de script. Ex: `fontFamily` é equivalente a `font-family`

▪ (scripting)

member operator

Member operators provide access to a property or method of an object.

An object is actually an associative *array* (a.k.a. *map*, *dictionary*, *hash*, *lookup table*). The keys in this array are the names of properties and methods (properties that refer to functions).

There are two ways to access these properties: dot notation and bracket notation (a.k.a. subscript operator).

property must be a valid JavaScript identifier, i.e. a sequence of alphanumerical characters, also including the underscore (“_”) and dollar sign (“\$”), that cannot start with a number. For example, `object.$1` is valid, while `object.1` is not.

Syntax

`get = object.property;`

`object.property = set;`

Ex

```
document.createElement('pre');
```

Here, the method named “createElement” is retrieved from *document* and is called.

▪ (CSS)

class selector

With the class selector you can define different styles for the same type of HTML element

Syntax

`.classname {....}`

`element.classname {...}`

Parameters

classname	String that defines the class
element	An HTML tag

Ex

CSS

```
p.right {text-align: right}
```

```
p.center {text-align: center}
```

```
.center {text-align: center}
```

HTML

```
<p class="right">
```

This paragraph will be right-aligned.

```
</p>
```

```
<p class="center">
```

This paragraph will be center-aligned.

```
</p>
```

```
<h1 class="center">
```

This heading will be center-aligned

```
</h1>
```

[]

member operator

Bracket notation

Member operators provide access to a property or method of an object.

Syntax

`get = object[property_name];`

`object[property_name] = set;`

property_name is a string. The string does not have to be a valid identifier; it can have any value, e.g. “1foo”, “!bar!”, or even “ ” (a space).

Ex

```
document['createElement']('pre');
```

This does the exact same thing as

```
document.createElement('pre');
```

{ }

(block)

Statement

A block statement is used to group zero or more statements.

The block is delimited by a pair of curly brackets.

Syntax

```
{
  statement_1
  statement_2
  ...
  statement_n
}
```

Ex

```
while (x < 10) {
  x++;
}
```

#

(id selector)

Define styles for HTML elements with the specified ID. The ID name may not start with a number to work on Geko-based software. An ID is a unique identifier to an element.

Syntax

`#idName {....}`

`htmlElement#idName {.....}`

Ex

```
#green {color: green}
```

? ... :

(conditional)

The conditional operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.

Syntax

`condition ? expr1 : expr2`

Parameters

condition	An expression that evaluates to true or false.
expr1, expr2	Expressions with values of any type.

Ex

If condition is true, the operator returns the value of `expr1`; otherwise, it returns the value of `expr2`. For example, to display a different message based on the value of the `isMember` variable, you could use this statement:

```
document.write ("The fee is " + (isMember ? "$2.00" : "$10.00"))
```

,

(comma)

The comma operator evaluates both of its operands and returns the value of the second operand.

You can use the comma operator when you want to include multiple expressions in a location that requires a single expression. The most common usage of this operator is to supply multiple parameters in a for loop

Syntax

`expr1, expr2`

Ex

For example, if `a` is a 2-dimensional array with 10 elements on a side, the following code uses the comma operator to increment two variables at once. The code prints the values of the diagonal elements in the array:

```
for (var i=0, j=9; i <= 9; i++, j--)
  document.writeln("a[" + i + "," + j + "] = " + a[i],
```

+

(addition)

Arithmetic Operator

The addition operator returns addition of two variables. It can be used with numbers or strings

Syntax

var1 + var2

++

(increment)

Arithmetic Operator

This operator increments (adds one to) its operand and returns a value. If used postfix, with operator after operand (for example, x++), then it returns the value before incrementing. If used prefix with operator before operand (for example, ++x), then it returns the value after incrementing.

Syntax

var++ or ++var

Ex

For example, if x is three, then the statement y = x++ sets y to 3 and increments x to 4. If x is 3, then the statement y = ++x increments x to 4 and sets y to 4.

-

(subtraction)

Arithmetic Operator

The subtraction operator returns subtraction of two variables.

Syntax

var1 - var2

-

(Unary Negation)

Arithmetic Operator

The unary negation operator precedes its operand and negates it.

Syntax

-var

Ex

y = -x negates the value of x and assigns that to y; that is, if x were 3, y would get the value -3 and x would retain the value 3.

--

(decrement)

Arithmetic Operator

This operator decrements (subtracts one from) its operand and returns a value. If used postfix (for example, x--), then it returns the value before decrementing. If used prefix (for example, --x), then it returns the value after decrementing.

Syntax

var--

--var

Ex

For example, if x is three, then the statement y = x-- sets y to 3 and decrements x to 2. If x is 3, then the statement y = --x decrements x to 2 and sets y to 2.

=

(equal)

Assignment Operator

An assignment operator assigns a value to its left operand based on the value of its right operand.

The basic assignment operator is equal (=), which assigns the value of its right operand to its left operand. That is, x = y assigns the value of y to x. The other assignment operators are usually shorthand for standard operations, as shown in the following table.

Shorthand operator	Meaning
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y
x <<= y	x = x << y
x >>= y	x = x >> y
x >>>= y	x = x >>> y
x &= y	x = x & y
x ^= y	x = x ^ y
x = y	x = x y

In unusual situations, the assignment operator is not identical to the Meaning expression in this table. When the left operand of an assignment operator itself contains an assignment operator, the left operand is evaluated only once.

Ex

a[i++] += 5 // i is evaluated only once

a[i++] = a[i++] + 5 // i is evaluated twice

==

(equal)

Comparison operator

Retorna verdadeiro se os operandos forem iguais. Se dois operandos não são do mesmo tipo, o JavaScript tenta converter os operandos para um tipo apropriado para a comparação

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:

3 == var1

"3" == var1

3 === '3'

===

(strict equal)

Comparison operator

Retorna verdadeiro se os operandos são iguais e do mesmo tipo

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:

3 === var1

(multiplication)

Arithmetic Operator

The multiplication operator returns multiplication of two variables.

Syntax

var1 * var2

/

(division)

Arithmetic Operator

The addition operator returns the division of the first variable by the second, as a floating-point number.

Syntax

var1/var2

//

(JavaScript coment)

Single-line notations by the author to explain what a script does. Comments are ignored by the interpreter.

Syntax

// comment text

/* ... */

(coment) [JavaScript] [CSS]

Multi-line notations by the author to explain what a script does. Comments are ignored by the interpreter.

Syntax

/* comment text in
many lines */

%

(modulus)

Arithmetic Operator

The modulus operator returns the first operand modulo the second operand, that is, var1 modulo var2, in the preceding statement, where var1 and var2 are variables. The modulo function is the integer remainder of dividing var1 by var2. For example, 12 % 5 returns 2.

Syntax

var1 % var2

|

(bitwise OR)

Bitwise operator

Returns a one in each bit position for which the corresponding bits of both operands are ones.

Syntax

a | b

&

(bitwise AND)

Bitwise operator

Returns a one in each bit position for which the corresponding bits of both operands are ones.

Syntax

a & b

&&

(logical AND)

Retorna a expressão 1 se ela puder se converter em falsa; senão; retorna a expressão 2. Quando utilizado com valores

booleanos, && retorna verdadeiro se ambos operandos forem verdadeiros; de outro modo, retorna falso

Syntax

expr1 && expr2

^

(bitwise XOR)

Bitwise operator

Returns a one in each bit position for which the corresponding bits of either but not both operands are ones.

Syntax

a ^ b

~

(bitwise NOT)

Bitwise operator

Inverts the bits of its operand.

Syntax

~ a

<<

(left shift)

Bitwise operator

Shifts a in binary representation b bits to the left, shifting in zeros from the right.

Syntax

a << b

>>

(sign-propagating right shift)

Bitwise operator

Shifts a in binary representation b bits to the right, discarding bits shifted off.

Syntax

a >> b

>>>

(zero-fill right shift)

Bitwise operator

Shifts a in binary representation b bits to the right, discarding bits shifted off, and shifting in zeros from the left.

Syntax

a >>> b

||

(logical OR)

retorna a expressão 1 se ela puder ser convertida em verdadeira; de outra forma, retorna a expressão 2. Quando utilizado com valores booleanos, vai retornar verdadeiro se um dos dois operandos forem verdadeiros se ambos forem falsos, retorna falso

Syntax

expr1 || expr2

!

(logical NOT)

Retorna falso se um operador sozinho pode ser convertido para

verdadeiro; de outra forma, retorna falso

Syntax

!expr

!=

(not equal)

Comparison operator

Retorna verdadeiro se os operandos forem diferentes. Se dois operandos não forem do mesmo tipo, o JavaScript tenta converter os operandos a um tipo apropriado para a comparação

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:
var1 != 4
var1 != "3"

!==

(strict not equal)

Comparison operator

Retorna verdadeiro se os operandos não são iguais e/ou do mesmo tipo

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:
var1 !== 3
3 !== '3'

>

(greater than)

Comparison operator

Retorna verdadeiro se o operador da esquerda for maior que o da direita

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:
var2 > var1

>=

(greater than or equal)

Comparison operator

Retorna verdadeiro se o operando da esquerda for maior ou igual ao operando da direita

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:
var2 >= var1
var1 >= 3

<

(less than)

Comparison operator

Retorna verdadeiro se o operador da esquerda for menor que o da direita

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:
var1 < var2

<=

(less than or equal to)

Comparison operator

Retorna verdadeiro se o operando da esquerda for menor ou igual ao operando da direita

Ex

Assuming that var1 has been assigned to 3 and var 2 to 4, these expressions return true:

var1 <= var2
var2 <= 5

<!-- ... -->

HTML Comment

Prevents any enclosed text or HTML source code from being parsed and displayed in the browser window.

Ex

<!-- This text will not appear in the browser window. -->

Properties

Property	Description
innerHTML	Sets or retrieves the HTML between the start and end tags of the object.
outerHTML	Sets or retrieves the object and its content in HTML.
tagName	Retrieves the tag name of the object.
text	Retrieves or sets the text of the object as a string.

<!DOCTYPE>

Specifies the Document Type Definition (DTD) to which the document conforms.

This declaration must occur at the beginning of the document, before the html tag.

The !DOCTYPE element does not require a closing tag.

Possible Values

TopElement	Specifies the top-level element type declared in the DTD. This corresponds to the Standard Generalized Markup Language (SGML) document type being declared.HTML Default. HTML.
Availability	Specifies whether the formal public identifier (FPI) is a publicly accessible object or a system resource. PUBLIC Default. Publicly accessible object. SYSTEM System resource, such as a local file or URL.
Registration	Specifies whether the organization is registered by the International Organization for Standardization (ISO). + Default. Organization name is registered. - Organization name is not registered. The Internet Engineering Task Force (IETF) and World Wide Web Consortium (W3C) are not registered ISO organizations.
Organization	Specifies a unique label indicating the name of the entity or organization responsible for the creation and maintenance of the DTD being referenced by the !DOCTYPE declarationthe Can be: OwnerID, IETF IETF , W3C W3C
Type	Specifies the public text class, the type of object being referenced. DTD Default.
Label	Specifies the public text description, a unique descriptive name for the public text being referenced. Can be appended with the version number of the public text.HTML Default. HTML.
Definition	Specifies the document type definition. Frameset frameSet documents. Strict Excludes the presentation attributes and elements that the W3C expects to phase out as support for style sheets matures. Transitional Contains everything except frameSet elements.
Language	Specifies the public text language, the natural language encoding system used in the creation of the referenced object. It is written as an ISO 639 language code (uppercase, two letters).EN Default. English language.
URL	Specifies the location of the referenced object.

Ex

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

!important

Increases the weight or importance of a particular rule.

Applies To

a, address, b, big, blockquote, body, caption, center, cite, code, col, colgroup, dd, dfn, dir, div, dl, dt, em, form, hn, html, i, img, input, input type=button, input

type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

:active

Sets the style of an <a> element when the link is engaged or active.

Applies To
A

:first-child [ie]

Applies one or more styles to any element that is the first child of its parent.

Applies To
a, abbr, acronym, address, b, big, blockquote, body, caption, center, cite, code, col, colgroup, dd, dfn, dir, div, dl, dt, em, form, hn, html, i, img, input, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

:first-letter

Applies one or more styles to the first letter of the object.

Applies To
address, blockquote, body, center, dd, div, dl, dt, fieldset, form, hn, legend, li, listing, marquee, menu, p, plaintext, pre, xmp

:first-line

Applies one or more styles to the first line of the object.

Applies To
address, blockquote, body, center, dd, div, dl, dt, fieldset, form, hn, legend, li, listing, marquee, menu, p, plaintext, pre, xmp

:hover

Sets the style of an element when the user hovers the mouse pointer over it.

Applies To
a, abbr, acronym, address, b, big, blockquote, body, caption, center, cite, code, col, colgroup, dd, dfn, dir, div, dl, dt, em, form, hn, html, i, img, input, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

:link

Sets the style of an a element when the link has not been visited recently.

Applies To
A

:visited

Sets the style of an a element when the link has been visited recently.

Applies To
A

@charset

Sets the character set for an external style sheet.

Ex
@charset "Windows-1251";

@font-face

Sets a font to embed in the HTML document. The URL must point to an embedded OpenType file (.eot or .otf format). The file contains compressed font data that is converted to a TrueType font.

Ex

```
<STYLE>
  @font-face {
    font-family:comic;
    src:url(http://valid_url/some_font_file.eot);
  }
</STYLE>
```

@import

Imports an external style sheet.

Ex

```
<STYLE TYPE="text/css">
  @import url("URL");
  P {color:blue}
</STYLE>
ou
<STYLE type="text/css">
  @import "URL";
  P {color:blue}
</STYLE>
```

@media

Sets the media types for a set of rules in a styleSheet object.

Media types

screen	Output is intended for computer screens.
print	Output is intended for printed material and for documents viewed in Print Preview mode.
all	Output is intended for all devices.
braille	Used for braille tactile feedback devices (not at IE)
embossed	Used for paged braille printers (not at IE)
handheld	Used for small or handheld devices (not at IE)
projection	Used for projected presentations, like slides (not at IE)
tty	Used for media using a fixed-pitch character grid, like teletypes and terminals (not at IE)
tv	Used for television-type devices (not at IE)
aural	Used for speech and sound synthesizers (not at IE)

Ex

```
// For computer screens, the font size is 12pt.
@media screen {
  body {font-size:12pt;}
}
// When printed, the font size is 8pt.
@media print {
  body {font-size:8pt;}
}
```

@page

Sets the dimensions, orientation, and margins of a page box in a styleSheet.

Page types

:first	Rule applies to the first page in the collection.
:left	Rule applies to pages on the left side of the bindingverso pages.
:right	Rule applies to pages on the right side of the bindingrecto pages.

<a>

defines an anchor. An anchor can be used in two ways:
To create a link to another document by using the href attribute,
to create a bookmark inside a document, by using the name or
id attribute

Ex
An anchor:
<p name="topo">blablabla</p>
...

 volte ao topo

Opening in a new window:
concretoarmado

open a link in a child directory:
seção 2

Back to a link in a root directory:
home

Optional Attributes

Attribute, charset, coords, href, name, rel, rev, shape, target, type

Standard Attributes

id, class, title, style, dir, lang, xml:lang, tabindex, accesskey

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite,
saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Event Attributes

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes,
click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop,
fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode,
getBoundingClientRect, getClientRects, getElementsByTagName, getExpression,
hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText,
insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute,
removeAttributeNode, removeBehavior, removeChild, removeExpression,
removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView,
setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:active, :first-child, :hover, :link, :visited, accelerator, background,
backgroundAttachment, backgroundColor, backgroundImage,
backgroundPosition, backgroundPositionX, backgroundPositionY,
backgroundRepeat, behavior, border, borderBottom, borderBottomColor,
borderBottomStyle, borderBottomWidth, borderColor, borderLeft,
borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor,
borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor,
borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color,
cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle,
fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid,
layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom,
marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight,
minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft,
paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight,
pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop,
posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink,
textDecorationLineThrough, textDecorationNone, textDecorationOverline,
textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition,
top, unicodeBidi, visibility, width, wordSpacing, wordWrap, writingMode, zIndex,
zoom

<abbr>

Inserts an abbreviation into an HTML page.

Ex
<abbr title="File Transfer Protocol">FTP</abbr>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite,
saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes,
cloneNode, componentFromPoint, detachEvent, fireEvent, focus,
getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects,
getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement,
insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior,
removeChild, removeExpression, removeNode, replaceAdjacentText,
replaceChild, replaceNode, setActive, setAttributeNode, setExpression,
swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY,
behavior, border, borderBottom, borderBottomColor, borderBottomStyle,
borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle,
borderLeftWidth, borderRight, borderRightColor, borderRightStyle,
borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle,
borderTopWidth, borderWidth, direction, display, filter, hasLayout, height,
layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft,
marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow,
overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight,
paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth,
posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace,
textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap,
writingMode, zoom

abbr

Sets or retrieves abbreviated text for the object. This property
can be used for rendering to non-visual media such as speech
or Braille.

Applies To

address, blockquote, body, center, dd, div, dl, dt, fieldset, form, hn, legend, li,
listing, marquee, menu, p, plaintext, pre, xmp

abort()

Cancels the current HTTP request.

syntax:

XMLHttpRequest.abort()

Applies to

XMLHttpRequest

abs() [JavaScript]

The abs() method returns the absolute value of a number.

Syntax

Math.abs(x)

Parameter

x	Required. Must be a numeric value
---	-----------------------------------

Ex

In this example get the absolute values of positive and negative numbers:

```
<script type="text/javascript">  
document.write(Math.abs(7.25) + "<br />")  
document.write(Math.abs(-7.25) + " , ")  
document.write(Math.abs(7.25-10))  
</script>
```

The output of the code above will be:

7.25
7.25, 2.75

Applies to

Math

accelerator [ie]

Sets or retrieves a string that indicates whether the object
contains an accelerator key (true or false).

Syntax

CSS

{ ACCELERATOR : *sIsAccessible* }

Scripting

object.style.accelerator(v) [= *sIsAccessible*]

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, i, iframe, ins, kbd, label, legend, li, listing, menu, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

accept

Sets or retrieves a comma-separated list of content types.

Examples of content types include “text/html”, “image/png”, “image/gif”, “video/mpeg”, “audio/basic”, “text/tcl”, “text/javascript”, and “text/vbscript”.

Syntax

HTML

<input ACCEPT = *sAccept*... >

Scripting

input.accept(v) [= *sAccept*]

Applies To

input

acceptCharset

Sets or retrieves a list of character encodings for input data that must be accepted by the server processing the form.

Syntax

HTML

<form acceptcharset = *sChar*... >

Scripting

FORM.acceptCharset(v) [= *sChar*]

Applies To

FORM

accessKey

Sets or retrieves the accelerator key for the object.

Syntax

HTML

<*ELEMENT* ACCESSKEY = *sAccessKey*... >

Scripting

object.accessKey [= *sAccessKey*]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

acos() [JavaScript]

The acos() method returns the arccosine of a number as a numeric value value between 0 and PI radians. If the parameter x is outside the range -1 to 1, the browser will return NaN.

Syntax

Math.acos(x)

Parameter

x	Required. Must be a numeric value
---	-----------------------------------

Ex

In this example we will get the arccosine of different numbers:

```
<script type="text/javascript">
document.write(Math.acos(0.64) + "<br />")
document.write(Math.acos(0) + "<br />")
document.write(Math.acos(-1) + "<br />")
document.write(Math.acos(1) + ", ")
document.write(Math.acos(2))
</script>
```

The output of the code above will be:

0.8762980611683406
1.5707963267948965
3.141592653589793
0, NaN

Applies to

Math

<acronym>

Indicates an acronym abbreviation. By marking up acronyms you can give useful information to browsers, spell checkers, translation systems and search-engine indexers.

Standard Attributes

id, class, title, style, dir, lang, xml:lang, tabindex, accesskey

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

action

Sets or retrieves the URL to which the form content is sent for processing.

Syntax

HTML

<*ELEMENT* ACTION = *sURL*... >

Scripting

object.action [= *sURL*]

Applies To

form, isindex

activeElement [ie]

Retrieves the object that has the focus when the parent document has focus.

Syntax

[*oActive* =] *document.activeElement*

Applies To

document

add()

(areas, controlRange, options)

Creates a new namespace object and adds it to the collection.

sintaxe:

object.add(oElement [,iIndex])

Parameters

<i>oElement</i>	Required. Objectthat specifies the element to add to the collection.
<i>iIndex</i>	Optional. Integerthat specifies the index position in the col-lection where the element is placed. If no value is given, the method places the element at the end of the collection.

Applies To

areas, controlRange, options, select

add() [ie]

(namespace)

Creates a new namespace object and adds it to the collection.

Syntax

oNamespace = *namespaces.add(sNamespace, sUrn [,sUrl])*

Parameters

<i>sNamespace</i>	Required. String that specifies the name of the namespace.
<i>sUrn</i>	Required. String that specifies the Uniform Resource Name (URN) of the namespace to create.
<i>sUrl</i>	Optional. String that specifies the URL of the element behavior to import into the namespace. Specifying this parameter is the same as calling the dolmport method.

Applies To
namespaces

addBehavior() [ie]

Attaches a behavior to the element.

Syntax

iID = *object.addBehavior(sUrl)*

Parameters

<i>sUrl</i>	Required. Stringthat specifies one of the following values: sValue Location of the behavior, in URL format. The URL must be bounded by quotation marks, as illustrated in the code sample below. #default#behaviorName One of the Microsoft Internet Explorerdefault behaviors, identified by its behavior name. #objID The ID attribute specified in an object tag for the binary implementation of a behavior.
-------------	--

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, STYLE, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

AddChannel() [ie]

Presents a dialog box that enables the user to either add the channel specified, or change the channel URL if it is already installed.

Syntax

window.external.AddChannel(sURLToCDF)

Parameters

<i>sURLToCDF</i>	Required. String that specifies the URL of a Channel Definition Format (CDF) file to be installed. Note The links in the CDF file must use HTTP, Secure Hypertext Transfer Protocol (HTTPS), or File Transfer Protocol (FTP) protocols. If the CDF file contains any other protocol, the addition of the channel will fail and no dialog box will appear.
------------------	--

Applies To
external

AddDesktopComponent() [ie]

Adds a Web site or image to the Microsoft Active Desktop.

Syntax

external.AddDesktopComponent(sURL, sType [,iLeft] [,iTop] [,iWidth] [,iHeight])

Parameters

<i>sURL</i>	Required. String that specifies the location of the Web site or image to be added to the Active Desktop.
<i>sType</i>	Required. String containing one of the following values that specify the type of item to add. image Specifies the component is an image. website Specifies the component is a Web site.

<i>iLeft</i>	Optional. Integer that specifies the position of the left edge, in screen coordinates.
<i>iTop</i>	Optional. Integer that specifies the position of the top edge, in screen coordinates.
<i>iWidth</i>	Optional. Integer that specifies the width, in screen units.
<i>iHeight</i>	Optional. Integer that specifies the height, in screen units.

Applies To
external

addElement()

Adds an element to the controlRange collection.

sintaxe

controlRange.addElement(oElement)

Parameter

<i>oElement</i>	Required. Objectthat specifies the element to add to the collection.
-----------------	--

Applies to
controlRange

AddFavorite() [ie]

Prompts the user with a dialog box to add the specified URL to the Favorites list.

Syntax

external.AddFavorite(sURL [,sTitle])

Parameters

<i>sURL</i>	Required. String that specifies the URL of the favorite to be added to the Favorites list.
<i>sType</i>	Optional. String that specifies the suggested title to be used in the Favorites list. The user can change the title in the Add Favorite dialog box.

Applies To
external

addImport() [ie]

Adds a style sheet to the imports collection for the specified style sheet.

Syntax

iIndexActual = *styleSheet.addImport(sURL [,iIndexRequest])*

Parameters

<i>sURL</i>	Required. String that specifies the URL of the favorite to be added to the Favorites list.
<i>iIndexRequest</i>	Optional. Integer that specifies the requested position for the style sheet in the collection. If this value is not given, the style sheet is added to the end of the collection.

Applies To
styleSheet

additive

Sets or retrieves a value that indicates whether the animation is additive with other animations.

Syntax

HTML

<ELEMENT ADDITIVE = sType... >

Scripting

object.additive(va) [= *sType*]

Stypes

replace	Default. The animation overrides the underlying value of the attribute and other lower priority animations.
sum	The animation adds to the underlying value of the attribute and other lower priority animations.

Applies To
t:TRANSITIONFILTER, t:ANIMATE, t:ANIMATECOLOR, t:ANIMATEMOTION

addPageRule() [ie]

Creates a new page object for a style sheet.

Syntax

pINewIndex = *styleSheet.addPageRule(sSelector, sStyle, iIndex)*

Parameters

<i>sSelector</i>	Required. String that specifies the selector for the new page object.
<i>iIndexRequest</i>	Required. String that specifies the style assignments for this page object. This style takes the same form as an inline style specification. For example, "color:blue" is a valid style parameter.
<i>iIndex</i>	Required. Integer that specifies the zero-based position in the pages collection where the new page object should be placed. -1 Default. The page object is added to the end of the collection.

Applies To
styleSheet

addReadRequest() [ie]

Adds an entry to the queue for read requests. Returns a boolean value

Syntax

bSuccess = *userProfile.addReadRequest(sAttributeName [, vReserved])*

Parameters

<i>sAttributeName</i>	Required. String that specifies one of the standard vCard names. If anything else is used, the request is ignored and nothing is added to the read-requests queue.
<i>vReserved</i>	Optional. Microsoft Internet Explorer currently ignores this parameter.

Applies To
userProfile

<address>

Specifies information, such as address, signature, and authorship, of the current document. The address usually renders in italic. Most browsers will add a line break before and after the address element, but line breaks inside the text you have to insert yourself.

Ex

<address>
GFAU

Rua do lago 876

Cidade Universitaria
</address>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth,

posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

addRule() [ie]

Creates a new rule for a style sheet.

Syntax

external.addDesktopComponent(sURL, sType [, iLeft] [, iTop] [, iWidth] [, iHeight])

Parameters

<i>sSelector</i>	Required. String that specifies the selector for the new rule. Single contextual selectors are valid. For example, "divpb" is a valid contextual
<i>sStyle</i>	Required. String that specifies the style assignments for this style rule. This style takes the same form as an inline style specification. For example, "color:blue" is a valid style parameter.
<i>iIndex</i>	Optional. Integer that specifies the zero-based position in the rules collection where the new style rule should be placed. -1 Default. The page object is added to the end of the collection.

Applies to
styleSheet

AddSearchProvider()

Adds a search provider to the registry.

Syntax

window.external.addSearchProvider(sUrl)

Parameters

<i>sUrl</i>	Required. String that specifies an absolute or relative URL to the OpenSearch Description World Wide Web link file for the search provider.
-------------	---

Applies to
external

alert()

Displays a dialog box containing an application-defined message.

Syntax

window.alert([sMessage])

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to
window

align

(applet, embed, fieldset, img, input, input type=image, object, select)

Sets or retrieves how the object is aligned with adjacent text.

Syntax

HTML

<ELEMENT ALIGN = *sAlign*... >

Scripting

object.align [= *sAlign*]

sAlign Values

absbottom	Aligns the bottom of the object with the absolute bottom of the surrounding text. The absolute bottom is equal to the baseline of the text minus the height of the largest descender in the text.
absmiddle	Aligns the middle of the object with the middle of the surrounding text. The absolute middle is the midpoint between the absolute bottom and text top of the surrounding text.
baseline	Aligns the bottom of the object with the baseline of the surrounding text.
bottom	Aligns the bottom of the object with the bottom of the surrounding text. The bottom is equal to the baseline minus the standard height of a descender in the text.

left	Default. Aligns the object to the left of the surrounding text. All preceding and subsequent text flows to the right of the object.
middle	Aligns the middle of the object with the surrounding text.
right	Aligns the object to the right of the surrounding text. All subsequent text flows to the left of the object.
texttop	Aligns the top of the object with the absolute top of the surrounding text. The absolute top is the baseline plus the height of the largest ascender in the text.
top	Aligns the top of the object with the top of the text. The top of the text is the baseline plus the standard height of an ascender in the text.

Applies To

applet, embed, fieldset, img, input, input type=image, object, select

align

(caption, legend)

Sets or retrieves the alignment of the caption or legend.

Syntax

HTML

<ELEMENT ALIGN = *sAlign*... >

Scripting

object.align [= *sAlign*]

sAlign values

bottom	Aligns bottom-center.
center	Aligns center.
left	Aligns left.
right	Aligns right.
top	Aligns top-center.

Applies To

caption, legend

align

(col, colgroup, div, hn, hr, p, tbody, td, tfoot, th, thead, tr)

Sets or retrieves a value that indicates the table alignment.

Syntax

HTML

<ELEMENT ALIGN = *sAlign*... >

Scripting

object.align [= *sAlign*]

sAlign values

center	Aligns center.
justify	Aligns to the left and right edge.
left	Aligns left.
right	Aligns right.

Applies To

col, colgroup, div, hn, hr, p, tbody, td, tfoot, th, thead, tr

align

Sets or retrieves the alignment of the object relative to the display or table.

Syntax

HTML

<ELEMENT ALIGN = *sAlign*... >

Scripting

object.align [= *sAlign*]

sAlign values

center	Aligns center.
left	Aligns left.
right	Aligns right.

Applies To

table, iframe

aLink

Sets or retrieves the color of all active links in the element.

Syntax

HTML

<body ALINK = *sColor*... >

Scripting

body.aLink [= *sColor*]

Applies To

body

alinkColor [ie]

Sets or retrieves the color of all active links in the document.

Syntax

HTML

document.alinkColor(*v*) [= *vColor*]

Applies To

document

all [ie]

Returns a reference to the collection of elements contained by the object. The all collection includes one element object for each valid HTML tag. If a valid tag has a matching end tag, both tags are represented by the same element object.

Syntax

[*collAll* =] *object*.all

[*oObject* =] *object*.all(*vIndex* [, *iSubIndex*])

Possible Values

<i>collAll</i>	Array of elements contained by the object.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>vIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, ins, kbd, label, legend, li, link, listing, map, marquee, menu, object, ol, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

allowTransparency [ie]

Sets or retrieves whether the object can be transparent.

Syntax

HTML

<ELEMENT ALLOWTRANSPARENCY ... >

Scripting

object.allowTransparency(*v*) [= *bClear*]

bClear values

false	Default. Object is opaque. It cannot be transparent.
true	Object can be transparent.

Applies To

frame, iframe

alt

Sets or retrieves a text alternative to the graphic.

Syntax

HTML

<ELEMENT ALT = *sTxt*... >

Scripting

object.alt [= *sTxt*]

Applies To

object, applet, area, input, input type=image, img

altHTML [ie]

Sets the optional alternative HTML script to execute if the object fails to load.

Syntax

HTML

N/A

Scripting

object.altHTML [= *sHTMLCode*]

Applies To

applet, object

altKey [ie]

Sets or retrieves a boolean value that indicates the state of the ALT key.

Syntax

[*bAltKeyDown* =] *event.altKey*

Applies To

event

altLeft [ie]

Sets or retrieves a value that indicates the state of the left ALT key.

Syntax

[*bPressed* =] *event.altLeft*

Applies To

event

anchor() [JavaScript]

The anchor() method is used to create an HTML anchor.

Syntax

stringObject.anchor(*anchordname*)

Parameter

<i>anchordname</i>	Required. Defines a name for the anchor
--------------------	---

Ex

```
<script type="text/javascript">
var txt="Nem a pau!"
document.write(txt.anchor("myanchor"))
</script>
```

The code above could be written in plain html, like this:

```
<a name="myanchor">Nem a pau!</a>
```

Applies to

String

anchors

Retrieves a collection of all a objects that have a name and/or id property. Objects in this collection are in HTML source order.

Syntax

[*collAnchors* =] *document.anchors*

[*oObject* =] *document.anchors*(*vIndex* [, *iSubIndex*])

Possible Values

<i>collAnchors</i>	Array of a objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>vIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .

Applies To

document

appName [ie]

Retrieves the code name of the browser.

Syntax

[*sCodeName* =] *object.appCodeName*

The *sCodeName* property has a default value of Mozilla.

Applies To

clientInformation, navigator

appendChild()

Appends an element as a child to the object.

Syntax

oElement = *object.appendChild*(*oNode*)

Parameter

<i>oNode</i>	Required. Object that specifies the element to append.
--------------	--

Applies to

a, abbr, acronym, address, attribute, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

appendData()

Adds a new character string to the end of the object.

Syntax

object.appendData(*sString*)

Parameter

<i>sString</i>	Required. String that specifies the new character string.
----------------	---

Applies to

comment, TextNode

<applet>

Places executable content on the page. Deprecated, use the object element.

Required Attributes

Attribute	Description
<i>height</i>	Defines the height of the applet, in px
<i>width</i>	Defines the width of the object, in px

Optional Attributes

Attribute	Value	Description
align	left right top bottom middle baseline texttop absmiddle absbottom	Defines the text alignment around the applet
alt	text	An alternate text to be displayed if the browser support applets but cannot run this applet
archive	URL	A URL to the applet when it is stored in a Java Archive or ZIP file
code	URL	A URL that points to the class of the applet
codebase	URL	Indicates the base URL of the applet if the code attribute is relative
hspace	pixels	Defines the horizontal spacing around the applet
name	unique_name	Defines a unique name for the applet (to use in scripts)

object	name	Defines the name of the resource that contains a serialized representation of the applet
title	text	Additional information to be displayed in tool tip
vspace	pixels	Defines the vertical spacing around the applet

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Event Attributes

accesskey, tabindex, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

applets

Retrieves a collection of all applet objects in the document.

Syntax

[*collApplets* =] *document*.applets

[*oObject* =] *document*.applets(*vIndex* [, *iSubIndex*])

Possible Values

<i>collApplets</i>	Array of applet objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>vIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .

Applies To

document

APPLICATION [ie]

Indicates whether the content of the object is an HTML Application (HTA) and, therefore, exempt from the browser security model.

Syntax

HTML

<ELEMENT APPLICATION=bApplication ... >

Scripting

N/A

bApplication values

no	Default. Browser security rules for unsafe content are applied; thus, all content of the frame or iframe is untrusted.
yes	All content of the frame or iframe is trusted.

Applies To

frame, iframe

applyElement() [ie]

Makes the element either a child or parent of another element. Returns a reference to the applied element.

Syntax

oElement = *object*.applyElement(*oNewElement* [, *sWhere*])

Parameters	
<i>oNewElement</i>	Required. Object that becomes the child or parent of the current element.
<i>sWhere</i>	Optional. String that specifies one of the following values: outside: Default. Specified element becomes a parent of the current element. inside: Specified element becomes a child of the current element, but contains all children of the current element.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form,

frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

applicationName [ie]

Sets or retrieves the name of the HTML Application (HTA).

Syntax

HTML

<HTA:APPLICATION APPLICATIONNAME = sAppName... >

Scripting

[*sAppName* =] *HTA:APPLICATION*.applicationName

Applies To

HTA:APPLICATION

appMinorVersion [ie]

Retrieves the application's minor version value.

Syntax

[*iMinorVersion* =] *object*.appMinorVersion

Applies To

clientInformation, navigator

appName [ie]

Retrieves the name of the browser.

Syntax

[*sAppName* =] *object*.appName

The sAppName property has a default value of Microsoft Internet Explorer.

Applies To

clientInformation, navigator

appVersion [ie]

Retrieves the platform and version of the browser.

Syntax

[*sVersion* =] *object*.appVersion

Applies To

clientInformation, navigator

archive

Sets or retrieves a character string that can be used to implement your own archive functionality for the object.

Syntax

HTML

<ELEMENT ARCHIVE = sArch... >

Scripting

object.archive(v) [= sArch]

Applies To

object, APPLET

<area>

Defines the shape, coordinates, and associated URL of one hyperlink region within a client-side image map.

In HTML the <area> tag has no end tag.

In XHTML the <area> tag must be properly closed.

Ex

<p>Click on one of the planets:</p>

```
<img src = "planets.gif"
width="145" height="126" alt="Planets" usemap = "#planetmap" />
<map id = "planetmap" name = "planetmap">
<area shape = "rect" coords = "0, 0, 82, 126" href = "sun.htm"
target = "_blank" alt = "Sun" />
<area shape = "circle" coords = "90, 58, 3" href = "mercur.htm"
target = "_blank" alt = "Mercury" />
<area shape = "circle" coords = "124, 58, 8" href = "venus.htm"
target = "_blank" alt = "Venus" />
</map>
```

Required Attributes

alt

Optional Attributes

coords, href, nohref, shape, target

Standard Attributes

id, class, title, style, dir, lang, xml:lang, tabindex, accesskey

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Event Attributes

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, onfocus, onblur

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Style Properties and attributes

behavior, max-height, max-width, min-height, min-width

Array [javascript]

The JavaScript Array object is used to store a set of values in a single variable name.

Syntax

new Array(arrayLength)

new Array(element0, element1, ..., elementN)

[element0, element1, ..., elementN]

Parameters

arrayLength	The initial length of the array. You can access this value using the length property. If the value specified is not a number, an array of length 1 is created, with the first element having the specified value. The maximum length allowed for an array is 4,294,967,295.
elementN	A list of values for the array's elements. When this form is specified, the array is initialized with the specified values as its elements, and the array's length property is set to the number of arguments.

Ex

The following example creates an array, msgArray, with a length of 0, then assigns values to msgArray[0] and msgArray[99], changing the length of the array to 100.

```
var msgArray = new Array();
msgArray[0] = "Hello";
msgArray[99] = "world";
// The following statement is true,
// because defined msgArray[99] element.
if (msgArray.length == 100)
    myVar = "The length is 100.";
```

Methods

concat, join, pop, push, reverse, shift, slice, sort, splice, toSource, toString, unshift, valueOf

Properties

Property, constructor, index, input, length, prototype

areas

Retrieves a collection of the area objects defined for the given map object.

Syntax

[oColl =] MAP.areas

[oObject =] MAP.areas(vIndex [,iSubIndex])

Possible Values

oColl	Array of area objects.
oObject	Reference to an individual item in the array of elements contained by the object.

vIndex	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
iSubIndex	Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To

document

assign() [ie]

Loads a new HTML document.

Syntax

location.assign(sURL)

Parameter

sURL	Required. String that specifies the URL of the document to load.
------	--

Applies to

location

atan() [javaScript]

The atan() method returns the arctangent of a number as a numeric value between -PI/2 and PI/2 radians.

Syntax

Math.atan(x)

Parameter

x	Required. Must be a numeric value in the range -1 to 1
---	--

Ex

```
<script type="text/javascript">
document.write(Math.atan(0.50) + "<br />")
document.write(Math.atan(-0.50) + "<br />")
document.write(Math.atan(5) + "<br />")
document.write(Math.atan(10) + "<br />")
document.write(Math.atan(-5) + "<br />")
document.write(Math.atan(-10))
</script>
```

The output of the code above will be:

```
0.4636476090008061
-0.4636476090008061
1.373400766945016
1.4711276743037347
-1.373400766945016
-1.4711276743037347
```

Applies to

Math

atan2() [javaScript]

The atan2() method returns the angle theta of an (x, y) point as a numeric value between -PI and PI radians.

Syntax

Math.atan2(x, y)

Parameter

x	Required. Must be a numeric value in the range -1 to 1
y	Required. A number

Ex

The following example returns the angle theta of different (x, y) points with the atan2() method:

```
<script type="text/javascript">
document.write(Math.atan2(0.50, 0.50) + " ")
document.write(Math.atan2(-0.50, -0.50) + "<br />")
document.write(Math.atan2(5, 5) + " ")
document.write(Math.atan2(10, 20) + "<br />")
</script>
```

The output of the code above will be:

```
0.7853981633974483, -2.356194490192345
0.7853981633974483, 0.4636476090008061
```

Applies to
Math

ATOMICSELECTION [ie]

Specifies whether the element and its contents must be selected as a whole, indivisible unit.

Syntax

HTML

<ELEMENT ATOMICSELECTION=*bAtomicSelection ...* >

Scripting

N/A

bAtomicSelection values

false	Default. The element's content can be selected individually.
true	The element and its content may only be selected as a single unit.

Applies To

a, abbr, acronym, address, area, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, textarea, tfoot, thead, td, th, tt, u, ul, var, xmp, tr, col, COLGROUP

attachEvent() [ie]

Binds the specified function to an event, so that the function gets called whenever the event fires on the object. Returns a boolean value

Syntax

bSuccess = *object*.attachEvent(*sEvent*, *fpNotify*)

Parameter

<i>sEvent</i>	Required. String that specifies any of the standard DHTML Events.
<i>fpNotify</i>	Required. Pointer that specifiesthe function to call when <i>sEvent</i> fires.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, namespace, nobr, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, window, xmp

attribute

Represents an attribute or property of an HTML element as an object. The attribute object is accessible through the attributes collection.

Ex

This example uses the attribute object to create a list of attributes that are specified.

```
<script>
function fnFind(){
    for(var i=0;i<oList.attributes.length;i++){
        if(oList.attributes[i].specified){
            alert(oList.attributes[i].nodeName + " = "
                + oList.attributes[i].nodeValue);
        }
    }
}
</script>
```

```
<UL onclick="fnFind()">
<li ID = "oList" ACCESSKEY = "L">List Item 1
</UL>
```

Properties

Property	Description
expando	Sets or retrieves a value indicating whether arbitrary variables can be created within the object.
firstChild	Retrieves a reference to the first child in the childNodes collection of the object.
lastChild	Retrieves a reference to the last child in the childNodes collection of an object.
name	Sets or retrieves the name of the object.
nextSibling	Retrieves a reference to the next child of the parent for the object.
nodeName	Retrieves the name of a particular type of node.
nodeType	Retrieves the type of the requested node.
nodeValue	Sets or retrieves the value of a node.
ownerDocument	Retrieves the document object associated with the node.
parentNode	Retrieves the parent object in the document hierarchy.
previousSibling	Retrieves a reference to the previous child of the parent for the object.
specified	Retrieves whether an attribute has been specified.
value	Sets or retrieves the value of the object.

Collections

attributes, childNodes

Methods

appendChild, cloneNode, hasChildNodes, insertBefore, removeChild, replaceChild

attributes

Retrieves a collection of attributes of the object.

Syntax

[*oColl* =] *object*.attributes

[*oObject* =] *object*.attributes(*iIndex*)

Possible Values

<i>oColl</i>	Zero-based array of attributes applied to the object.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextid, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

autocomplete [ie]

Sets or retrieves the status of AutoComplete for the object.

Syntax

HTML

<ELEMENT AUTOCOMPLETE = *sDisabled...>*

Scripting

object.autocomplete(*v*) [= *sDisabled*]

sDisabled values

off	AutoComplete is disabled.
on	AutoComplete is enabled. Any string other than off enables AutoComplete

Applies To

input type=password, input type=text, FORM

Specifies that the text should be rendered in bold.

Ex

Bold text

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

back() [ie]

Loads a previous URL from the History list.

Syntax

history.back([iDistance])

Parameters

iDistance	Optional. Integer that specifies the number of URLs to go back. If no value is provided, the previous URL is loaded.
-----------	--

Applies to

history

background

(body)

Sets or retrieves up to five separate background properties of the object. Deprecated in HTML.

Syntax

HTML

<body BACKGROUND = sURL... >

Scripting

body.background [= sURL]

Applies To

body

background

(css)

Sets or retrieves the background picture tiled behind the text and graphics on the page.

Syntax

CSS

{ background : sBackground }

Scripting

object.style.background [= sBackground]

sBackground values

color	Any of the range of color values available to the backgroundColor property.
image	Any of the range of image values available to the backgroundImage property.
repeat	Any of the range of repeat values available to the backgroundRepeat property.
attachment	Any of the range of attachment values available to the backgroundAttachment property.
position	Any of the range of position values available to the backgroundPosition property

Ex

```
<STYLE>
.style1{background:red url(nn1.jpg) no-repeat top center}
.style2{background:black url(nn1.jpeg) no-repeat bottom right}
</STYLE>
</head>
<body>
<span onmouseover="this.className='style1'"
      onmouseout="this.className='style2'">
```

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

background

Sets or retrieves the background picture tiled behind the text and graphics in the object.

Syntax

HTML

<ELEMENT BACKGROUND = sURL... >

Scripting

object.background [= sURL]

Applies To

table, td, th

backgroundAttachment

Sets or retrieves how the background image is attached to the object within the document.

Syntax

CSS

{ background-attachment : sAttachment }

Scripting

object.style.backgroundAttachment [= sAttachment]

sAttachment values

scroll	Default. Background image scrolls with the object as the document is scrolled
fixed	Background image stays fixed within the viewable area of the object

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, layoutrect, legend, li, listing, marquee, menu, ol, option, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

backgroundColor

Sets or retrieves the color behind the content of the object.

Syntax

CSS

{ background-color : sColor }

Scripting

object.style.backgroundColor [= sColor]

Color values

transparent	Default. Color of the next parent object through which the background is visible.
color	Any color value, including those specified in the Color Table.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, option, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

backgroundImage

Sets or retrieves the background image of the object.

Syntax

CSS

{ background-image : *sLocation* }

Scripting

object.style.backgroundImage [= *sLocation*]

sLocation values

none	Default. Color of the next parent through which the background is visible.
url(<i>sUrl</i>)	Location of the background image, where <i>sUrl</i> is an absolute or relative URL.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

backgroundPosition

Sets or retrieves the position of the background of the object.

Syntax

CSS

{ background-position : *sPosition* }

Scripting

object.style.backgroundPosition [= *sPosition*]

sPosition values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	Integer, followed by a percent sign (%). The value is a percentage of the width or height of the object.
<i>vAlignment</i>	Vertical alignment value. Possible values: top, center, bottom
<i>hAlignment</i>	Horizontal alignment value. Possible values: left, center, right

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

backgroundPositionX

Sets or retrieves the x-coordinate of the backgroundPosition property.

Syntax

CSS

{ background-position-x : *iPositionX* }

Scripting

object.style.backgroundPositionX [= *iPositionX*]

iPositionX values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
---------------	--

%	Integer, followed by a percent sign (%). The value is a percentage of the width or height of the object.
---	--

<i>vAlignment</i>	Vertical alignment value. Possible values: top, center, bottom
-------------------	--

<i>hAlignment</i>	Horizontal alignment value. Possible values: left, center, right
-------------------	--

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

backgroundPositionY

Sets or retrieves the y-coordinate of the backgroundPosition property.

Syntax

CSS

{ background-position-y : *iPositionY* }

Scripting

object.style.backgroundPositionY [= *iPositionY*]

iPositionX values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
---------------	--

%	Integer, followed by a percent sign (%). The value is a percentage of the width or height of the object.
---	--

<i>vAlignment</i>	Vertical alignment value. Possible values: top, center, bottom
-------------------	--

<i>hAlignment</i>	Horizontal alignment value. Possible values: left, center, right
-------------------	--

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

backgroundRepeat

Sets or retrieves how the backgroundImage property of the object is tiled.

Syntax

CSS

{ background-repeat : *sRepeat* }

Scripting

object.style.backgroundRepeat [= *sRepeat*]

sRepeat values

repeat	Default. Image is repeated horizontally and vertically
no-repeat	Image is not repeated.
repeat-x	Image is repeated horizontally.
repeat-y	Image is repeated vertically.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

balance [ie]

Sets or retrieves the value indicating how the volume of the background sound is divided between the left and right speakers. The ibalance variant specifies or receives a value within the -10, 000 to +10, 000 range. A -10, 000 value indicates that all sound is directed to the left speaker. A +10, 000 value indicates that all sound is directed to the right

speaker. A zero value indicates that the sound is balanced between the left and right speakers.

Syntax

HTML

<bgsound BALANCE = *iBalance*... >

Scripting

bgsound.balance [= *iBalance*]

Applies To

bgsound

Banner

Retrieves the Banner content of an entry in an Advanced Stream Redirector (ASX) file.

Syntax

HTML

N/A

Scripting

[*sBanner* =] *object.playlist.activeTrack*.Banner

Applies To

activeTrack, t:AUDIO, t:MEDIA, playlist, t:REF, t:VIDEO

BannerAbstract

Retrieves the BannerAbstract content of an entry in an ASX file.

Syntax

HTML

N/A

Scripting [*sBannerAbstract* =] *object.playlist.activeTrack*.

BannerAbstract

Applies To

activeTrack, t:AUDIO, t:MEDIA, playlist, t:REF, t:VIDEO

<base>

Specifies an explicit URL used to resolve links and references to external sources such as images and style sheets. When used, the base element must appear within the head of the document, before any elements that refer to an external source.

Ex

This example sets the base URL of the document to a reference folder. Internet Explorer uses the base element to resolve the link to http://www.gfau.org.

br/contravento/antigo.html

<HTML>

<head>

<base HREF=gfau.org.br/" />

</head>

<body>

Click here to go to the site

</body>

</HTML>

Required Attributes

href

Optional Attributes

target

Standard Attributes

none

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, layoutGrid, layoutGridMode, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition

<baseFont>

Sets a base font value to be used as the default font when rendering text. Deprecated, use CSS.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, fontFamily, layoutGrid, layoutGridMode, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition, whiteSpace

BaseHref [ie]

Retrieves a string of the URL where the object tag can be found.

This is often the href of the document that the object is in, or the value set by a base element.

Syntax

HTML

N/A

Scripting

[*sURL* =] *object*.BaseHref

Applies To

object

<bdo>

Allows authors to disable the bidirectional algorithm for selected fragments of text.

Ex

<bdo dir="rtl">strike back</bdo>

output

kcab ekirts

Required Attributes

dir

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle

Style Properties and attributes

accelerator, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clip, direction, display, filter, hasLayout, height, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, position, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, zoom

behavior [ie]

(marquee)

Sets or retrieves how the text scrolls in the marquee.

Syntax

HTML

<marquee BEHAVIOR = *sScroll*... >

Scripting

marquee.behavior [= *sScroll*]

sScroll values

scroll	Default. Marquee scrolls in the direction specified by the direction property. The text scrolls off the end and starts over.
alternate	Marquee scroll direction reverses when its content reaches the edge of the container.
slide	Marquee scrolls in the direction specified by the direction property. The text scrolls to the end and stops.

Applies To

marquee

behavior [ie]

(DHTML behaviors)

Deprecated. Sets or retrieves the location of the Introduction to DHTML Behaviors.

Syntax

CSS

{ behavior : *sBehavior* }

Scripting

object.style.behavior [= *sBehavior*]

sBehavior values

url(sLocation)	Script implementation of a Dynamic HTML (DHTML) behavior, where sLocation is an absolute or relative URL
url(#objID)	Binary implementation of a DHTML behavior, where objID is the ID attribute specified in an object tag
url(#default#behaviorName)	Microsoft Internet Explorer default behavior, identified by its behaviorName

Applies To

body, marquee, table, tbody, td, tfoot, th, thead, tr

behaviorUrns [ie]

Returns a collection of Uniform Resource Name (URN) strings identifying the behaviors attached to the element.

Syntax

[*oColl* =] *object*.behaviorUrns

[*oObject* =] *object*.behaviorUrns(*iIndex*)

Possible Values

<i>oColl</i>	Array of URNs identifying the behaviors attached to the element.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bg sound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextid, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

bgcolor

(col, colgroup)

Deprecated. Sets or retrieves the background color behind the object.

Syntax

HTML

<ELEMENT BGCOLOR=*sColor* ... >

Applies To

col, COLGROUP

bgColor

(document)

Deprecated. Sets or retrieves a value that indicates the background color behind the object.

Syntax

document.bgColor [= *sColor*]

Applies To

document

bgColor

Deprecated. Sets or retrieves the background color behind the object.

Syntax

HTML

<ELEMENT BGCOLOR = *sColor*... >

Scripting

object.bgColor [= *sColor*]

Applies To

body, marquee, table, tbody, td, tfoot, th, thead, tr

bgProperties [ie]

Sets or retrieves the properties of the background picture.

Syntax

HTML

<body BGPROPERTIES = *sProperties*... >

Scripting

body.bgProperties [= *sProperties*]

sProperties values

empty string	Default. Object is opaque. It cannot be transparent.
fixed	Background is fixed and cannot scroll.

Applies To

body

<bgSound> [ie]

Enables an author to create pages with background sounds or soundtracks. Supported by Internet Explorer. This element does not require a closing tag.

Standard Attributes

balance, id, loop, src, volume

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, detachEvent, dragDrop, fireEvent, getAttribute, getAttributeNode, getElementsByTagName, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, setAttribute, setAttributeNode, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

behavior, textAutospace, textUnderlinePosition

<big>

Specifies that the enclosed text should be displayed in a larger font than the current font.

Ex

<big>Big text</big>

Standard Attributes

id, class, title, style, dir, lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundColor, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

big() [JavaScript]

The big() method is used to display a string in a big font.

Syntax

stringObject.big()

Parameter

<i>anchorname</i>	Required. Defines a name for the anchor
-------------------	---

Ex

```
<script type="text/javascript">
var str="Nem a Pau!"
document.write(str.big())
</script>
```

Applies to

String

blink() [JavaScript]

The blink() method is used to display a blinking string.

Syntax

stringObject.blink()

Parameter

<i>anchorname</i>	Required. Defines a name for the anchor
-------------------	---

Ex

```
<script type="text/javascript">
var str="Nem a Pau!"
document.write(str.blink())
</script>
```

Applies to

String

blockDirection [ie]

Retrieves a string value that indicates whether the content in the block element flows from left to right, or from right to left. (read only)

Syntax

[*sFlowDirection* =] *object*.blockDirection

sFlow values

ltr	Content flows from left to right.
rtl	Content flows from right to left.

Applies To

address, blockquote, body, center, currentStyle, custom, dd, defaults, div, dl, dt, fieldset, form, hn, isindex, li, listing, menu, ol, p, plaintext, pre, ul, xmp

blockDirection [ie]

Retrieves a collection of strings that specify the names of the available block format tags.

Syntax

[*oColl* =] Dialog Helper.blockFormats

[*oObject* =] Dialog Helper.blockFormats(*ilIndex*)

Possible Values

<i>oColl</i>	A collection of the names of the available block format tags.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>ilIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

Dialog Helper

<blockquote>

Sets apart a quotation in text.

Ex

Here comes a long quotation: <blockquote> here is a long quotation here is a long quotation here is a long quotation here is a long quotation here is a long quotation </blockquote>

Optional Attributes

cite

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundColor, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

blur()

Causes the element to lose focus and fires the onblur event.

Syntax

object.blur()

Applies to

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, window, xmp

<body>

Specifies the beginning and end of the document body.

Ex

<html>
<head>
</head>

<body>

The content of the document.....

</body>

</html>

Optional Attributes

alink, background, bgcolor, link, text, vlink (deprecated, use CSS)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder

Collection

all, attributes, behaviorUrns, childNodes, children, filters, timeAll, timeChildren

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createControlRange, createTextRange, detachEvent, doScroll, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, pause, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, resume, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, textAlign, textAlignLast, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, unicodeBidi, visibility, whiteSpace, wordBreak, wordSpacing, wordWrap, zoom

bold() [JavaScript]

The bold() method is used to display a string in bold.

Syntax

stringObject.bold()

Parameter

<i>anchormame</i>	Required. Defines a name for the anchor
-------------------	---

Ex

```
<script type="text/javascript">  
var str="Nem a Pau!"  
document.write(str.bold())  
</script>
```

Applies to

String

bookmarks [ie]

Returns a collection of Microsoft ActiveX Data Objects (ADO)

bookmarks tied to the rows affected by the current event.

Syntax

[*oColl* =] event.bookmarks

[*oObject* =] event.bookmarks(*iIndex*)

Possible Values

<i>oColl</i>	Array of ADO bookmarks.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

event

Boolean [javascript]

The JavaScript Boolean object is an object wrapper for a

Boolean value.

Methods

toSource, ToString, valueOf()

Properties

constructor, prototype

border

Sets or retrieves the properties to draw around the object.

Syntax

HTML

<ELEMENT ALLOWTRANSPARENCY ... >

Scripting

object.allowTransparency(*v*) [= *bClear*]

bClear values

false	Default. Object is opaque. It cannot be transparent.
true	Object can be transparent.

Applies To

frame, iframe

border

(css)

Sets or retrieves the space between the frames, including the

3-D border.

Syntax

CSS

{ border : *sBorder* }

Scripting

object.style.border [= *sBorder*]

sBorder values

<i>width</i>	Any of the range of width values available to the borderWidth property.
<i>style</i>	Any of the range of style values available to the borderStyle property.
<i>color</i>	Any of the range of color values available to the borderColor property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

border

(frames)

Sets or retrieves the width of the border to draw around the

object. Deprecated for object, img, layoutrect, table

Syntax

HTML

<ELEMENT BORDER = *vSpace*... >

Scripting

object.border [= *vSpace*]

Applies To

frameset, iframe, object, img, layoutrect, table

borderBottom

Sets or retrieves the properties of the bottom border of the object.

Syntax

CSS

{ border-bottom : *sBottom* }

Scripting

object.style.borderBottom [= *sBottom*]

sBottom values

<i>width</i>	Any of the range of width values available to the borderWidth property.
<i>style</i>	Any of the range of style values available to the borderStyle property.
<i>color</i>	Any of the range of color values available to the borderColor property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderBottomColor

Sets or retrieves the color of the bottom border of the object.

Syntax

CSS

{ border-bottom-color : *sColor* }

Scripting

object.style.borderBottomColor [= *sColor*]

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderBottomStyle

Sets or retrieves the style of the bottom border of the object.

Syntax

CSS

{ border-bottom-style : *sStyle* }

Scripting

object.style.borderBottomStyle [= *sStyle*]

sStyle values

none	Default. Border is not drawn, regardless of any borderWidth
dotted	Border is a dotted line
dashed	Border is a dashed line
solid	Border is a solid line
double	Border is a double line drawn on top of the background of the object. The sum of the two single lines and the space between equals the borderWidth value
groove	3-D groove is drawn in colors based on the value
ridge	3-D ridge is drawn in colors based on the value
inset	3-D inset is drawn in colors based on the value
outset	3-D outset is drawn in colors based on the value

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed,

fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderBottomWidth

Sets or retrieves the width of the bottom border of the object.

Syntax

CSS

{ border-bottom-width : *sWidth* }

Scripting

object.style.borderBottomWidth [= *sWidth*]

sWidth values

medium	Default
thin	Less than the default width
thick	Greater than the default width
<i>width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderCollapse

Sets or retrieves a value that indicates whether the row and cell borders of a table are joined in a single border or detached as in standard HTML.

Syntax

CSS

{ border-collapse : *sCollapse* }

Scripting

object.style.borderCollapse [= *sCollapse*]

sCollapse values

separate	Default. Object is opaque. It cannot be transparent
collapse	Borders are collapsed, where adjacent, into a single border

Applies To

currentStyle, runtimeStyle, style, table

borderColor

(css)

Sets or retrieves the border color of the object.

Syntax

CSS

{ border-color : *sColor* }

Scripting

object.style.borderColor [= *sColor*]

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderColor

(frames)

Sets or retrieves the border color of the object.

Syntax

HTML

<ELEMENT BORDERCOLOR = *sColor*... >

Scripting

`object.borderColor [ = sColor ]`

Applies To

frame, frameset, layoutrect, table, td, th, tr

borderColorDark [ie]

Sets or retrieves the color for one of the two colors used to draw the 3-D border of the object.

Syntax

HTML

N/A

Scripting

`object.borderColorDark [ = sColor ]`

Applies To

table, td, th, tr

borderColorLight [ie]

Sets or retrieves the color for one of the two colors used to draw the 3-D border of the object.

Syntax

HTML

N/A

Scripting

`object.borderColorLight [ = sColor ]`

Applies To

table, td, th, tr

borderLeft

Sets or retrieves the properties of the left border of the object.

Syntax

CSS

`{ border-bottom : sLeft }`

Scripting

`object.style.borderBottom [ = sLeft ]`

sLeft values

<i>width</i>	Any of the range of width values available to the <code>borderWidth</code> property.
<i>style</i>	Any of the range of style values available to the <code>borderStyle</code> property.
<i>color</i>	Any of the range of color values available to the <code>borderColor</code> property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, frame, hn, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderLeftColor

Sets or retrieves the color of the left border of the object.

Syntax

CSS

`{ border-bottom-color : sColor }`

Scripting

`object.style.borderBottomColor [ = sColor ]`

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderLeftStyle

Sets or retrieves the style of the left border of the object.

Syntax

CSS

`{ border-bottom-style : sStyle }`

Scripting

`object.style.borderBottomStyle [ = sStyle ]`

sStyle values

none	Default. Border is not drawn, regardless of any <code>borderWidth</code>
dotted	Border is a dotted line
dashed	Border is a dashed line
solid	Border is a solid line
double	Border is a double line drawn on top of the background of the object. The sum of the two single lines and the space between equals the <code>borderWidth</code> value
groove	3-D groove is drawn in colors based on the value
ridge	3-D ridge is drawn in colors based on the value
inset	3-D inset is drawn in colors based on the value
outset	3-D outset is drawn in colors based on the value

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderLeftWidth

Sets or retrieves the width of the left border of the object.

Syntax

CSS

`{ border-bottom-width : sWidth }`

Scripting

`object.style.borderBottomWidth [ = sWidth ]`

sWidth values

medium	Default
thin	Less than the default width
thick	Greater than the default width
<i>width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderRight

Sets or retrieves the properties of the right border of the object.

Syntax

CSS

`{ border-bottom : sBottom }`

Scripting

`object.style.borderBottom [ = sBottom ]`

sBottom values

<i>width</i>	Any of the range of width values available to the <code>borderWidth</code> property.
<i>style</i>	Any of the range of style values available to the <code>borderStyle</code> property.
<i>color</i>	Any of the range of color values available to the <code>borderColor</code> property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, frame, hn, i, iframe, img, input type=button, input

type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderRightColor

Sets or retrieves the color of the right border of the object.

Syntax

CSS

{ border-right-color : *sColor* }

Scripting

object.style.borderRightColor [= *sColor*]

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderRightStyle

Sets or retrieves the style of the right border of the object.

Syntax

CSS

{ border-bottom-style : *sStyle* }

Scripting

object.style.borderBottomStyle [= *sStyle*]

sStyle values

none	Default. Border is not drawn, regardless of any borderWidth
dotted	Border is a dotted line
dashed	Border is a dashed line
solid	Border is a solid line
double	Border is a double line drawn on top of the background of the object. The sum of the two single lines and the space between equals the borderWidth value
groove	3-D groove is drawn in colors based on the value
ridge	3-D ridge is drawn in colors based on the value
inset	3-D inset is drawn in colors based on the value
outset	3-D outset is drawn in colors based on the value

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderRightWidth

Sets or retrieves the width of the right border of the object.

Syntax

CSS

{ border-bottom-width : *sWidth* }

Scripting

object.style.borderBottomWidth [= *sWidth*]

sWidth values

medium	Default
thin	Less than the default width
thick	Greater than the default width
<i>width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input

type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderStyle

Sets or retrieves the style of the left, right, top, and bottom borders of the object.

Syntax

CSS

{ border-style : *sStyle* }

Scripting

object.style.borderBottomStyle [= *sStyle*]

sStyle values

none	Default. Border is not drawn, regardless of any borderWidth
dotted	Border is a dotted line
dashed	Border is a dashed line
solid	Border is a solid line
double	Border is a double line drawn on top of the background of the object. The sum of the two single lines and the space between equals the borderWidth value
groove	3-D groove is drawn in colors based on the value
ridge	3-D ridge is drawn in colors based on the value
inset	3-D inset is drawn in colors based on the value
outset	3-D outset is drawn in colors based on the value

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderTop

Syntax

CSS

{ border-bottom : *sTop* }

Scripting

object.style.borderBottom [= *sTop*]

sBottom values

<i>width</i>	Any of the range of width values available to the borderWidth property.
<i>style</i>	Any of the range of style values available to the borderStyle property.
<i>color</i>	Any of the range of color values available to the borderColor property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, frame, hn, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderTopColor

Sets or retrieves the color of the top border of the object.

Syntax

CSS

{ border-bottom-color : *sColor* }

Scripting

object.style.borderBottomColor [= *sColor*]

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input

type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderTopStyle

Sets or retrieves the style of the top border of the object.

Syntax

CSS

{ border-bottom-style : *sStyle* }

Scripting

object.style.borderBottomStyle [= *sStyle*]

sStyle values

none	Default. Border is not drawn, regardless of any borderWidth
dotted	Border is a dotted line
dashed	Border is a dashed line
solid	Border is a solid line
double	Border is a double line drawn on top of the background of the object. The sum of the two single lines and the space between equals the borderWidth value
groove	3-D groove is drawn in colors based on the value
ridge	3-D ridge is drawn in colors based on the value
inset	3-D inset is drawn in colors based on the value
outset	3-D outset is drawn in colors based on the value

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderTopWidth

Sets or retrieves the width of the top border of the object.

Syntax

CSS

{ border-bottom-width : *sWidth* }

Scripting

object.style.borderBottomWidth [= *sWidth*]

sWidth values

medium	Default
thin	Less than the default width
thick	Greater than the default width
<i>width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

borderWidth

Sets or retrieves the width of the left, right, top, and bottom borders of the object.

Syntax

CSS

{ border-bottom : *sWidth* }

Scripting

object.style.borderBottom [= *sWidth*]

sWidth values

medium	Default
--------	---------

thin	Less than the default width
thick	Greater than the default width
<i>width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

bottom

(text retangle)

Set or retrieves the bottom coordinate of the rectangle surrounding the object content.

Syntax

TextRectangle.bottom [= *iCoord*]

Applies To

TextRectangle

bottom

(css)

Sets or retrieves the bottom position of the object in relation to the bottom of the next positioned object in the document hierarchy.

Syntax

CSS

{ bottom : *sBottom* }

Scripting *object*.style.bottom [= *sBottom*]

sBottom values

auto	Default. Default position, according to the regular HTML layout of the page
<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
percentage	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, address, applet, b, big, blockquote, button, center, cite, code, currentStyle, custom, dd, defaults, dfn, dir, div, dl, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

bottomMargin

Sets or retrieves the bottom margin of the entire body of the page.

Syntax

HTML

<body BOTTOMMARGIN = *sPixels*... >

Scripting

body.bottomMargin [= *sPixels*]

Applies To

body

boundElements [ie]

Returns a collection of all elements on the page bound to a data set.

Syntax

[*oColl* =] event.bookmarks

[*oObject* =] event.bookmarks(*iIndex*)

Possible Values

<i>oColl</i>	A collection of the names of the available block format tags.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.

<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.
---------------	---

Applies To
event

boundingHeight [ie]

Retrieves the height of the rectangle that bounds the TextRange object.

Syntax

HTML

[*iHeight* =] *TextRange*.boundingHeight

Applies To
TextRange

boundingLeft [ie]

Retrieves the distance between the left edge of the rectangle that bounds the TextRange object and the left side of the object that contains the TextRange.

Syntax

HTML

[*iLeft* =] *TextRange*.boundingLeft

Applies To
TextRange

boundingTop [ie]

Retrieves the distance between the top edge of the rectangle that bounds the TextRange object and the top side of the object that contains the TextRange.

Syntax

HTML

[*iTop* =] *TextRange*.boundingTop

Applies To
TextRange

boundingWidth [ie]

Retrieves the width of the rectangle that bounds the TextRange object.

Syntax

HTML

[*iWidth* =] *TextRange*.boundingWidth

Applies To
TextRange

Inserts a line break.

In HTML the
 tag has no end tag, in XHTML the
 tag must be properly closed, like this:

Ex

This text contains
a line break

output:

This text contains
a line break

Standard Attributes

id, class, title, style

Behaviors

clientCaps, download, homePage

Collection

attributes, behaviorUrns

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

display, maxHeight, maxWidth, minHeight

break

The break statement terminates the current loop, switch, or label statement and transfers program control to the statement following the terminated statement.

Syntax

break [*label*];

Where label is the identifier associated with the label of the statement.

Ex

```
function testBreak(x) {
    var i = 0;
    while (i < 6) {
        if (i == 3)
            break;
        i++;
    }
    return i * x;
}
```

browserLanguage [ie]

Retrieves the current browser language, according to the language codes

Syntax

[*sLanguage* =] *object*.browserLanguage

Applies To
clientInformation, navigator

BrandImageUri() [ie]

Retrieves the Uniform Resource Identifier (URI) of the image used for branding. Returns a string that receives the brand image URI from the user's registry

Syntax

sUri = window.external.BrandImageUri()

Applies to
external

bufferDepth [ie]

Sets or retrieves the number of bits per pixel used for colors in the off-screen bitmap buffer.

Syntax

screen.bufferDepth(*v*) [= *iBitsPerPixel*]

iBitsPerPixel values

0	Default. No explicit buffering occurs. The colorDepth property is set to the screen depth
-1	Buffering occurs at screen depth. The colorDepth property is set to the screen depth
1, 4, 8, 15, 16, 24, 32	Number of bits per pixel to use for the off-screen buffer. The colorDepth property is also set to this value. The value 15 specifies 16 bits per pixel, in which only 15 bits are used in a 5-5-5 layout of RGB values

Applies To
screen

<button>

Specifies a container for rich HTML that is rendered as a button.

Ex

<button>Clique!</button>

Optional Attributes

Attribute	Value	Description
disabled	disabled	Disables the button
name	<i>button_name</i>	Specifies a unique name for the button
type	button reset submit	Defines the type of button
value	<i>some_value</i>	Specifies an initial value for the button. The value can be changed by a script

Standard Attributes

id, class, title, style, dir, lang, xml:lang, accesskey, tabindex

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

button

Sets or retrieves the mouse button pressed by the user.

Syntax

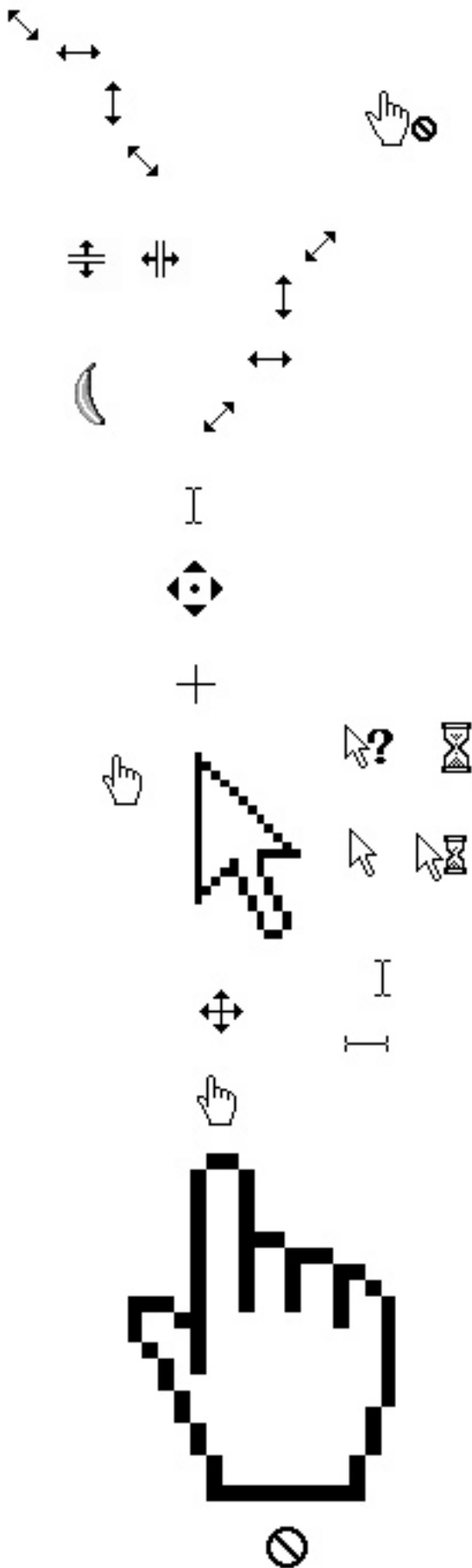
event.button [= iWhichButton]

iWhichButton values

0	Default. No button is pressed
1	Left button is pressed
2	Right button is pressed
3	Left and right buttons are both pressed
4	Middle button is pressed
5	Left and middle buttons both are pressed
6	Right and middle buttons are both pressed
7	All three buttons are pressed

Applies To

event



cancelBubble

Sets or retrieves whether the current event should bubble up the hierarchy of event handlers.

Syntax

event.cancelBubble [= *bCancel*]

bCancel values

false	Default. Bubbling is enabled, allowing the next event handler in the hierarchy to receive the event
true	Bubbling is disabled for this event, preventing the next event handler in the hierarchy from receiving the event

Applies To

event

canHaveChildren [ie]

Retrieves a value indicating whether the object can contain children.

Syntax

HTML

N/A

Scripting

[*bChildren* =] *object*.canHaveChildren

bChildren values

false	Object cannot be a parent to child objects
true	Object can be a parent to child objects

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

canHaveHTML [ie]

Sets or retrieves the value indicating whether the object can contain rich HTML markup.

Syntax

HTML – N/A

Scripting

[*bHTMLContainer* =] *object*.canHaveHTML

bHTMLContainer values

false	Object cannot contain HTML markup
true	Object can contain HTML markup

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

<caption>

Specifies a brief description for a table. The “align” attribute of the caption element is not supported in XHTML 1.0 Strict DTD and is deprecated in HTML 4.01

Ex

```
<table border="1">
<caption>This is a caption</caption>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
</tr>
</table>
```

Optional Attributes

align

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clear, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, unicodeBidi, visibility, width, wordSpacing, writingMode, zIndex, zoom

caption

Retrieves the caption object of the table.

Syntax

HTML

N/A

Scripting

[*oCaption* =] *table*.caption

oCaption values

null	Default. No caption exists for the table.
------	---

Applies To

table

ceil() [JavaScript]

The ceil() method returns the value of a number rounded UPWARDS to the nearest integer.

Syntax

Math.ceil(x)

Parameter

x	Required. A number
---	--------------------

Ex

```
<script type="text/javascript">
document.write(Math.ceil(0.60) + " , ")
document.write(Math.ceil(0.40) + " , ")
document.write(Math.ceil(5) + " , ")
document.write(Math.ceil(5.1) + " , ")
document.write(Math.ceil(-5.1) + " , ")
document.write(Math.ceil(-5.9))
</script>
```

The output of the code above will be:

1, 1, 5, 6, -5, -5

Applies to

Math

cellIndex

Retrieves the position of the object in the cells collection of a row.

Syntax

HTML

N/A

Scripting

[*iIndex* =] *object*.cellIndex

Applies To

D, th

cellPadding

Sets or retrieves the amount of space between the border of the cell and the content of the cell.

Syntax

HTML

<table CELLPADDING = *sPadding*... >

Scripting

table.cellPadding(*v*) [= *sPadding*]

sPadding values

<i>padding</i>	Integer that specifies the amount of space between the border and the content, in pixels.
<i>i%</i>	Integer, followed by a %. The value is a percentage of the available amount of space between the border and the content

Applies To

table

cells

Retrieves a collection of all cells in the table row or in the entire table.

Syntax

[*oColl* =] event.boundElements

[*oObject* =] event.boundElements(*vIndex* [, *iSubIndex*])

Possible Values

<i>oColl</i>	Array of elements found on a page that are bound to a data set.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .

Applies To

TABLE, TR

cellSpacing

Sets or retrieves the amount of space between cells in a table.

Syntax

HTML

<table CELLSPACING = *sSpacing*... >

Scripting

table.cellSpacing [= *sSpacing*]

sSpacing values

<i>spacing</i>	Integer that specifies the amount of space between cells, in pixels
<i>i%</i>	Integer, followed by a %. The value is a percentage of the available amount of space between the border and the content

Applies To

table

<center>

Centers subsequent text and images. Deprecated in HTML 4.01, the center element is not supported in XHTML 1.0 Strict DTD.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundColor, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

ch

Sets or retrieves a value that you can use to implement your own ch functionality for the object.

Syntax

HTML

<ELEMENT CH = *sChar*... >

Scripting

object.ch(*v*) [= *sChar*]

Applies To

tr, td, th, col, tbody, thead, tfoot

charAt() [JavaScript]

The charAt() method returns the character at a specified position. the first position is 0

Syntax

stringObject.charAt(index)

Parameter

<i>index</i>	Required. A number representing a position in the string
--------------	--

Ex

```
<script type="text/javascript">
var str="Nem a Paul"
document.write(str.charAt(1))
</script>
```

The output of the code above will be:

e

Applies to

String

charCodeAt() [JavaScript]

The charCodeAt() method returns the Unicode of the character at a specified position.

Syntax

stringObject.charCodeAt(*index*)

Parameter

<i>index</i>	Required. A number representing a position in the string
--------------	--

Ex

In the string “Nem a Pau!”, we will return the Unicode of the character at position 1:
<script type=“Nem a Pau!”>
var str=“Nem a Pau!”
document.write(str.charCodeAtAt(1))
</script>
The output of the code above will be:
101

Applies to
String

charset

Sets or retrieves the character set used to encode the object.

Syntax

HTML
N/A
Scripting
object.charset [= *sCharSet*]

Applies To
a, document, link, meta, script

checked

Sets or retrieves the state of the check box or radio button.

Syntax

HTML
<*ELEMENT* CHECKED ... >
Scripting
object.checked [= *bChecked*]

bChecked values

false	Default. Control is not selected
true	Control is selected

Applies To
input type=checkbox, input type=radio

childNodes

Retrieves a collection of HTML Elements and TextNode objects that are direct descendants of the specified object.

Syntax

[*oColl* =] *object*.childNodes
[*oObject* =] *object*.childNodes(*iIndex*)

Possible Values

<i>oColl</i>	Array containing the children of a specified object.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To
a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextid, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

children [ie]

Retrieves a collection of DHTML Objects that are direct descendants of the object.

Syntax

[*oColl* =] *object*.children
[*oObject* =] *object*.children(*vIndex* [, *iSubIndex*])

Possible Values

<i>oColl</i>	Array containing the direct descendants of an object.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.

<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
---------------	---

<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .
------------------	--

Applies To
a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextid, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, thead, title, tr, tt, u, ul, var, xmp

chOff

Sets or retrieves a value that you can use to implement your own chOff functionality for the object.

Syntax

HTML
<ELEMENT CHOFF = *sCharstring*... >
Scripting
object.chOff(v) [= *sCharstring*]

Applies To
tr, td, th, col, tbody, thead, tfoot

ChooseColorDlg() [ie]

Opens the system color-selection dialog box.

Syntax

rgbColor = *Dialog Helper*.ChooseColorDlg([*initColor*])

<i>initColor</i>	Optional. Variant that specifies the RGB value of the initial color selected in the color-selection dialog box. The value is specified as 0xrrggbb where rr is the red hex value, gg is the green hex value, and bb is the blue hex value.
------------------	--

Applies to
Dialog Helper

<cite>

Indicates a citation by rendering text in italic.

Ex

<cite>Citation</cite>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop,

borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

cite

Sets or retrieves reference information about the object.

Syntax

HTML

<ELEMENT CITE = *sCitation...* >

Scripting

object.cite(v) [= *sCitation*]

Applies To

blockquote, q, ins, DEL

classid

Sets or retrieves the class identifier for the object.

Syntax

HTML

<object CLASSID = *sID...* >

Scripting

object.classid(v) [= *sID*]

Applies To

object

class

Sets or retrieves the class of the object.

Syntax

HTML

<ELEMENT CLASS = *sClass...* >

Scripting

object.className [= *sClass*]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clear()

(document)

Not currently supported by I.E. . To clear all elements in the current document, use document.write(“”), followed by document.close.

clear()

(selection)

Clears the contents of the selection.

Syntax

selection.clear()

Applies to

selection

clear

Sets or retrieves whether the object allows floating objects on its left side, right side, or both, so that the next text displays past the floating objects.

Syntax

HTML

<BR CLEAR = *sValue...* >

Scripting

BR.clear [= *sValue*]

sValue values

all	Object is moved below any floating object.
left	Object is moved below any floating object on the left side.
right	Object is moved below any floating object on the right side.
none	Floating objects are allowed on all sides.

Applies To

BR

clear

Sets or retrieves the side on which floating objects are not to be positioned when a line break is inserted into the document.

Syntax

CSS

{ clear : *sClear* }

Scripting

object.style.clear(v) [= *sClear*]

sClear values

none	Default. Floating objects are allowed on both sides.
left	Object is moved below any floating object on the left side.
right	Object is moved below any floating object on the right side.
both	Object is moved below any floating object

Applies To

a, address, applet, b, big, blockquote, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, option, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clearAttributes() [ie]

Removes all attributes and values from the object.

Syntax

object.clearAttributes()

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

clearData() [ie]

Removes one or more data formats from the clipboard through dataTransfer or clipboardData object.

Syntax

pret = *object.clearData*([*sDataFormat*])

Parameter

<i>sDataFormat</i>	Optional. String that specifies one or more of the following data format values:
Text	Removes the text format.
URL	Removes the URL format.
File	Removes the file format.
HTML	Removes the HTML format.
Image	Removes the image format.

Applies to

clipboardData, dataTransfer

clearInterval() [ie]

Cancels the interval previously started using the setInterval method.

Syntax

window.clearInterval(intervalID)

Parameter

<i>intervalID</i>	Required. Integer that specifies the interval to cancel. This value must have been previously returned by the setInterval method.
-------------------	---

Applies to

window

clearRequest() [ie]

Clears all requests in the read-requests queue to prepare for new profile-information requests.

Syntax

userProfile.clearRequest()

Applies to

userProfile

clearTimeout() [ie]

Cancels a time-out that was set with the setTimeout method.

Syntax

window.clearTimeout(timeoutID)

Parameter

<i>timeoutID</i>	Required. Integer that specifies the time-out setting returned by a previous call to the setTimeout method.
------------------	---

Applies to

window

click()

Simulates a click by causing the onclick event to fire.

Syntax

object.click()

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

a, address, applet, area, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clientHeight [ie]

Retrieves the height of the object including padding, but not including margin, border, or scroll bar, in pixels.

Syntax

[*iHeight* =] *object.clientHeight*

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clientInformation

Contains information about the browser.

Ex

This example shows how to determine whether the browser can run Java applets.

```
<script LANGUAGE="JavaScript">
if (window.clientInformation.javaEnabled() == true )
    // Java is enabled; applets can run.
</script>
```

Properties

Property	Description
appName	Retrieves the code name of the browser.
appMinorVersion	Retrieves the application's minor version value.
appName	Retrieves the name of the browser.
appVersion	Retrieves the platform and version of the browser.
browserLanguage	Retrieves the current browser language.
cookieEnabled	Retrieves whether client-side persistent cookies are enabled in the browser. Persistent cookies are those that are stored on the client-side computer.
cpuClass	Retrieves a string denoting the CPU class.
onLine	Retrieves a value indicating whether the system is in global offline mode.
platform	Retrieves the name of the user's operating system.
systemLanguage	Retrieves the default language used by the operating system.
userAgent	Retrieves a string equivalent to the HTTP user-agent request header.
userLanguage	Retrieves the operating system's natural language setting.

Collection

plugins

Methods

javaEnabled, taintEnabled

Objects

userProfile

clientLeft [ie]

Retrieves the distance between the offsetLeft property and the true left side of the client area.

Syntax

HTML

N/A

Scripting

[*iDistance* =] *object.clientLeft*

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clientTop [ie]

Retrieves the distance between the offsetTop property and the true top of the client area.

Syntax

HTML

N/A

Scripting

[*iDistance* =] *object.clientTop*

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clientWidth [ie]

Retrieves the width of the object including padding, but not including margin, border, or scroll bar.

Syntax

HTML

N/A

Scripting [iWidth =] object.clientWidth

iWidth values

<i>iWidth</i>	Integer that receives the width of the object, in pixels.
---------------	---

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

clientX [ie]

Sets or retrieves the x-coordinate of the mouse pointer's position relative to the client area of the window, excluding window decorations and scroll bars.

Syntax

event.clientX [= *iXPos*]

Applies To

event

clientY [ie]

Sets or retrieves the y-coordinate of the mouse pointer's position relative to the client area of the window, excluding window decorations and scroll bars.

Syntax

event.clientY [= *iYPos*]

Applies To

event

clip

Sets or retrieves which part of a positioned object is visible.

Syntax

CSS

{ clip : *sClip* }

Scripting

object.style.clip [= *sClip*]

sClip values

auto	Default. Clip to expose entire object.
<i>rect</i> (<i>top right</i> <i>bottom left</i>)	Top, right, bottom, and left specify length values, any of which can be replaced by auto, leaving that side not clipped. The value of top specifies that everything above this value on the Y axis (with 0 at the top) is clipped. The value of right specifies that everything above this value on the X axis (with 0 at the left) is clipped. The value of bottom specifies that everything below this value on the Y axis (with 0 at the top) is clipped. The value of left specifies that everything to the left of this value on the X axis (with 0 at the left) is clipped

Applies To

a, address, applet, b, bdo, big, blockquote, button, center, cite, code, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, pre, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tr, tt, u, ul, var, xmp

clipboardData

Provides access to predefined clipboard formats for use in editing operations. This feature requires Microsoft Internet Explorer 5 or later.

Methods

clearData, getData, setData

clipBottom [ie]

Retrieves the bottom coordinate of the object clipping region.

Syntax

[*sBottom* =] *currentStyle*.clipBottom

sBottom values

auto	Bottom side of object is fully exposedthat is, the bottom side is not clipped
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

currentStyle

clipLeft [ie]

Retrieves the left coordinate of the object clipping region.

Syntax

[*sLeft* =] *currentStyle*.clipLeft

sLeft values

auto	Bottom side of object is fully exposedthat is, the bottom side is not clipped
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

currentStyle

clipRight [ie]

Retrieves the right coordinate of the object clipping region.

Syntax

[*sRight* =] *currentStyle*.clipRight

sRight values

auto	Right side of object is fully exposedthat is, the right side is not clipped
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

currentStyle

clipTop [ie]

Retrieves the top coordinate of the object clipping region.

Syntax

[*sTop* =] *currentStyle*.clipRight

sTop values

auto	Top side of object is fully exposedthat is, the top side is not clipped
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)

Applies To

currentStyle

cloneNode()

Copies a reference to the object from the document hierarchy.

Syntax

oClone = *object*.cloneNode([*bCloneChildren*])

Parameter

<i>bCloneChildren</i>	Optional. Boolean that specifies: false: Default. Cloned objects do not include childNodes. true: Cloned objects include childNodes.
-----------------------	--

Applies to

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

close()

(document)

Closes an output stream and forces the sent data to display.

Syntax

document.close()

Applies to

document

close()

Closes the current browser window or HTML Application (HTA).

Syntax

window.close()

Applies to

window

closed [ie]

Retrieves whether the referenced window is closed.

Syntax

[*bClosed* =] *window.closed*

bClosed values

false	Default. Window is open.
true	Window is closed.

Applies To

window

<code>

Specifies a code sample. Text shown in keyboard font.

Ex

<code>Computer code text</code>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

code

Sets or retrieves the URL of the file containing the compiled Java class.

Syntax

HTML

<*ELEMENT CODE* = *sURL*... >

Scripting

object.code [= *sURL*]

Applies To

applet, object

codeBase

Sets or retrieves the URL of the component.

Syntax

<*ELEMENT CODEBASE* = *sLocation*... >

Scripting

object.codeBase [= *sLocation*]

sLocation values

String that specifies or receives the URL of the component, which can be appended with a version string of the form #Version=*a*, *b*, *c*, *d*, as described in the remarks section. If the client computer has a newer version installed, no download occurs.

a High-order word of the major version of the component available at the URL.

b Low-order word of the major version of the component available at the URL.

c High-order word of the minor version of the component available at the URL.

d Low-order word of the minor version of the component available at the URL.

Applies To

applet, object

codeType

Sets or retrieves the Internet media type for the code associated with the object.

Syntax

HTML

<object CODETYPE = *sType*... >

Scripting

object.codeType [= *sType*]

Applies To

object

<col>

Specifies column-based defaults for the table properties. In

HTML the <col> tag has no end tag.

In XHTML the <col> tag must be properly closed.

Ex

This example shows a colgroup that has three columns of different widths:

```
<table border="1">
<colgroup span="3">
  <col width="20"></col>
  <col width="50"></col>
  <col width="80"></col>
</colgroup>
<tr>
  <td>1</td>
  <td>2</td>
  <td>3</td>
  <td>4</td>
</tr>
</table>
```

Optional Attributes

align, char, charoff, span, valign, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setAttribute, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, color, cursor,

direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, layoutGrid, layoutGridMode, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, overflow, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordSpacing, zIndex, zoom

<colGroup>

Specifies property defaults for a column or group of columns in a table. This element is only valid inside the <table> tag. The colgroup element is an empty element that contains attributes only. To create columns, you must specify td elements within a tr element.

```
Ex
<table border="1">
<colgroup span="3" style="color:#FF0000;">
</colgroup>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</table>
```

Optional Attributes

align, char, charoff, span, valign, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setAttribute, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, layoutGrid, layoutGridMode, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, overflow, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordSpacing, zIndex, zoom

collapse() [ie]

Moves the insertion point to the beginning or end of the current range.

Syntax

TextRange.collapse([bStart])

Parameters

<i>bStart</i>	Optional. Boolean that specifies one of the following values: true: Default. Moves the insertion point to the beginning of the text range. false: Moves the insertion point to the end of the text range.
---------------	---

Applies to
TextRange

color

(basefont, font, hr)

Sets or retrieves the color to be used by the object. Deprecated, use CSS.

Syntax

```
HTML
<ELEMENT COLOR = sColor... >

Scripting
object.color [ = sColor ]

Applies To
basefont, font, hr
```

color

(css)

Sets or retrieves the color of the text of the object.

Syntax

```
{ color : sColor }

Scripting
object.style.color(v) [ = sColor ]
```

Applies To

a, address, applet, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, option, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

colorDepth [ie]

Retrieves the number of bits per pixel used for colors on the destination device or buffer.

Syntax

[iBitsPerPixel =] screen.colorDepth

iBitsPerPixel values

1, 4, 8, 15, 16, 24, 32	Number of bits per pixel, expressed as an integer, to use for the off-screen buffer. The value 15 specifies 16 bits per pixel, in which only 15 bits are used in a 5-5-5 layout of RGB values
-------------------------	---

Applies To

screen

cols

(frameset)

Sets or retrieves the frame widths of the object.

Syntax

```
<FRAMESET COLS = sWidth... >

Scripting
FRAMESET.cols [ = sWidth ]

sWidth values
```

<i>width</i>	Integer that specifies the frame width, in pixels
<i>i%</i>	Integer, followed by a percent sign (%). The value is a percentage of total available width
<i>width*</i>	Integer that specifies the frame width as a relative value. After allocating pixel or percentage values, the remaining space is divided among all relative-sized frames

Applies To

FRAMESET

cols

(table)

Sets or retrieves the number of columns in the table.

Syntax

```
<table COLS = iCount... >

Scripting
table.cols [ = iCount ]

Applies To
table
```

cols

(textarea)

Sets or retrieves the width of the object.

Syntax

```
<textarea COLS = iCount... >
```

Scripting

```
textarea.cols(v) [= iCount]
```

Applies To

textarea

colSpan

Sets or retrieves the number columns in the table that the object should span.

Syntax

HTML

```
<ELEMENT COLSPAN = iCount... >
```

Scripting

```
object.colSpan [= iCount]
```

Applies To

td, th

compact

Sets or retrieves a Boolean value indicating whether the list should be compacted by removing extra space between list objects.

Syntax

HTML

```
<ELEMENT COMPACT ... >
```

Scripting

```
object.compact(v) [= bCompactList]
```

bCompactList values

false	Removes extra space between objects in list
true	Default. Does not remove extra space between objects in list

Applies To

frame, frameset, layoutrect, table, td, th, tr

compareEndpoints() [ie]

Compares an end point of a TextRange object with an end point of another range. Returns one of the following possible values:

-1	The end point of the object is further to the left than the end point of oRange
0	The end point of the object is at the same location as the end point of oRange
1	The end point of the object is further to the right than the end point of oRange

Syntax

```
iResult = TextRange.compareEndpoints(sType, oRange)
```

Parameters

sType	Required. String that specifies one of the following values: StartToEnd Compare the start of the TextRange object with the end of the oRange parameter. StartToStart Compare the start of the TextRange object with the start of the oRange parameter. EndToStart Compare the end of the TextRange object with the start of the oRange parameter. EndToEnd Compare the end of the TextRange object with the end of the oRange parameter.
oRange	Required. TextRange object that specifies the range to compare with the object.

Applies to

TextRange

compatMode [ie]

Retrieves a value that indicates whether standards-compliant mode is switched on for the object.

Syntax

```
[ sMode = ] document.compatMode
```

sMode values

BackCompat	Standards-compliant mode is not switched on
CSS1Compat	Standards-compliant mode is switched on

Applies To

document

complete [ie]

Retrieves whether the object is fully loaded.

Syntax

HTML

N/A

Scripting [bComplete =] object.complete

bComplete values

false	Default. Object has not been loaded.
true	Object has been loaded

Applies To

img, input, input type=image

componentFromPoint() [ie]

Returns the component located at the specified coordinates via certain events.

empty string	Component is inside the client area of the object.
outside	Component is outside the bounds of the object.
scrollbarDown	Down scroll arrow is at the specified location.
scrollbarHThumb	Horizontal scroll thumb or box is at the specified location.
scrollbarLeft	Left scroll arrow is at the specified location.
scrollbarPageDown	Page-down scroll bar shaft is at the specified location.
scrollbarPageLeft	Page-left scroll bar shaft is at the specified location.
scrollbarPageRight	Page-right scroll bar shaft is at the specified location.
scrollbarPageUp	Page-up scroll bar shaft is at the specified location.
scrollbarRight	Right scroll arrow is at the specified location.
scrollbarUp	Up scroll arrow is at the specified location.
scrollbarVThumb	Vertical scroll thumb or box is at the specified location.
handleBottom	Bottom sizing handle is at the specified location.
handleBottomLeft	Lower-left sizing handle is at the specified location.
handleBottomRight	Lower-right sizing handle is at the specified location.
handleLeft	Left sizing handle is at the specified location.
handleRight	Right sizing handle is at the specified location.
handleTop	Top sizing handle is at the specified location.
handleTopLeft	Upper-left sizing handle is at the specified location.
handleTopRight	Upper-right sizing handle is at the specified location.

Syntax

```
sScrollComponent = object.componentFromPoint(iCoordX, iCoordY)
```

Parameters

iCoordX	Optional. String that specifies the message to display in the dialog box.
iCoordY	Required. Integer that specifies the client window coordinate of y.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

concat() [JavaScript]

The `concat()` method is used to join two or more arrays or strings. This method does not change the existing arrays, it only returns a copy of the joined arrays.

Syntax

`arrayObject.concat(arrayX, arrayX, ....., arrayX)`

`stringObject.concat(stringX, stringX, ..., stringX)`

Parameters

<i>arrayX</i>	Required. One or more array objects to be joined to an array
<i>stringX</i>	Required. One or more string objects to be joined to a string

Ex

Here we create two arrays and show them as one using `concat()`:

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Lenin"
arr[1] = "Trotsky"
arr[2] = "Marx"
var arr2 = new Array(3)
arr2[0] = "Rosa"
arr2[1] = "Engels"
arr2[2] = "Mao"
document.write(arr.concat(arr2))
</script>
```

The output of the code above will be:
Lenin, Trotsky, Marx, Rosa, Engels, Mao

Now we will create two strings and show them as one using `concat()`:

```
<script type="text/javascript">
var str1 = "Nem"
var str2 = "Fudendo!"
document.write(str1.concat(str2))
</script>
```

The output of the code above will be:
Nem Fudendo!

Applies to

Array, String

confirm() [ie]

Displays a confirmation dialog box that contains an optional message as well as OK and Cancel buttons. Returns a boolean value

Syntax

bConfirmed = *window.confirm()* [*sMessage*]

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

window

contains() [ie]

Checks whether the given element is contained within the object. Returns a boolean value

Syntax

bContained = *object.contains(oElement)*

Parameters

<i>oElement</i>	Required. Element object that specifies the element to check.
-----------------	---

Applies to

a, address, applet, area, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

content

Sets or retrieves meta-information to associate with `httpEquiv` or name.

Syntax

HTML

`<meta CONTENT = vContent... >`

Scripting

meta.content [= *vContent*]

vContent values

<i>description</i>	Meta-information.
<i>refresh</i>	Integer consisting of the number of seconds to elapse before the document is refreshed. This value requires the <code>httpEquiv</code> to be set with the refresh value
<i>url</i>	Location that is loaded when the document is refreshed. This value requires the <code>httpEquiv</code> to be set with the refresh value
<i>mimetype</i>	MIME type used for the charset value
<i>charset</i>	Character set of the document. This value requires the <code>mimetype</code> value to be set with a valid MIME type, and the <code>httpEquiv</code> to be set with the refresh value
<i>noopen</i>	Microsoft Internet Explorer 6 for Microsoft Windows XP Service Pack 2 (SP2) and later. Hides the Open or Run button on the File Download dialog box. This value requires the name attribute to be set with the <code>DownloadOptions</code> value
<i>nosave</i>	Internet Explorer 6 for Windows XP SP2 and later. Hides the Save button on the File Download dialog box. This value requires the name attribute to be set with the <code>DownloadOptions</code> value

Applies To

meta

contentEditable [ie]

Sets or retrieves the string that indicates whether the user can edit the content of the object.

Syntax

HTML

`<ELEMENT CONTENTEDITABLE = sCanEdit... >`

Scripting

object.contentEditable(v) [= *sCanEdit*]

sCanEdit values

false	Content cannot be edited by the user.
true	Content can be edited by the user
inherit	Default. Content's ability to be edited by user is inherited from object's parent

Applies To

defaults, a, abbr, acronym, address, b, bdo, big, blockquote, body, button, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, i, input type=button, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, p, plaintext, pre, q, rt, ruby, s, samp, small, span, strike, strong, sub, sup, textarea, tt, u, ul, var, xmp

contentOverflow [ie]

Retrieves a value that indicates whether the document contains additional content after processing the current `LayoutRect` object.

Syntax

[*bOverflow* =] *event.contentOverflow*

bOverflow values

false	All document content has been formatted
true	Additional content needs to be formatted using a new <code>LayoutRect</code>

Applies To

event

contentWindow [ie]

Retrieves the window object of the specified frame or iframe.

Syntax

HTML

N/A

Scripting

[*oWindow* =] *document.all.object.contentWindow*

Applies To

frame, iframe

controlRange [ie]

A collection of elements returned by the createControlRange or createRange method.

Ex

This example demonstrates how to use the createRange method to retrieve the controlRange collection.

```
...
function fnChangeFontFamily () {
  if (document.selection.type == "Control"){
    var oControlRange = document.selection.createRange();
    for (i = 0; i < oControlRange.length; i++)
      if (oControlRange(i).tagName != "IMG")
        oControlRange(i).style.fontFamily=event.srcElement.style.fontFamily;
  }
}
...
<!-- Text Font-Family Controls -->
<span onclick="fnchangeFontFamily();">
  <div style="height: 25px; cursor:hand; font-family:times;
    font-size:14pt; font-weight:normal; color:white">times</div>
  <div style="height: 25px; cursor:hand; font-family:arial;
    font-size:14pt; font-weight:normal; color:white">arial</div>
  <div style="height: 25px; cursor:hand; font-family:georgia;
    font-size:14pt; font-weight:normal; color:white">georgia</div>
  <div style="height: 25px; cursor:hand; font-family:verdana;
    font-size:14pt; font-weight:normal; color:white">verdana</div>
</span><br/>
...
```

const

The const statement declares a read-only, named constant. The value of a constant cannot change through re-assignment, and a constant cannot be re-declared.

Syntax

const *varname1* [= *value1*], *varname2* [= *value2*], ..., *varnameN* [= *valueN*];

Parameters

varnameN	Constant name. It can be any legal identifier.
valueN	Value of the constant. It can be any legal expression.

Ex

The following example produces the output "a is 7."

```
const a = 7;
document.writeln("a is " + a + ".");
```

constructor [javascript]

The constructor property is a reference to the function that created an object.

Syntax

object.constructor

Ex

```
<script type="text/javascript">
var test=new Array()
if (test.constructor==Array)
{document.write("This is an Array")}
if (test.constructor==Boolean)
{document.write("This is a Boolean")}
if (test.constructor==Date)
{document.write("This is a Date")}
if (test.constructor==String)
{document.write("This is a String")}
</script>
```

The output of the code above will be:

This is an Array

Applies To

JavaScript objects

continue

The continue statement terminates execution of the statements in the current iteration of the current or labelled loop, and continues execution of the loop with the next iteration.

In contrast to the break statement, continue does not terminate

the execution of the loop entirely: instead, in a while loop, it jumps back to the condition and in a for loop, it jumps to the update expression.

The continue statement can include an optional label that allows the program to jump to the next iteration of a labelled loop statement instead of the current loop. In this case, the continue statement needs to be nested within this labelled statement.

Syntax

continue [*label*];

The label param is optional

Ex

In the following example, a statement labeled checkiandj contains a statement labeled checkj. If continue is encountered, the program continues at the top of the checkj statement.

```
checkiandj:
while (i < 4) {
  document.write(i + "<br>");
  i += 1;

  checkj:
  while (j > 4) {
    document.write(j + "<br>");
    j -= 1;
    if ((j % 2) == 0)
      continue checkj;
    document.write(j + " is odd.<br>");
  }
  document.write("i = " + i + "<br>");
  document.write("j = " + j + "<br>"); }
```

cookie

Sets or retrieves the string value of a cookie.

Syntax

document.cookie [= *sCookie*]

sCookie values

expires=date;	Setting no expiration date on a cookie causes it to expire when the browser closes. If you set an expiration date, the cookie is saved across browser sessions. If you set an expiration date in the past, the cookie is deleted. Use GMT format to specify the date
domain=domainname;	Setting the domain of the cookie allows pages on a domain made up of more than one server to share cookie information
path=path;	Setting a path for the cookie allows the current document to share cookie information with other pages within the same domainthat is, if the path is set to /thispathname, all pages in /thispathname and all pages in subfolders of /thispathname can access the same cookie information
secure;	Setting a cookie as secure; means the stored cookie information can be accessed only from a secure environment

Applies To

document

cookieEnabled [ie]

Retrieves whether client-side persistent cookies are enabled in the browser. Persistent cookies are those that are stored on the client-side computer.

Syntax

[*bEnabled* =] *object*.cookieEnabled

bEnabled values

false	Browser does not support cookies
true	Browser does support cookies

Applies To

clientInformation, navigator

coords

Sets or retrieves the coordinates of the object.

Syntax

HTML

<ELEMENT COORDS = *sCoords*... >

Scripting

object.coords [= *sCoords*]

For area objects, the format of sCoords depends on the value of the SHAPE attribute of the object, as follows:

SHAPE= circ or circle	COORDS= "x1, y1, r" Where x1, y2 are the coordinates of the center of the circle, and r is the radius of the circle.
SHAPE= poly or polygon	COORDS= "x1, y1, x2, y2...xn, yn" Where each x, y pair contains the coordinates of one vertex of the polygon.
SHAPE= rect or rectangle	COORDS= "x1, y1, x2, y2" Where x1, y1 are the coordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the lower-right coordinates of the rectangle.

Applies To

a, area

Count

Retrieves the number of available block format tags.

Syntax

[*p* =] *object.Count*

Applies To

blockFormats, fonts

cos() [JavaScript]

The cos() method returns the cosine of a number, a numeric value between -1 and 1.

Syntax

Math.cos(x)

Parameter

<i>x</i>	Required. A number
----------	--------------------

Ex

```
<script type="text/javascript">
document.write(Math.cos(3) + "<br />")
document.write(Math.cos(-3) + "<br />")
document.write(Math.cos(0) + "<br />")
document.write(Math.cos(Math.PI) + "<br />")
document.write(Math.cos(2*Math.PI))
</script>
```

The output of the code above will be:

-0.9899924966004454

-0.9899924966004454

1

-1

1

Applies to

Math

createAttribute()

Creates an attribute objectwith a specified name.

Syntax

oAttribute = *document.createAttribute(sName)*

Parameters

<i>sName</i>	Required. String that sets the attribute object's name. Must contain only valid characters.
--------------	---

Applies to

document

createCaption()

Creates an empty caption element in the table. Returns a caption object

Syntax

oCaption = *table.createCaption()*

Applies to

table

createComment()

Creates a comment objectwith the specified data. Returns a reference to the new comment object.

Syntax

oComment = *document.createComment(sData)*

Parameters

<i>sData</i>	Required. String that sets the comment object's data.
--------------	---

Applies to

document

createControlRange() [ie]

Creates a controlRange collection of nontext elements. Returns a controlRange collection

Syntax

oControlRange = *body.createControlRange()*

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

body

createDocumentFragment()

Creates a new document. Returnsthe newly created document

Syntax

oNewDoc = *document.createDocumentFragment()*

Applies to

document

createElement()

Creates an instance of the element for the specified tag.

Returns a reference to the new element

Syntax

oElement = *document.createElement(sTag)*

Parameters

<i>sTag</i>	Required. String that specifies the name of an element.
-------------	---

Applies to

document

createEventObject() [ie]

Generates an event object for passing event context information when using the fireEvent method. Returns an event object

Syntax

oNewEvent = *document.createEventObject([oExistingEvent])*

Parameters

<i>oExistingEvent</i>	Optional. Object that specifies an existing event object on which to base the new object.
-----------------------	---

Applies to

document

createPopup() [ie]

Creates a popup window. Returns the popup window object

Syntax

oPopup = *window.createPopup([vArgs])*

Parameters

<i>vArgs</i>	Optional. This argument is reserved.
--------------	--------------------------------------

Applies to

window

createRange() [ie]

Creates a TextRange object from the current text selection, or a controlRange collection from a control selection. Returnsthe created TextRange object

Syntax

range = *selection.createRange()*

Applies to

selection

createRangeCollection() [ie]

Creates a TextRange object collection from the current selection. Returns a collection of TextRange objects

Syntax

rangeCollection = *selection*.createRangeCollection()

Applies to
selection

createStyleSheet() [ie]

Creates a style sheet for the document. Returns a styleSheet object

Syntax

oStyleSheet = *document*.createStyleSheet([*sURL*] [,*iIndex*]

Parameters

<i>sURL</i>	Optional. String that specifies how to add the style sheet to the document. If a file name is specified for the URL, the style information will be added as a link object. If the URL contains style information, this information will be added to the style object.
<i>iIndex</i>	Optional. Integer that specifies the index that indicates where the new style sheet is inserted in the styleSheets collection. The default is to insert the new style sheet at the end of the collection.

Applies to
document

createTextNode()

Creates a text string from the specified value. Returns the created TextNode object

Syntax

oTextNode = *document*.createTextNode([*sText*]

Parameters

<i>sText</i>	Optional. String that specifies the nodeValue property of the text node.
--------------	--

Applies to
document

createTextRange() [ie]

Creates a TextRange object for the element. Returns a TextRange object if successful, or null otherwise

Syntax

oTextRange = *object*.createTextRange()

Applies to

body, button, input type=button, input type=hidden, input type=password, input type=reset, input type=submit, input type=text, textarea

createTFoot()

Creates an empty tFoot element in the table. Returns the tFoot element object if successful, or null otherwise.

Syntax

oTFoot = *table*.createTFoot()

Applies to
window

createTHead()

Creates an empty tHead element in the table.

Syntax

alert([*sMessage*]

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to
table

cpuClass [ie]

Retrieves a string denoting the CPU class.

Syntax

[*sCPU* =] *object*.cpuClass

sCPU values

x86	Intel processor.
68K true	Motorola processor.
Alpha	Digital processor.
PPC	Motorola processor.
Other	Other CPU classes, including Sun SPARC

Applies To
clientInformation, navigator

cssText [ie]

Sets or retrieves the persisted representation of the style rule.

Syntax

object.cssText [= *sTxt*]

Applies To
runtimeStyle, style, styleSheet

ctrlKey [ie]

Sets or retrieves the state of the CTRL key.

Syntax

event.ctrlKey [= *bEvent*]

bEvent values

false	CTRL key is not pressed.
true	CTRL key is pressed.

Applies To
event

ctrlLeft [ie]

Sets or retrieves the state of the left CTRL key.

Syntax

event.ctrlKey [= *bEvent*]

bEvent values

false	CTRL key is not pressed.
true	CTRL key is pressed.

Applies To
event

currentStyle

Represents the cascaded format and style of the object as specified by global style sheets, inline styles, and HTML attributes. This feature requires Microsoft Internet Explorer 5 or later.

Ex

This example uses the currentStyle object to retrieve values of the user-defined property created in the style rule. The alert returns the value myvalue.

```
<style>
  p { myproperty:myvalue }
</style>
<body>
<p id=oprgrph>
```

```
<script>
alert(oPrgrph.currentStyle.myproperty)
</script>
```

Properties

blockDirection	Retrieves a string value that indicates whether the content in the block element flows from left to right, or from right to left.
clipBottom	Retrieves the bottom coordinate of the object clipping region.
onOffBehavior	Retrieves an object indicating whether the specified Microsoft DirectAnimation behavior is running.

Methods

getAttribute, getExpression, setAttribute, setExpression

Style Properties and attributes

accelerator, backgroundAttachment, backgroundColor, backgroundImage, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, borderBottomColor, borderBottomStyle, borderBottomWidth, borderCollapse, borderColor, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clipLeft, clipRight,

clipTop, color, cursor, direction, display, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, imeMode, layoutFlow, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, msInterpolationMode, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, position, right, rubyAlign, rubyOverhang, rubyPosition, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, tableLayout, textAlign, textAlignLast, textAutospace, textDecoration, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, wbr, xmp

cursor

Sets or retrieves the type of cursor to display as the mouse pointer moves over the object.

Syntax

CSS

{ cursor : sCursor }

Scripting

object.style.cursor(v) [= sCursor]

sCursor values

all-scroll	Microsoft Internet Explorer 6 and later. Arrows pointing up, down, left, and right with a dot in the middle, indicating that the page can be scrolled in any direction.
auto	Default. Browser determines which cursor to display based on the current context.
col-resize	Internet Explorer 6 and later. Arrows pointing left and right with a vertical bar separating them, indicating that the item/column can be resized horizontally.
crosshair	Simple cross hair.
default	Platform-dependent default cursor; usually an arrow.
hand	Hand with the first finger pointing up, as when the user moves the pointer over a link.
help	Arrow with question mark, indicating help is available.
move	Crossed arrows, indicating something is to be moved.
no-drop	Internet Explorer 6 and later. Hand with a small circle with a line through it, indicating that the dragged item cannot be dropped at the current cursor location.
not-allowed	Internet Explorer 6 and later. Circle with a line through it, indicating that the requested action will not be carried out.
pointer	Internet Explorer 6 and later. Hand with the first finger pointing up, as when the user moves the pointer over a link. Identical to hand.
progress	Internet Explorer 6 and later. Arrow with an hourglass next to it, indicating that a process is running in the background. User interaction with the page is unaffected.
row-resize	Internet Explorer 6 and later. Arrows pointing up and down with a horizontal bar separating them, indicating that the item/row can be resized vertically.
text	Editable text; usually an I-bar.
url(uri)	Internet Explorer 6 and later. Cursor is defined by the author, using a custom Uniform Resource Identifier (URI), such as url("mycursor.cur"). Cursors of type .CUR and .ANI are the only supported cursor types.

vertical-text	Internet Explorer 6 and later. Editable vertical text, indicated by a horizontal I-bar.
wait	Hourglass or watch, indicating that the program is busy and the user should wait.
-resize	Arrows, indicating an edge is to be moved; the asterisk () can be N, NE, NW, s, SE, SW, E, or Weach representing a compass direction.

Applies To

a, address, applet, b, big, blockquote, body, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, rt, ruby, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

custom

Represents a user-defined element.

The Microsoft Internet Explorer support for custom tags on an HTML page requires that a namespace be defined for the tag. Otherwise, the custom tag is treated as an unknown tag when the document is parsed. Although navigating to a page with an unknown tag in Internet Explorer does not result in an error, unknown tags have the disadvantage of not being able to contain other tags, nor can they have behaviors applied to them. To declare a namespace, use the XMLNS attribute of the HTML element.

When defining custom tags, you must enclose custom tag definitions within an @media wrapper.

Ex

```
<HTML XMLNS:CUSTOMTAG>
<head>
<STYLE>
@media all {
    CUSTOMTAG\RED { color: red; }
    CUSTOMTAG\GREEN { color: green; }
    CUSTOMTAG\BLUE { color: blue; }
}
</STYLE>
</head>
```

```
<body>
<CUSTOMTAG:RED>
This text is red because it is enclosed within opening
and closing CUSTOMTAG:RED tags.
</CUSTOMTAG:RED>
<CUSTOMTAG:GREEN>
This text is green because it is enclosed within opening
and closing CUSTOMTAG:GREEN tags.
</CUSTOMTAG:GREEN>
<CUSTOMTAG:BLUE>
This text is blue because it is enclosed within opening
and closing CUSTOMTAG:BLUE tags.
</CUSTOMTAG:BLUE>
</body>
</HTML>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Collection

all, attributes, behaviorUrns, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, click, componentFromPoint, contains, detachEvent, doScroll, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression

Objects

currentStyle, document, runtimeStyle, style

Style Properties and attributes

accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeigth, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

CustomizeSettings() [ie]

Saves the user settings from a “first run” page.

Syntax

window.external.CustomizeSettings(fSQM, fPhishing, sLocale)

Parameters

fSQM	Required. Boolean that specifies true to enable Software Quality Monitoring (SQM), or false to opt out.
fPhishing	Required. Boolean that specifies true to turn on Microsoft Phishing Filter, or false to turn it off.
sLocale	Required. String that specifies the default region and language setting for the browser.

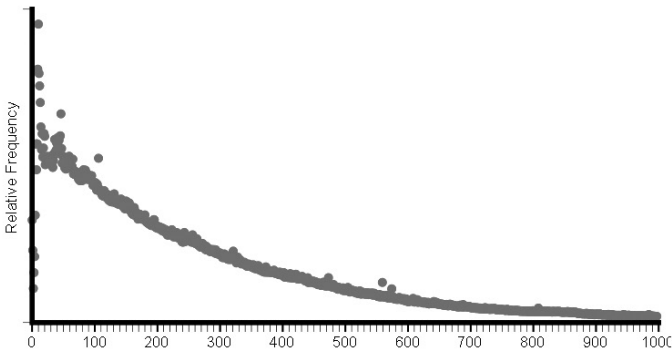
Applies to external



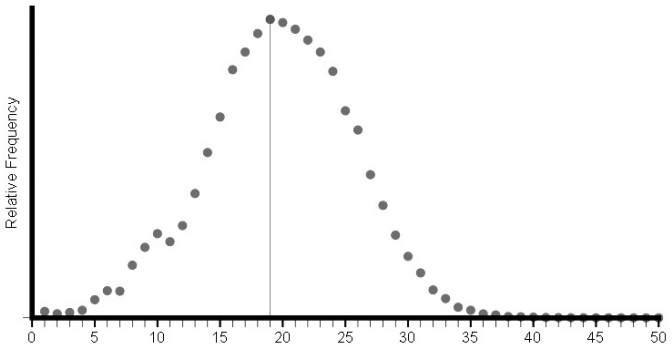
Google Code Home

Pages and elements

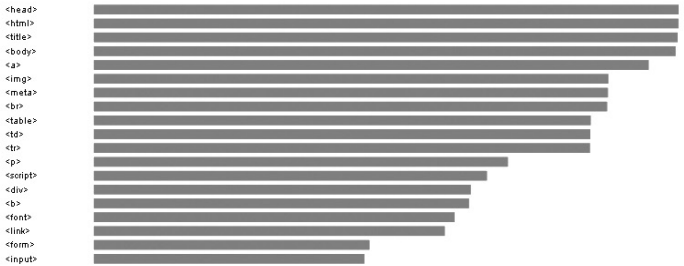
How many elements do Web pages typically have?



And how many different element types does an average Web page use? Apparently, nineteen:



What are those elements? Well, the nineteen elements used on the most pages are:



Elements and attributes

The ten elements used on the most pages, and the twenty attributes that the most pages use with those elements:



Most people (roughly 98%) include `head`, `html`, `title` and `body` elements. This is somewhat ironic, since three of those four elements are optional in HTML. It's interesting to see that most pages have a `title`, though.

The most-used element other than those four is the `a` element. Almost all pages that have an `a` element specify the `href` attribute on an `a` element somewhere, and more than half of pages use the `target` attribute on the `a` element somewhere.

The next most used element is the `img` element, with three quarters of all pages using the deprecated `border` attribute on at least one of `img` element, and around 24% of pages that use `img` elements not specifying `alt` attributes on any of their images.

The next most-frequently-featured element is `meta`, followed by `br`. The `img`, `meta` and `br` elements are each present on roughly the same number of pages.

The `table` element, along with `td` and `tr` (not shown above), are the ninth, tenth, and eleventh most frequently used elements respectively.

It seems most pages use presentational attributes: the fourth most used attribute across all elements is the `table` element's `border` attribute, followed by the `height` and `width` attributes on `img`, followed by `<table cellpadding="">`, `<table cellspacing="">`, `<img border="">`, and `<table cellpadding="">`. Interestingly, though, the most frequently used attribute on the `body` element (namely `bgcolor`) is only used on around half of pages, with all the other presentational attributes on `body` being used even less. One possible explanation is that on average, colors are mostly done using CSS, while layout is mostly done using HTML tables.

data

(comment, object)

Sets or retrieves the URL that references the data of the object.

Syntax

HTML

<ELEMENT DATA = sURL... >

Scripting

object.data(v) [= sURL]

Applies To

comment, object

data

(TextNode)

Sets or retrieves the value of a TextNode object.

Syntax

TextNode.data [= sData]

Applies To

TextNode

dataFld [ie]

Sets or retrieves which field of a given data source, as specified by the dataSrc property, to bind to the specified object.

Syntax

HTML

<ELEMENT DATAFLD = sField... >

Scripting

object.dataFld [= sField]

Applies To

a, applet, button, div, fieldset, frame, iframe, img, input type=button, input type=checkbox, input type=hidden, input type=image, input type=password, input type=radio, input type=text, label, marquee, select, span, textarea

dataFld [ie]

Sets or retrieves the data column affected by the oncellchange event.

Syntax

event.dataFld [= sdataFld]

Applies To

event

DATAFLD [ie]

Sets the field of a given data source for data binding.

Syntax

HTML

<param DATAFLD=sField ... >

Scripting

N/A

Applies To

param

dataFormatAs [ie]

Sets or retrieves how to render the data supplied to the object.

Syntax

HTML

<ELEMENT DATAFORMATAS = sFormat... >

Scripting

object.dataFormatAs [= sFormat]

sFormat values

text	Default. Data is rendered as text
html	Data is rendered as HTML
localized-text	Microsoft Internet Explorer 5.01 and later versions. Data is rendered using the locale settings of the client machine

Applies To

button, div, input type=button, label, legend, marquee, span

DATAFORMATAS [ie]

Sets whether data supplied to the object should be rendered as text or HTML.

Syntax

HTML

<param DATAFORMATAS=sFormat ... >

Scripting

N/A

sFormat values

text	Default. Data is rendered as text
html	Data is rendered as HTML

Applies To

param

dataPageSize [ie]

Sets or retrieves the number of records displayed in a table bound to a data source.

Syntax

HTML

<table DATAPAGESIZE = iSize... >

Scripting

table.dataPageSize [= iSize]

Applies To

table

dataSrc [ie]

Sets or retrieves the source of the data for data binding.

Syntax

<ELEMENT DATASRC = sID... >

Scripting

object.dataSrc [= sID]

Applies To

a, applet, button, div, frame, iframe, img, input type=button, input type=checkbox, input type=hidden, input type=image, input type=password, input type=radio, input type=text, label, marquee, select, span, table, textarea

DATASRC [ie]

Sets the source of the data for data binding.

Syntax

<ELEMENT DATASRC = sID... >

Scripting

object.dataSrc [= sID]

Applies To

param

dataTransfer

Provides access to predefined clipboard formats for use in drag-and-drop operations.

Properties

Property	Description
dropEffect	Sets or retrieves the type of drag-and-drop operation and the type of cursor to display.
effectAllowed	Sets or retrieves, on the source element, which data transfer operations are allowed for the object.

Ex

This example uses the setData and getData methods of the dataTransfer object to perform a drag-and-drop operation.

```
<head>
<script>
var sAnchorURL;
```

```
function InitiateDrag()
/* The setData parameters tell the source object to transfer data as a URL and
provide the path. */
{
    event.dataTransfer.setData("URL", oSource.href);
}
```

```
function FinishDrag()
/* The parameter passed to getData tells the target object what data format to
expect. */
{
    sAnchorURL = event.dataTransfer.getData("URL")
    oTarget.innerText = sAnchorURL;
```

```
}
</script>
</head>
<body>
<A ID=oSource HREF="about:Example_Complete" onclick="return(false)"
  ondragstart="InitiateDrag()">Test Anchor</A>
<span ID=oTarget ondragenter="FinishDrag()">Drop the Link Here</span>
</body>
```

Methods

clearData, getData, setData

Date [javascript]

The JavaScript Date object is used to work with dates and times.

Methods

Date, getDate, getDay, getMonth, getFullYear, getYear, getHours, getMinutes, getSeconds, getMilliseconds, getTime, getTimezoneOffset, getUTCDate, getUTCDay, getUTCMonth, getUTCFullYear, getUTCHours, getUTCMinutes, getUTCSeconds, getUTCMilliseconds, parse, setDate, setMonth, setFullYear, setYear, setHours, setMinutes, setSeconds, setMilliseconds, setTime, setUTCDate, setUTCMonth, setUTCFullYear, setUTCHours, setUTCMinutes, setUTCSeconds, setUTCMilliseconds, toSource, toString, toGMTString, toUTCString, toLocaleString, UTC, valueOf

Properties

constructor, prototype

Date() [javaScript]

The Date() method returns today's date and time.

Syntax

Date()

Ex

```
<script type="text/javascript">
document.write(Date())
</script>
```

The output of the code above will be:

Wed Jun 28 2006 03:08:23 GMT-0300 (Hora oficial do Brasil)

Applies to

Date

dateTime

Sets or retrieves the date and time of a modification to the object.

Syntax

HTML

<ELEMENT DATETIME = *sDatetime*... >

Scripting

object.dateTime(*v*) [= *sDatetime*]

Datetime syntax

YYYY-MM-DDThh:mm:ssTZD — Standard Date-Time Format

YYYY	Four-digit year
MM	Two-digit month (01=January, etc.)
DD	Two-digit day of month (01 through 31)
hh	Two digit hour (00 through 23) (A.M./P.M. NOT allowed)
mm	Two digit minute (00 through 59)
ss	Two digit second (00 through 59)
TZD	Time zone designator
Z	(Must be uppercase) Specifies coordinated universal time (UTC)
+hh:mm	Specified time is hh hours and mm minutes ahead of UTC.
-hh:mm	Specified time is hh hours and mm minutes behind UTC.

Applies To

ins, DEL

<dd>

Indicates the definition in a definition list. The definition is usually indented in the definition list.

Ex

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

declare

Sets or retrieves a character string that can be used to implement your own declare functionality for the object.

Syntax

HTML

<object DECLARE ... >

Scripting

object.declare(*v*) [= *sDec*]

Applies To

object

decodeURI() [javaScript]

The decodeURI() function decodes a URI encoded with the encodeURI() function.

Syntax

decodeURI(*sURI*)

Parameters

<i>sURI</i>	Required. The URI to be decoded
-------------	---------------------------------

Ex

```
<script type="text/javascript">
var test1="http://www.w3schools.com/My first/"
document.write(encodeURIComponent(test1)+" <br />")
document.write(decodeURI(test1))
</script>
```

The output of the code above will be:

http://www.w3schools.com/My%20first/

http://www.w3schools.com/My first/

Applies to

all of the built-in JavaScript objects

decodeURIComponent()

[JavaScript]

The decodeURIComponent() function decodes a URI encoded with the encodeURIComponent() function.

Syntax

decodeURIComponent(*sURI*)

Parameters

<i>sURI</i>	Required. The URI to be decoded
-------------	---------------------------------

Ex

```
<script type="text/javascript">
var test1="http://www.w3schools.com/My first/"
document.write(encodeURIComponent(test1)+ "<br />")
document.write(decodeURIComponent(test1))
</script>
```

The output of the code above will be:
http%3A%2F%2Fwww.w3schools.com%2FMy%20first%2F
http://www.w3schools.com/My first/

Applies to

all of the built-in JavaScript objects

defaultCharset [ie]

Retrieves the default character set from the current regional language settings.

Syntax

[*sCharset* =] *document.defaultCharset*

Applies To

document

defaultChecked

Sets or retrieves the state of the check box or radio button.

Syntax

HTML

N/A

Scripting

object.defaultChecked [= *bChecked*]

bChecked values

false	Check box or radio button is not selected by default
true	Default. Check box or radio button is selected by default

Applies To

input type=checkbox, input type=radio

defaults

Programmatically sets default properties on an element behavior.

Objects

style

Style Properties and attributes

accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, top, unicodeBidi, verticalAlign, visibility, zIndex

defaultSelected

Sets or retrieves the status of the option.

Syntax

HTML

N/A

Scripting

option.defaultSelected [= *bSelected*]

bSelected values

false	Option is not selected by default.
true	Default. Option is selected by default.

Applies To

option

defaultStatus [ie]

Sets or retrieves the default message displayed in the status bar at the bottom of the window.

Syntax

window.defaultStatus [= *sMessage*]

Applies To

window

defaultValue

Sets or retrieves the initial contents of the object.

Syntax

HTML

N/A

Scripting

object.defaultValue [= *sValue*]

Applies To

input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, textarea

defer

Sets or retrieves the status of the script.

Syntax

HTML

<script DEFER ... >

Scripting

script.defer [= *bDefer*]

sDefer values

| | |
|-------|---|
| false | Default. Inline executable function is not deferred |
| true | Inline executable function is deferred |

Applies To

script

Indicates text that has been deleted from the document.

Renders with a strikethrough line.

Ex

A contravento HUM! primeira segunda edicao

Optional Attributes

cite, datetime

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, width, wordWrap, writingMode, zoom

delete

The delete operator deletes an object, an object's property, or an element at a specified index in an array. You can use the delete operator to delete variables declared implicitly but not those declared with the var statement. If the delete operator succeeds, it sets the property or element to undefined. The delete operator returns true if the operation is possible; it returns false if the operation is not possible.

Syntax

delete *objectName*
delete *objectName.property*
delete *objectName[index]*
delete *property* // legal only within a with statement

Parameters

| | |
|------------|--|
| objectName | The name of an object. |
| property | The property to delete. |
| index | An integer representing the array index to delete. |

Ex

The fourth form is legal only within a with statement, to delete a property from an object.
x=42
var y= 43
myobj=new Number()
myobj.h=4 // create property h
delete x // returns true (can delete if declared implicitly)
delete y // returns false (cannot delete if declared with var)
delete Math.PI // returns false (cannot delete predefined properties)
delete myobj.h // returns true (can delete user-defined properties)
delete myobj // returns true (can delete objects)

Deleting array elements
When you delete an array element, the array length is not affected. For example, if you delete a[3], a[4] is still a[4] and a[3] is undefined. In the following example, trees[3] is removed with delete.

```
trees=new Array("redwood","bay","cedar","oak","maple")
delete trees[3]
if (3 in trees) {
    // this does not get executed
}
```

If you want an array element to exist but have an undefined value, use the undefined keyword instead of the delete operator. In the following example, trees[3] is assigned the value undefined, but the array element still exists:
trees=new Array("redwood","bay","cedar","oak","maple")
trees[3]=undefined
if (3 in trees) {
 // this gets executed
}

deleteCaption()

Deletes the caption element and its contents from the table.

Syntax

table.deleteCaption()

Applies to
table

deleteCell()

Removes the specified cell (td) from the table row, as well as from the cells collection.

Syntax

tr.deleteCell([iIndex])

Parameters

| | |
|---------------|---|
| <i>iIndex</i> | Optional. Integer that specifies the zero-based position of the cell to remove from the table row. If no value is provided, the last cell in the cells collection is deleted. |
|---------------|---|

Applies to
tr

deleteData()

Removes a specified range of characters from the object.]

Syntax

object.deleteData(nOffset, nCount)

Parameters

| | |
|----------------|--|
| <i>nOffset</i> | Required. Integer that specifies the offset from which to start. |
| <i>nCount</i> | Required. Integer that specifies the number of characters to remove. |

Applies to
comment, TextNode

deleteRow()

Removes the specified row (tr) from the element and from the rows collection.

Syntax

object.deleteRow([iRowIndex])

Parameters

| | |
|------------------|---|
| <i>iRowIndex</i> | Optional. Integer that specifies the zero-based position in the rows collection of the row to remove. |
|------------------|---|

Applies to
table, tbody, tfoot, thead

deleteTFoot()

Deletes the tFoot element and its contents from the table.

Syntax

object.deleteTFoot()

Applies to
table, tbody

deleteTHead()

Deletes the tHead element and its contents from the table.

Syntax

object.deleteTHead()

Applies to
table, tbody

designMode [ie]

Sets or retrieves a value that indicates whether the document can be edited.

Syntax

document.designMode [= *sMode*]

sMode values

| | |
|----------------|-------------------------------------|
| On | Document can be edited |
| off
inherit | Default. Document cannot be edited. |

Applies To
document

detachEvent() [ie]

Unbinds the specified function from the event, so that the function stops receiving notifications when the event fires.

Syntax

object.detachEvent(sEvent, fpNotify)

Parameters

| | |
|-----------------|--|
| <i>sEvent</i> | Required. String that specifies any of the standard DHTML Events. |
| <i>fpNotify</i> | Required. Pointer that specifies the function previously set using the attachEvent method. |

Applies to
a, abbr, acronym, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input

type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, namespace, nobr, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, window, xmp

deviceXDPI [ie]

Retrieves the actual number of horizontal dots per inch (DPI) of the system's screen.

Syntax

[nActDPI =] screen.deviceXDPI

Applies To

screen

deviceYDPI [ie]

Retrieves the actual number of vertical dots per inch (DPI) of the system's screen.

Syntax

[nActDPI =] screen.deviceYDPI

Applies To

screen

<dfn>

Indicates the defining instance of a term. Render italics.

Ex

<dfn>Definition term</dfn>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

dialogArguments [ie]

Retrieves the variable or array of variables passed into the modal dialog window.

Syntax

[vVariables =] window.dialogArguments

Applies To

window

dialogHeight [ie]

Sets or retrieves the height of the modal dialog window.

Syntax

window.dialogHeight [= sHeight]

Applies To

window

Dialog Helper

Provides access the color dialog box, as well as the block format and fonts collections. To create this object, use the object element and provide the class identifier (CLSID) for the dialog helper.

Ex

<object id=dlgHelper CLASSID="clsid:3050f819-98b5-11cf-bb82-00aa00bdce0b" width="0px" height="0px">

</object>

Collection

blockFormats

fonts

Methods

ChooseColorDlg, getCharSet

dialogLeft [ie]

Sets or retrieves the left coordinate of the modal dialog window.

Syntax

window.dialogLeft [= sLeft]

Applies To

window

dialogTop [ie]

Sets or retrieves the top coordinate of the modal dialog window.

Syntax

window.dialogTop [= sTop]

Applies To

window

dialogWidth [ie]

Sets or retrieves the width of the modal dialog window.

Syntax

window.dialogWidth [= sWidth]

Applies To

window

<dir>

Denotes a directory list. Deprecated in HTML 4.01, not supported in XHTML 1.0 Strict DTD, use CSS

Ex

<dir>

html

xhtml

</dir>

Optional Attributes

compact

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

dir

(a, abbr...)

Sets or retrieves the reading order of the object.

Syntax

HTML

<ELEMENT DIR = *sDir*... >

Scripting

object.dir [= *sDir*]

sDir values

| | |
|-----|--------------------------------------|
| ltr | Default. Content flows left to right |
| rtl | Content flows right to left |

Applies To

a, abbr, acronym, address, area, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

dir

(document)

Sets or retrieves a value that indicates the reading order of the object.

Syntax

document.dir [= sDir]

sDir values

| | |
|-----|--------------------------------------|
| ltr | Default. Content flows left to right |
| rtl | Content flows right to left |

Applies To

document

direction

(css)

Sets or retrieves the direction in which the text should scroll.

Syntax

CSS

{ direction : *sDirection* }

Scripting

object.style.direction(*v*) [= *sDirection*]

sDirection values

| | |
|-----|--------------------------------------|
| ltr | Default. Content flows left to right |
| rtl | Content flows right to left |

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div,

dl, dt, em, embed, fieldset, font, form, hn, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

direction

(marquee)

Sets or retrieves the reading order of the object.

Syntax

HTML

<marquee DIRECTION = *sDirection*... >

Scripting

marquee.direction [= *sDirection*]

sDirection values

| | |
|-------|-------------------------------|
| left | Default. Marquee scrolls left |
| right | Marquee scrolls right |
| down | Marquee scrolls down |
| up | Marquee scrolls up |

Applies To

marquee

disabled

Sets or retrieves a value that you can use to implement your own disabled functionality for the object.

Syntax

HTML

<*ELEMENT* DISABLED ... >

Scripting

object.disabled(*v*) [= *sDisabled*]

sDisabled values

| | |
|-------|---------------------------------|
| false | Default. The element is enabled |
| true | The element is disabled |

Applies To

option, OPTGROUP

disabled

Sets or retrieves the value that indicates whether the user can interact with the object.

Syntax

HTML

N/A

Scripting

object.disabled(*v*) [= *p*]

p values

| | |
|-------|-------------------------|
| false | The element is enabled |
| true | The element is disabled |

Applies To

area, basefont, bgsound, body, br, button, caption, col, comment, dd, div, dt, embed, font, form, frame, head, hr, html, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, label, legend, li, map, marquee, meta, object, ol, p, script, select, span, table, td, TEXT, textarea, title, tr, UL

disabled

Sets or retrieves the status of the object.

Syntax

HTML

<*ELEMENT* DISABLED ... >

Scripting

object.disabled [= *bDisabled*]

bDisabled values

| | |
|-------|----------------------------------|
| false | Default. Element is not disabled |
| true | Element is disabled |

Applies To

defaults, a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, strike, strong, STYLE, sub, sup, textarea, tt, u, ul, var, xmp

disabled

Sets or retrieves whether a style sheet is applied to the object.

Syntax

HTML

N/A

Scripting

object.disabled [= bDisabled]

bDisabled values

| | |
|-------|---|
| false | Default. Style sheet is applied to the object |
| true | Style sheet is not applied to the object |

Applies To

link, STYLE, styleSheet

display

Sets or retrieves whether the object is rendered.

Syntax

CSS

{ display : sDisplay }

Scripting

object.style.display [= sDisplay]

sDisplay values

| | |
|--------------------|--|
| block | Object is rendered as a block element. |
| none | Object is not rendered. |
| inline | Default. Object is rendered as an inline element sized by the dimensions of the content. |
| inline-block | Object is rendered inline, but the contents of the object are rendered as a block element. Adjacent inline elements are rendered on the same line, space permitting. |
| list-item | Microsoft Internet Explorer 6 and later. Object is rendered as a block element, and a list-item marker is added. |
| table-header-group | Table header is always displayed before all other rows and row groups, and after any top captions. The header is displayed on each page spanned by a table. |
| table-footer-group | Table footer is always displayed after all other rows and row groups, and before any bottom captions. The footer is displayed on each page spanned by a table. |

The default value for all elements is inline except for address, blockquote, body, center, col, colgroup, dd, dir, div, dl, dt, fieldset, form, th, hn, hr, iframe, legend, listing, marquee, menu, ol, p, plaintext, pre, table, tr, td, ul, xmp, that have a default value of "block"; tfoot, tbody, thead, frame, that have a default value of "none" and li which default value is "list-item"

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

<div>

Specifies a container that renders HTML.

Ex

This is some text

```
<div style="color:#FF0000;">
<h4>This is a header in a div section</h4>
<p>This is a paragraph in a div section</p>
</div>
```

Optional Attributes

align (deprecated)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, doScroll, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

<dl>

Denotes a definition list.

Ex

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects
currentStyle, runtimeStyle, style

Style Properties and attributes
:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

do ... while

The do... while statement creates a loop that executes a specified statement until the test condition evaluates to false. The condition is evaluated after executing the statement, resulting in the specified statement executing at least once.

Syntax

do
 statement
while (*condition*);

Parameters

| | |
|------------------|---|
| <i>statement</i> | A statement that is executed at least once and is re-executed each time the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ ... }) to group those statements. |
| <i>condition</i> | An expression evaluated after each pass through the loop. If condition evaluates to true, the statement is re-executed. When condition evaluates to false, control passes to the statement following the do...while. |

Ex

n the following example, the do...while loop iterates at least once and reiterates until i is no longer less than 5.

```
do {  
    i += 1;  
    document.write(i);  
} while (i < 5);
```

doctype

Retrieves the document type declaration associated with the current document.

Syntax

[*oDocType* =] *document.doctype*

oDocType values

| | |
|-----------------|--|
| <i>oDocType</i> | Object that receives the document type declaration |
| NULL | Default. Return value for HTML documents |

Applies To

document

document

Represents the HTML document in a given browser window.

Ex

This example uses the document object to check for a document title and displays the title (if not null) in a message box:

```
if (document.title!="")  
    alert("The title is " + document.title)
```

Properties

| | |
|---------------|---|
| activeElement | Retrieves the object that has the focus when the parent document has focus. |
|---------------|---|

| | |
|------------------|--|
| alinkColor | Sets or retrieves the color of all active links in the document. |
| bgColor | Deprecated. Sets or retrieves a value that indicates the background color behind the object. |
| charset | Sets or retrieves the character set used to encode the object. |
| compatMode | Retrieves a value that indicates whether standards-compliant mode is switched on for the object. |
| cookie | Sets or retrieves the string value of a cookie. |
| defaultCharset | Retrieves the default character set from the current regional language settings. |
| designMode | Sets or retrieves a value that indicates whether the document can be edited. |
| dir | Sets or retrieves a value that indicates the reading order of the object. |
| doctype | Retrieves the document type declaration associated with the current document. |
| documentElement | Retrieves a reference to the root node of the document. |
| domain | Sets or retrieves the security domain of the document. |
| expando | Sets or retrieves a value indicating whether arbitrary variables can be created within the object. |
| fgColor | Sets or retrieves the foreground (text) color of the document. |
| fileCreatedDate | Retrieves the date the file was created. |
| fileModifiedDate | Retrieves the date the file was last modified. |
| fileSize | Retrieves the file size. |
| implementation | Retrieves the implementation object of the current document. |
| lastModified | Retrieves the date the page was last modified, if the page supplies one. |
| linkColor | Sets or retrieves the color of the document links. |
| parentWindow | Retrieves a reference to the container object of the window. |
| protocol | Sets or retrieves the protocol portion of a URL. |
| readyState | Retrieves a value that indicates the current state of the object. |
| referrer | Retrieves the URL of the location that referred the user to the current page. |
| uniqueID | Retrieves an autogenerated, unique identifier for the object. |
| URL | Sets or retrieves the URL for the current document. |
| URLUnencoded | Retrieves the URL for the document, stripped of any character encoding. |
| vlinkColor | Sets or retrieves the color of the links that the user has visited. |
| XMLDocument | Retrieves a reference to the Extensible Markup Language (XML) |
| XSLDocument | Retrieves a reference to the top-level node of the Extensible Stylesheet Language (XSL) document. |

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, anchors, applets, childNodes, embeds, forms, frames, images, links, namespaces, scripts, styleSheets

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

attachEvent, clear, close, createAttribute, createComment, createDocumentFragment, createElement, createEventObject, createStyleSheet, createTextNode, detachEvent, elementFromPoint, execCommand, focus, getElementById, getElementsByName, getElementsByTagName, hasFocus, mergeAttributes, open, queryCommandEnabled, queryCommandIndeterm, queryCommandState, queryCommandSupported, queryCommandValue, recalc, releaseCapture, setActive, write, writeln

Applies To
custom, window

document [ie]

Retrieves the HTML document in a given popup window.

Syntax

[*oDocument* =] *popup*.document

Applies To
popup

documentElement

Retrieves a reference to the root node of the document.

Syntax

[*oElement* =] *document*.documentElement

Applies To
document

doImport() [ie]

Dynamically imports an element behavior.

Syntax

namespace.doImport(*sUrl*)

Parameter

| | |
|-------------|---|
| <i>sUrl</i> | Required. String that specifies the URL of the element behavior to import into the namespace. |
|-------------|---|

Applies to
namespace

domain

Sets or retrieves the security domain of the document (domain suffix).

Syntax

document.domain [= *sDomain*]

Applies To
document

doReadRequest() [ie]

Performs all requests located in the read-requests queue.

Syntax

userProfile.doReadRequest(*vUsageCode* [, *vFriendlyName*] [, *vDomain*] [, *vPath*] [, *vExpiration*] [, *vReserved*])

Parameters

| | |
|-------------------|---|
| <i>vUsageCode</i> | Required. Integer that specifies the code that notifies the user of the type of access requested. This usage code should be one of the following 13 codes defined by the Internet Privacy Working Group (IPWG). |
| 0 | Used for system administration. |
| 1 | Used for research and product development. |
| 2 | Used for completion and support of the current transaction. |
| 3 | Used to customize the content and design of a site. |
| 4 | Used to improve the content of a site that includes advertisements. |
| 5 | Used for notifying visitors about updates to the site. |
| 6 | Used for contacting visitors for marketing of services or products. |
| 7 | Used for linking other collected information. |
| 8 | Used by a site for other purposes. |
| 9 | Disclosed to others for customization or improvement of the content and design of the site. |
| 10 | Disclosed to others, who may contact the user, for marketing of services or products. |
| 11 | Disclosed to others, who may contact the user, for marketing of services or products. The user can ask a site not to do this. |
| 12 | Disclosed to others for any other purpose. |

| | |
|----------------------|---|
| <i>vFriendlyName</i> | Optional. String that specifies the display name of the party requesting access to private information. For security reasons, it is not sufficient for the user agent to show this display name to the user. In addition to display name, the user agent must display the URL that originates the script requesting profile access. If this script originates from a secure connection (for example, SSL), the Web author can use the SSL certificate to reliably identify the party requesting access. |
| Domain | Optional. String that specifies the pages that the user's choice will apply to in addition to the current page. The specification follows the cookie standard. |
| <i>vPath</i> | Optional. String that specifies the path to the domain server requesting access. When <i>vExpiration</i> is set, the path is saved with the requested attributes. |
| <i>vExpiration</i> | Optional. Variant that specifies the amount of time the site has requested access to these attributes. This is currently ignored by Microsoft Internet |

Applies to
userProfile

doScroll() [ie]

Simulates a click on a scroll-bar component.

Syntax

object.doScroll([sScrollAction])

Parameters

| | |
|----------------------|--|
| <i>sScrollAction</i> | Optional. String that specifies how the object scrolls, using one of the following values: |
| scrollbarDown | Default. Down scroll arrow is at the specified location. |
| scrollbarLeft | Left scroll arrow is at the specified location. |
| scrollbarHThumb | Horizontal scroll thumb or box is at the specified location. |
| scrollbarPageDown | Page-down scroll bar shaft is at the specified location. |
| scrollbarPageLeft | Page-left scroll bar shaft is at the specified location. |
| scrollbarPageRight | Page-right scroll bar shaft is at the specified location. |
| scrollbarPageUp | Page-up scroll bar shaft is at the specified location. |
| scrollbarRight | Right scroll arrow is at the specified location. |
| scrollbarUp | Up scroll arrow is at the specified location. |
| scrollbarVThumb | Vertical scroll thumb or box is at the specified location. |
| down | Composite reference to scrollbarDown. |
| left | Composite reference to scrollbarLeft. |
| pageDown | Composite reference to scrollbarPageDown. |
| pageLeft | Composite reference to scrollbarPageLeft. |
| pageRight | Composite reference to scrollbarPageRight. |
| pageUp | Composite reference to scrollbarPageUp. |
| right | Composite reference to scrollbarRight. |
| up | Composite reference to scrollbarUp. |

Applies to
body, custom, div, span, textarea

dragDrop() [ie]

Initiates a drag event. Returns a boolean value corresponding to the success of the drag operation

Syntax

bDragOK = *object*.dragDrop()

Applies to
a, area, basefont, bgsound, body, br, button, caption, col, comment, dd, div, dt, embed, font, form, frame, head, hr, html, iframe, img, input type=button, input

type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, label, legend, li, map, marquee, meta, object, ol, option, p, script, select, span, STYLE, table, td, TEXT, textarea, title, tr, UL

dropEffect [ie]

Sets or retrieves the type of drag-and-drop operation and the type of cursor to display.

Syntax

dataTransfer.dropEffect [= *sCursorStyle*]

sCursorStyle values

| | |
|------|--|
| copy | Copy cursor is displayed. |
| link | Link cursor is displayed. |
| move | Move cursor is displayed. |
| none | Default. No cursor is specified. Instead, the no-drop cursor is displayed. |

Applies To

dataTransfer

<dt>

Indicates a definition term within a definition list.

Ex

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, styleFloat, textAlign, textAlignLast, textAutospace, textDecor, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

duplicate() [ie]

Returns a duplicate of the TextRange. Returns a TextRange object. Might not be available on non-Microsoft Win32

platforms

Syntax

oTextRange = *TextRange*.duplicate()

Applies to

TextRange

dynsrc [ie]

Sets or retrieves the address of a video clip or VRML world to display in the window.

Syntax

HTML

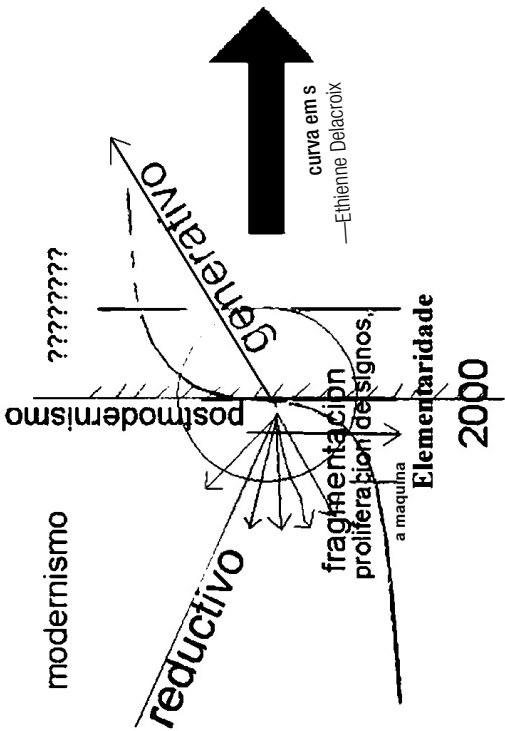
<*ELEMENT* DYN SRC = *sURL*... >

Scripting

object.dynsrc [= *sURL*]

Applies To

img, input, input type=image



E [javascript]

Returns Euler's constant (approx. 2.718)

Applies To
Math

effectAllowed [ie]

Sets or retrieves, on the source element, which data transfer operations are allowed for the object.

Syntax

dataTransfer.effectAllowed(*v*) [= *sEffect*]

sEffect values

| | |
|---------------|---|
| copy | Selection is copied. |
| link | Selection is linked to the drop target by the data transfer operation. |
| move | Selection is moved to the target location when dropped. |
| copyLink | Selection is copied or linked, depending on the target default. |
| copyMove | Selection is copied or moved, depending on the target default. |
| linkMove | Selection is linked or moved, depending on the target default. |
| all | All drop effects are supported. |
| none | Dropping is disabled and the no-drop cursor is displayed. |
| uninitialized | Default. No value has been set through the effectAllowed property. In this case, the default effect still works, although it cannot be queried through this property. |

Applies To
dataTransfer

elementFromPoint() [ie]

Returns the element for the specified x and y coordinates.
Returns an element object

Syntax

oElement = *document*.elementFromPoint(*iX*, *iY*)

Parameters

| | |
|-----------|---|
| <i>iX</i> | Required. Integer that specifies the X-offset, in pixels. |
| <i>iY</i> | Required. Integerthat specifies the Y-offset, in pixels. |

Applies to
document

elements

Retrieves a collection, in source order, of all controls in a given form. input type=image objects are excluded from the collection.

Syntax

[*oColl* =] FORM.elements

[*oObject* =] FORM.elements(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|------------------|---|
| <i>oColl</i> | Array of button, input, select, and textArea objects. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex. |

Applies To
FORM

Emphasizes text, usually by rendering it in italic.

Ex

Emphasized text

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

<embed>

Allows documents of any type to be embedded. The <embed> element is supported by both Internet Explorer and Netscape, but it is not a standard HTML or XHTML element. The World Wide Web Consortium (W3C) recommend using the <object> element instead.

Ex

<embed src="wow.mid" />

Standard Attributes

id, class, title, style, dir, lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clear, clip, cursor, direction, display, hasLayout, height, layoutGrid, layoutGridMode, left, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight,

maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAutospace, textUnderlinePosition, top, unicodeBidi, visibility, wordWrap, zoom

embeds [ie]

Retrieves a collection of all embed objects in the document.

Syntax

[*oColl* =] document.embeds

[*oObject* =] document.embeds(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|------------------|---|
| <i>oColl</i> | Array of embed objects. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when <i>vindex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> . |

Applies To
document

empty() [ie]

Cancels the current selection, sets the selection type to none, and sets the item property to null.

Syntax

selection.empty()

Applies to
selection

encoding

Sets or retrieves the MIME encoding for the form.

Syntax

HTML

<FORM ENCTYPE = *sType*... >

Scripting

FORM.encoding [= *sType*]

Applies To
FORM

encodeURIComponent() [JavaScript]

The encodeURIComponent() function encodes a string as a URI.

Syntax

encodeURIComponent(*sURI*)

Parameter

| | |
|-------------|---------------------------------|
| <i>sURI</i> | Required. The URI to be encoded |
|-------------|---------------------------------|

Ex

```
<script type="text/javascript">
document.write(encodeURIComponent("http://www.w3schools.com")+ "<br />")
document.write(encodeURIComponent("http://www.w3schools.com/My first/"))
document.write(encodeURIComponent(", /?:@&=+$#"))
</script>
```

The output of the code above will be:
http://www.w3schools.com
http://www.w3schools.com/My%20first/
, /?:@&=+\$#

Applies to

all of the built-in JavaScript objects

encodeURIComponent()

[JavaScript]

The encodeURIComponent() function encodes a string as a component of a URI.

Syntax

encodeURIComponent(*sURI*)

Parameter

| | |
|-------------|---------------------------------|
| <i>sURI</i> | Required. The URI to be encoded |
|-------------|---------------------------------|

Ex

```
<script type="text/javascript">
document.write(encodeURIComponent("http://www.w3schools.com"))
document.write("<br />")
document.write(encodeURIComponent("http://www.w3schools.com/p 1/"))
document.write("<br />")
document.write(encodeURIComponent(", /?:@&=+$#"))
</script>
```

The output of the code above will be:
http%3A%2F%2Fwww.w3schools.com
http%3A%2F%2Fwww.w3schools.com%2Fp%201%2F
%2C%2F%3F%3A%40%26%3D%2B%24%23

Applies to
all of the built-in JavaScript objects

enctype

Sets or retrieves the Multipurpose Internet Mail Extensions (MIME) encoding for the form.

Syntax

HTML

<FORM ENCTYPE = *sType*... >

Scripting

FORM.enctype(v) [= *sType*]

sType values

The *sType* default value is "application/x-www-form-urlencoded"

Applies To
FORM

escape() [JavaScript]

The escape() function encodes a string, so it can be read on all computers. The escape() function encodes special characters, with the exception of:

* @ - _ + . /

Syntax

escape(*string*)

Parameter

| | |
|---------------|------------------------------------|
| <i>string</i> | Required. The string to be encoded |
|---------------|------------------------------------|

Ex

```
<script type="text/javascript">
document.write(escape("Visit W3Schools!") + "<br />")
document.write(escape("'?!=()#%&"))
</script>
```

The output of the code above will be:
Visit%20W3Schools%21
%3F%21%3D%28%29%23%25%26

Applies to
all of the built-in JavaScript objects

eval() [JavaScript]

The eval() function evaluates a string and executes it as if it was script code.

Syntax

eval(*string*)

Parameter

| | |
|---------------|--------------------------------------|
| <i>string</i> | Required. The string to be evaluated |
|---------------|--------------------------------------|

Ex

```
<script type="text/javascript">
eval("x=10;y=20;document.write(x*y)")
document.write("<br />")
document.write(eval("2+2"))
document.write("<br />")
```

```
var x=10
document.write(eval(x+17))
document.write("<br />")
</script>
The output of the code above will be:
200
4
27
```

Applies to
all of the built-in JavaScript objects

event

Represents the state of an event, such as the element in which the event occurred, the state of the keyboard keys, the location of the mouse, and the state of the mouse buttons.

Ex

```
This example uses the event object to check whether the user clicked the mouse within a link, and to prevent the link from being navigated if the SHIFT key is down.
<HTML>
<head><title>Cancels Links</title>
<script LANGUAGE="JScript">
function cancelLink() {
    if (window.event.srcElement.tagName == "A" && window.event.shiftKey)
        window.event.returnValue = false;
}
</script>
<body onclick="cancelLink()">
```

```
This example displays the current mouse position in the browser's status window:
<body onmousemove="window.status = 'X=' + window.event.x + ' Y=' + window.event.y">
```

- Properties**
Description, Abstract, altKey, altLeft, Banner, button, cancelButton, clientX, clientY, contentOverflow, ctrlKey, ctrlLeft, dataFld, fromElement, keyCode, MoreInfo, nextPage, offsetX, offsetY, propertyName, qualifier, reason, recordset, repeat, returnValue, saveType, screenX, screenY, shiftKey, shiftLeft, srcElement, srcFilter, srcUrn, toElement, type, userName, wheelDelta, x, y
- Collection**
bookmarks, boundElements
- Object**
dataTransfer

event

Sets or retrieves the event for which the script is written.

Syntax
HTML
<script EVENT = *sEvent*... >
Scripting
script.event [= *sEvent*]

sEvent values
String that specifies or receives the event. The value for an onmouseover event, for example, is "onmouseover()"

Applies To
script

execCommand() [ie]

Executes a command on the current document, current selection, or the given range. Returns True if the command is successful

Syntax
bSuccess = *object*.execCommand(*sCommand* [,*bUserInterface*] [,*vValue*])

Parameter

| | |
|-----------------|--|
| <i>sCommand</i> | Required. String that specifies the command to execute. This command can be any of the command identifiers that can be executed in script. |
|-----------------|--|

| | |
|-----------------------|---|
| <i>bUserInterface</i> | Optional. Boolean that specifies one of the following values:
false: Default. Does not display a user interface; true: Displays a user interface, if the command supports one. |
| <i>vValue</i> | Optional. Variant that specifies the string, number, or other value to assign. Possible values depend on <i>sCommand</i> . |

Applies to
document, controlRange, TextRange

execScript() [ie]

Executes the specified script in the provided language.

Syntax
vReturn = *window*.execScript(*sExpression*, *sLanguage*)

Parameter

| | |
|--------------------|--|
| <i>sExpression</i> | Required. Stringthat specifies the code to be executed. |
| <i>sLanguage</i> | Required. Stringthat specifies the language in which the code is executed. The language defaults to Microsoft JScript. |

Applies to
window

exp() [JavaScript]

The exp() method returns the value of E^x, where E is Euler's constant (approximately 2.7183) and x is the number passed to it.

Syntax
Math.exp(x)

Parameter

| | |
|----------|--------------------|
| <i>x</i> | Required. A number |
|----------|--------------------|

Ex

```
<script type="text/javascript">
document.write(Math.exp(1) + "<br />")
document.write(Math.exp(-1) + "<br />")
document.write(Math.exp(5) + "<br />")
document.write(Math.exp(10) + "<br />")
</script>
```

```
The output of the code above will be:
2.718281828459045
0.36787944117144233
148.4131591025766
22026.465794806718
```

Applies to
Math

expand() [ie]

Expands the range so that partial units are completely contained. Returns a boolean value

Syntax
bSuccess = *TextRange*.expand(*sUnit*)

Parameters

| | |
|--------------|---|
| <i>sUnit</i> | Required. Stringthat specifies the units to move in the range, using one one of the following values: |
| character | Expands a character. |
| word | Expands a word. A word is a collection of characters terminated by a space or another white-space character, such as a tab. |
| sentence | Expands a sentence. A sentence is a collection of words terminated by an ending punctuation character, such as a period. |
| textedit | Expands to enclose the entire range. |

Applies to
TextRange

expando [ie]

Sets or retrieves a value indicating whether arbitrary variables can be created within the object.

Syntax
object.expando [= *bExpand*]

bExpand values

| | |
|-------|---|
| false | Creation of arbitrary variables is not allowed |
| true | Default. Creation of arbitrary variables is allowed |

Applies To

attribute, document

export

The export statement allows a signed script to provide properties, functions, and objects to other signed or unsigned scripts. This feature is not in ECMA-262, Edition 3.

The first syntax exports the specified properties, functions, and objects.

The second syntax exports all properties, functions, and objects from the script.

Syntax

```
export name1, name2, ..., nameN;
```

```
export *;
```

Parameters

| | |
|--------------|---|
| <i>nameN</i> | Property, function, or object to be exported. |
|--------------|---|

external

Allows access to an additional object model provided by host applications of the Microsoft Internet Explorer browser components.

Property

| Property | Description |
|---------------|---|
| menuArguments | Returns the window object where the context menu item was executed. |

Methods

AddChannel, AddDesktopComponent, AddFavorite, AddSearchProvider [New for IE 7], AutoCompleteSaveForm, AutoScan, BrandImageUri [New for IE 7], CustomizeSettings [New for IE 7], ImportExportFavorites, IsSubscribed, NavigateAndFind, PhishingEnabled [New for IE 7], RunOnceShown [New for IE 7], ShowBrowserUI, SkipRunOnce [New for IE 7], SqmEnabled [New for IE 7]

ORDEM Nº2 AO EXÉRCITO DAS ARTES

A vós
—barítonos redondos—
cuja voz
desde Adão até a nossa era

nos atros buracos chamados teatros
estronda o ribombo lírico de árias.

A vós
—pintores—
cavalos cevados,

rumino-relinchante galardão eslavo,
no fundo dos estúdios, cediços como dragos,
pintando anatomias e quadros de flores.

A vós
rugas na testa entre fólhos de mística

—micro-futurista,
—imagista,
—acméistas—
emaranhados no aranhol das rimas.

A vós—
descabelando cabelos bem-penteados,
barganhando escarpins por solados,
vates do Proletcult,
remendões do fraque velho de Púchkin.

A vós—
bailadores, sopradores de flauta,
amolecendo às claras
ou em furtivas faltas,
e figurando o futuro nos termos
de um imenso quinhão acadêmico.
A vós todos

eu—
que acabei com berloques e dou duro na Rosta—
gênio ou não gênio, tenho
a dizer: basta!
Abaixo com isso,
antes que vos abata o coice dos fuzis.

Basta!
Abaixo,
cuspi
no rimário,
nas árias,

nos róseos açafates
e mais minincolias
do arsenal das artes.
Quem se interessa
por ninharias
como estas: “Ah pobre coitado!”

Quanto amou sem ter sido amado...?
Artífices,
é o que o tempo exige,
e não sermonistas de juba.
Ouvi
o gemido das locomotivas,

que lufa das frinchas, do chão:
“Dai-nos, companheiros,
carvão do Don!
ao depósito, vamos,
serralheiros,
mecânicos!”

À nascente dos rios,
deitados com furos nas costas,
—Petróleo de Baku!—pedem navios
uivando nas docas.

Perdidos em disputas monótonas,

buscamos o sentido secreto,
quando um clamor sacode os objetivos:
“Dai-nos novas formas!”

Não há mais tolos boquiabertos,
esperando a palavra do “mestre”.

“Dai-nos, camaradas, uma arte nova
—nova—
que arranque a república da escória.

Vladimir Maiakovsky
—tradução de Haroldo de Campos

face

Sets or retrieves the current typeface family. Deprecated.

Syntax

HTML

<ELEMENT FACE = *sTypeface*... >

Scripting

object.face [= *sTypeface*]

Applies To

basefont, FONT

fgColor [ie]

Sets or retrieves the foreground (text) color of the document.

Syntax

document.fgColor(*v*) [= *vColor*]

Applies To

document

FieldDelim [ie]

Specifies the character used to mark the end of data fields. The default value is comma (,.)

Syntax

TDC.FieldDelim [= *sValue*]

Applies To

TDC

<fieldSet>

Draws a box around the text and other elements that the field set contains.

Ex

<fieldset>

name <input type="text" size="40" />

</fieldset>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-letter, :first-line, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

fileCreatedDate [ie]

Retrieves the date the file was created.

Syntax

HTML

N/A

Scripting

[*sDate* =] *object*.fileCreatedDate

Applies To

document, img

fileModifiedDate [ie]

Retrieves the date the file was last modified.

Syntax

HTML

N/A

Scripting

[*sDate* =] *object*.fileModifiedDate

Applies To

document, img

fileSize [ie]

Retrieves the file size.

Syntax

HTML

N/A

Scripting

[*iSize* =] *object*.fileSize

Applies To

document, img

fileUpdatedDate [ie]

Retrieves the date the file was last updated.

Syntax

N/A

Scripting

[*sDate* =] *img*.fileUpdatedDate

Applies To

img

filter [ie]

Sets or retrieves the filter or collection of filters applied to the object. (Part of a proposed addition to Cascading Style Sheets (CSS))

Syntax

CSS

{ filter : *sFilter* }

Scripting *object*.style.filter [= *sFilter*]

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, hn, iframe, frameset, i, ins, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, marquee, menu, nobr, ol, p, plaintext, pre, q, rt, ruby, s, samp, small, span, strike, strong, sub, sup, table, textarea, th, td, tt, u, ul, var, xmp

filters [ie]

Retrieves the collection of filters that have been applied to the object.

Syntax

[*oColl* =] *object*.filters

[*oObject* =] *object*.filters(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|----------------|---|
| <i>oColl</i> | Array of filters applied to the object. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |



| | |
|------------------|---|
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex. |

Applies To

bdo, body, button, custom, div, fieldset, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, marquee, nextid, rt, ruby, span, table, td, textarea, th

findText() [ie]

Searches for text in the document and positions the start and end points of the range to encompass the search string. Returns a boolean value

Syntax

bFound = *TextRange.findText(sText* [,*iSearchScope*] [,*iFlags*])

Parameter

| | |
|---------------------|--|
| <i>sText</i> | Required. String that specifies the text to find. |
| <i>iSearchScope</i> | Optional. Integer that specifies the number of characters to search from the starting point of the range. A positive integer indicates a forward search; a negative integer indicates a backward search. |
| <i>iFlags</i> | Optional. Integer that specifies one or more of the following flags to indicate the type of search: |
| 0 | Default. Match partial words. |
| 1 | Match backwards. |
| 2 | Match whole words only. |
| 4 | Match case. |
| 131072 | Match bytes. |
| 536870912 | Match diacritical marks. |
| 1073741824 | Match Kashida character. |
| 2147483648 | Match AlefHamza character. |

Applies to

TextRange

fireEvent() [ie]

Fires a specified event on the object. Returns a boolean value

Syntax

bFired = *object.fireEvent(sEvent* [,*oEventObject*])

Parameter

| | |
|---------------------|--|
| <i>sEvent</i> | Required. String that specifies the name of the event to fire. |
| <i>oEventObject</i> | Optional. Object that specifies the event object from which to obtain event object properties. |

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

firstChild

Retrieves a reference to the first child in the childNodes collection of the object.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object.firstChild*

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

firstPage() [ie]

Displays the first page of records in the data set to which the table is bound.

Syntax

table.firstPage()

Applies to

window

fixed() [JavaScript]

The fixed() method is used to display a string as teletype text.

Syntax

stringObject.fixed()

Ex

In this example "Hello world!" will be displayed as teletype text:

```
<script type="text/javascript">
var str="Hello world!"
document.write(str.fixed())
</script>
```

Applies to

String

float

Sets or retrieves on which side of the object the text will flow.

Syntax

CSS

{ float : *sFloat* }

Scripting

object.style.styleFloat [= *sFloat*]

sFloat values

| | |
|-------|---|
| none | Default. Object displays where it appears in the text |
| left | Text flows to the right of the object |
| right | Text flows to the left of the object |

Applies To

a, address, applet, b, big, blockquote, button, center, cite, code, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, pre, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

floor() [JavaScript]

The floor() method returns the value of a number rounded DOWNWARDS to the nearest integer.

Syntax

Math.floor(x)

Parameter

| | |
|----------|--------------------|
| <i>x</i> | Required. A number |
|----------|--------------------|

Ex

```
<script type="text/javascript">
document.write(Math.floor(0.60) + " , ")
document.write(Math.floor(0.40) + " , ")
document.write(Math.floor(5) + " , ")
document.write(Math.floor(5.1) + " , ")
document.write(Math.floor(-5.1) + " , ")
document.write(Math.floor(-5.9))
</script>
```

The output of the code above will be:
0, 0, 5, 5, -6, -6

Applies to
Math

focus()

Causes the element to receive the focus and executes the code specified by the onfocus event. As of Microsoft Internet Explorer 5, elements that expose the focus method must have the TABINDEX attribute set. Elements cannot receive focus until the document finishes loading

Syntax

object.focus()

Applies to

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, del, dfn, dir, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, window, xmp

Specifies a new font, size, and color to be used for rendering the enclosed text. The font element was deprecated in HTML 4.01and is not supported in XHTML 1.0 Strict DTD.

Ex

```
<font face="arial" color="red">
This is some text!
</font>
```

Optional Attributes

color, face, size

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, textAutospace, textOverflow, textUnderlinePosition, top, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

font

Sets or retrieves a combination of separate font properties of the object. Alternatively, sets or retrieves one or more of six user-preference fonts.

Syntax

{ font : *sFont* }

Scripting

object.style.font [= *sFont*]

sFont values

font-style	Any of the range of font-style values available to the fontStyle property.
font-variant	Any of the range of font-variant values available to the fontVariant property.
font-weight	Any of the range of font-weight values available to the fontWeight property.
font-size	Any of the range of font-size values available to the fontSize property. When this value is an integer followed by a percent (%), the value is a percentage of the parent object's font size. In Microsoft Internet Explorer 3.0, a percentage value is calculated as a percentage of the default font size.
line-height	Any of the range of line-height values available to the lineHeight property. When used with the font property, this attribute must include a slash (/) before the value. Line height percentage values are calculated as a percentage of the font size of the element itself, not of the parent.
font-family	Any of the range of font-family values available to the fontFamily property. This property can be set to multiple comma-separated values. Its default value depends on user settings.
caption	User-preference font used in objects that have captionsbut-tons, labels, and so on.
icon	User-preference font used in icon labels.
menu	User-preference font used in menus.
message-box	User-preference font used in dialog boxes.
small-caption	User-preference font used in small controls.
status-bar	User-preference font used in window status bars.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, nobr, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

fontcolor() [JavaScript]

The fontcolor() method is used to display a string in a specified color.

Syntax

stringObject.fontcolor(color)

Parameter

<i>color</i>	Required. Specifies a font-color for the string. The value can be a color name (red), an RGB value (rgb(255, 0, 0)), or a hex number (#FF0000)
--------------	--

Ex

In this example "Nem a paul!" will be displayed in red:

```
<script type="text/javascript">
var str="Nem a paul!"
document.write(str.fontcolor("Red"))
</script>
```

Applies to

String

fontFamily

Sets or retrieves the name of the font used for text in the object.

Syntax

CSS

{ font-family : *sFamily* }

Scripting

object.style.fontFamily [= *sFamily*]

sFamily values

<i>family-name</i>	Any of the available font families supported by the browser. For example, Times, Helvetica, Western, or Courier
<i>generic-name</i>	Any of the following font families: serif, sans-serif, cursive, fantasy, or monospace.

Applies To

a, address, b, basefont, big, blockquote, body, button, caption, center, cite,

code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

fonts [ie]

Retrieves a collection of all the system-supported fonts.

Syntax

[*oColl* =] Dialog Helper.fonts

[*oObject* =] Dialog Helper.fonts(*iIndex*)

Possible Values

oColl	A collection of system-supported fonts.
oObject	Reference to an individual item in the array of elements contained by the object.
iIndex	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

Dialog Helper

fontSize

Sets or retrieves a value that indicates the font size used for text in the object.

Syntax

CSS

{ font-size : *sSize* }

Scripting

object.style.fontSize [= *sSize*]

sSize values

<i>absolute-size</i>	Set of keywords that indicate predefined font sizes. Named font sizes scale according to the user's font setting preferences. Possible values include the following: xx-small, x-small, small, medium, large, x-large, xx-large.
<i>relative-size</i>	Set of keywords that are interpreted as relative to the font size of the parent object. Possible values include the following: larger, smaller.
<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	The value is a percentage of the parent object's font size

Applies To

a, address, applet, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=file, input type=image, input type=password, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

fontSize() [JavaScript]

The `fontSize()` method is used to display a string in a specified size.

Syntax

stringObject.fontSize(size)

Parameter

<i>size</i>	Required. A number that specifies the font size
-------------	---

Ex

```
<script type="text/javascript">
var str="Nem a pau!"
document.write(str.fontSize(7))
</script>
```

Applies to

String

fontSmoothingEnabled [ie]

Retrieves whether the user has enabled font smoothing in the Display control panel.

Syntax

[*bEnabled* =] *screen*.fontSmoothingEnabled

bEnabled values

false	Default. Font smoothing is disabled
true	Font smoothing is enabled

Applies To

screen

fontStyle

Sets or retrieves the font style of the object as italic, normal, or oblique.

Syntax

CSS

{ font-style : *sStyle* }

Scripting

object.style.fontStyle [= *sStyle*]

sStyle values

normal	Default. Font is normal
italic	Font is italic
oblique	Font is italic

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

fontVariant

Sets or retrieves whether the text of the object is in small capital letters.

Syntax

CSS

{ font-variant : *sVariant* }

Scripting

object.style.fontVariant [= *sVariant*]

sVariant values

normal	Default. Font is normal.
small-caps	Font is in small capital letters.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

fontWeight

Sets or retrieves the numeric weight of the font of the object.

Syntax

CSS

{ font-weight : *sWeight* }

Scripting

object.style.fontWeight [= *sWeight*]

[*nWeight* =] *object*.currentStyle.fontWeight

sWeight values

normal	Font is normal.
bold	Font is bold.
bolder	Font is at least as bold as the default bold weight.
lighter	Font is lighter than normal.
100	Font is at least as light as the 200 weight.
200	Font is at least as bold as the 100 weight and at least as light as the 300 weight.
300	Font is at least as bold as the 200 weight and at least as light as the 400 weight.

400	Font is normal.
500	Font is at least as bold as the 400 weight and at least as light as the 600 weight.
600	Font is at least as bold as the 500 weight and at least as light as the 700 weight.
700	Font is bold.
800	Font is at least as bold as the 700 weight and at least as light as the 900 weight.
900	Font is at least as bold as the 800 weight.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

for

The for statement creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement executed in the loop.

Syntax

for ([*initial-expression*]; [*condition*]; [*increment-expression*])

statement

Parameters

<i>initial-expression</i>	An expression (including assignment expressions) or variable declaration. Typically used to initialize a counter variable. This expression may optionally declare new variables with the var keyword. These variables are not local to the loop, i.e. they are in the same scope the for loop is in.
<i>condition</i>	An expression evaluated on each pass through the loop. If this condition evaluates to true, statement is executed. This conditional test is optional. If omitted, the condition always evaluates to true.
<i>increment-expression</i>	Generally used to update or increment the counter variable.
<i>statement</i>	A statement that is executed as long as the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ ... }) to group those statements.

Ex

The following for statement starts by declaring the variable i and initializing it to 0. It checks that i is less than nine, performs the two succeeding statements, and increments i by 1 after each pass through the loop:

```
for (var i = 0; i < 9; i++) {  
    n += i;  
    myfunc(n);  
}
```

for ... in

The for... in statement iterates a specified variable over all the properties of an object. For each distinct property, a specified statement is executed.

Syntax

for (*variable* in *object*)

statement

Parameters

<i>variable</i>	Variable to iterate over every property, optionally declared with the var keyword. This variable is local to the function, not to the loop.
<i>object</i>	Object for which the properties are iterated.
<i>statement</i>	A statement that is executed as long as the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ ... }) to group those statements.

Ex

```
function show_props(obj, objName) {  
    var result = "";  
    for (var i in obj) {
```

```
        result += objName + "." + i + " = " + obj[i] + "\n";  
    }  
    return result;  
}
```

for each... in

Iterates a specified variable over all values of object's properties. For each distinct property, a specified statement is executed.

Syntax

for each (*variable* in *object*)

statement

Parameters

<i>variable</i>	Variable to iterate over every property, optionally declared with the var keyword. This variable is local to the function, not to the loop.
<i>object</i>	Object for which the properties are iterated.
<i>statement</i>	A statement that is executed as long as the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ ... }) to group those statements.

Ex

```
var sum = 0;  
var arr = [5,7,9];  
for each (var item in arr) {  
    sum += item;  
}  
print(sum);
```

<form>

Specifies that the contained controls take part in a form. A form can contain textfields, checkboxes, radio-buttons and more.

Forms are used to pass user-data to a specified URL.

Using this object incorrectly can compromise the security of your application. Data submitted through a form using the HTTP protocol is not encrypted and can be read and possibly tampered with in transmission. The Secure Hypertext Transfer Protocol (HTTPS) can provide more secure data transmission.

Ex

```
<form action="form_action.asp"  
method="get">
```

First name:

```
<input type="text" name="fname" value="Tales" />  
<br />
```

Last name:

```
<input type="text" name="lname" value="Stolfi" />  
<br />
```

```
<input type="submit" value="Submit" />  
</form>
```

<p>

If you click the "Submit" button, you will send your input to a new page called form_action.asp.

</p>

Required Attributes

action (A URL that defines where to send the data when the submit button is pushed)

Optional Attributes

accept, accept-charset, enctype, method, name, target

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children, elements

Events

onsubmit, onreset, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore,

item, mergeAttributes, namedItem, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, reset, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, submit, swapNode, urns

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

form [ie]

Retrieves a reference to the form that the object is embedded in.

Syntax

HTML

N/A

Scripting

[*oForm* =] *object*.form

Applies To

button, fieldset, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, label, legend, object, option, select, textarea

forms

Retrieves a collection, in source order, of all form objects in the document.

Syntax

[*oColl* =] document.forms

[*oObject* =] document.forms(*vIndex* [, *iSubIndex*])

Possible Values

<i>oColl</i>	Array of form objects
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when vindex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To

document

forward() [ie]

Loads the next URL from the History list.

Syntax

history.forward()

Applies to

history

<frame>

Specifies an individual frame within a FRAMESET element.

In HTML the <frame> tag has no end tag and in XHTML the

<frame> tag must be properly closed.

Ex

View this page for the result

The source code in "planets.htm":

```
<html>
<frameset cols = "25%, 25%, *">
  <frame src = "venus.htm" />
  <frame src = "sun.htm" />
  <frame src = "mercur.htm" />
</frameset>
</html>
```

Optional Attributes

frameborder, longdesc, marginheight, marginwidth, name, noresize, scrolling, src

Standard Attributes

id, class, title, style (Only allowed in XHTML 1.0 Frameset DTD)

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setActive, setAttribute, setAttributeNode, swapNode

Objects

runtimeStyle, style

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, display, filter, height, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, zoom

frame

Sets or retrieves the way the border frame around the table is displayed.

Syntax

HTML

<table FRAME = *sFrame*... >

Scripting

table.frame [= *sFrame*]

sFrame values

void	Default. All outside table borders are removed.
above	Border on the top side of the border frame is displayed.
below	Border on the bottom side of the table frame is displayed.
border	Borders on all sides of the table frame are displayed.
box	Borders on all sides of the table frame are displayed.
hsides	Borders on the top and bottom sides of the table frame are displayed.
lhs	Border on the left side of the table frame is displayed.
rhs	Border on the right side of the table frame is displayed.
vsides	Borders on the left and right sides of the table frame are displayed.

Applies To

table

frames [ie]

Retrieves a collection of all window objects defined by the given document or defined by the document associated with the given window.

Syntax

```
[ oColl = ] object.frames
[ oObject = ] object.frames(vIndex [,iSubIndex])
```

Possible Values

<i>oColl</i>	Array of window objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To
document, window

frameBorder

Sets or retrieves whether to display a border for the frame.

Syntax

```
HTML
<ELEMENT FRAMEBORDER = sBorder... >
Scripting
object.frameBorder [ = sBorder ]
```

sBorder values

1 yes	Default. Inset border is drawn
0 no	No border is drawn

Applies To
frame, frameset, iframe

frameElement [ie]

Retrieves the frame or iframe object that is hosting the window in the parent document.

Syntax

```
HTML
[ oWinFrame = ] window.frameElement
```

Applies To
window

<frameSet>

Specifies a frameset, which is used to organize multiple frames and nested framesets.

```
Ex
<html>
<frameset cols = "25%, 25%, *">
  <frame src = "venus.htm" />
  <frame src = "sun.htm" />
  <frame src = "mercur.htm" />
</frameset>
</html>
```

Optional Attributes

cols, rows

Standard Attributes

id, class, title, style (Only allowed in XHTML 1.0 Frameset DTD)

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setActive, setAttribute, setAttributeNode, swapNode

Objects

runtimeStyle, style

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, display, filter, height, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, zoom

frameSpacing [ie]

Sets or retrieves the amount of additional space between the frames. Default value is 2.

Syntax

```
HTML
<FRAMESET FRAMESPACING = sPixels... >
Scripting
FRAMESET.frameSpacing [ = sPixels ]
```

Applies To
frame, frameset, layoutrect, table, td, th, tr

fromCharCode() [JavaScript]

The fromCharCode() takes the specified Unicode values and returns a string.

Syntax

```
String.fromCharCode(numX, numX, ..., numX)
```

Parameter

<i>numX</i>	Required. One or more Unicode values
-------------	--------------------------------------

Ex
<script type="text/javascript">
document.write(String.fromCharCode(72, 69, 76, 76, 79))
document.write("
")
document.write(String.fromCharCode(65, 66, 67))
</script>

The output of the code above will be:

HELLO
ABC

Applies to
String

fromElement [ie]

Sets or retrieves the object from which activation or the mouse pointer is exiting during the event.

Syntax

```
HTML
event.fromElement [ = oObject ]
```

Applies To
event

function ([] [])

The function statement declares a function with the specified parameters.

You can also define functions using the Function constructor and the function operator (function expression).

Syntax

```
function name([param] [,param] [..., param]) {
  statements
}
```

Parameters

name	The function name.
paramN	The name of an argument to be passed to the function. A function can have up to 255 arguments.
statements	The statements which comprise the body of the function.

Ex

```
The following code declares a function that returns the total dollar amount of sales, when given the number of units sold of products a, b, and c.
function calc_sales(units_a, units_b, units_c) {
  return units_a*79 + units_b * 129 + units_c * 699; }
```


galleryImg [ie]

Sets or retrieves whether the My Pictures image toolbar is visible for the current image.

Syntax

HTML

Scripting

img.galleryImg [= bToolbar]

bToolbar values

no false	Image toolbar is disabled for the image.
yes true	Default. Image toolbar is enabled for the image.

Applies To

img

getAdjacentText() [ie]

Returns the adjacent text string.

Syntax

text = object.getAdjacentText(sWhere)

Parameters

<i>sWhere</i>	Required. String that specifies where the text is located using one of the following values:
beforeBegin	Text is returned immediately before the element.
afterBegin	Text is returned after the start of the element but before all other content in the element.
beforeEnd	Text is returned immediately before the end of the element but after all other content in the element.
afterEnd	Text is returned immediately after the end of the element.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

getAllResponseHeaders()

Returns the complete list of response headers.

Syntax

sHeaders = XMLHttpRequest.getAllResponseHeaders()

Applies to

XMLHttpRequest

getAttribute()

(tags)

Retrieves the value of the specified attribute. Returns a String, number, or Boolean value as defined by the attribute. If the attribute is not present, this method returns null.

Syntax

vAttrValue = object.getAttribute(sAttrName [,iFlags])

Parameters

<i>sAttrName</i>	Required. Stringthat specifies the name of the attribute.
<i>iFlags</i>	Optional. Integerthat specifies one or more of the following flags:
0	Default. Performs a property search that is not case-sensitive, and returns an interpolated value if the property is found.

1	Performs a case-sensitive property search. To find a match, the uppercase and lowercase letters in sAttrName must exactly match those in the attribute name. If the iFlags parameter for getAttribute is set to 1 and this option is set to 0 (default), the specified property name might not be found.
2	Returns the value exactly as it was set in script or in the source document.

Applies to

a, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xmp

getAttribute()

(user profile)

Returns the value of the named attribute from the userProfile object.

Syntax

sValue = userProfile.getAttribute(sAttributeName)

Parameters

<i>sAttributeName</i>	Optional. String that specifies the message to display in the dialog box.
-----------------------	---

Applies to

userProfile

getAttributeNode()

Retrieves an attribute objectreferenced by the attribute. nameproperty.

Syntax

oAttribute = object.getAttributeNode(sName)

Parameters

<i>sName</i>	Required. String that specifies the name property of the requested attribute object.
--------------	--

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

getBookmark() [ie]

Retrieves a bookmark (opaque string) that can be used with moveToBookmark to return to the same range. Might not be available on non-Microsoft Win32 platforms

Syntax

sBookmark = TextRange.getBookmark()

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

TextRange

getBoundingClientRect() [ie]

Retrieves an object that specifies the bounds of a collection of TextRectangle objects. Returns a TextRectangle object. Each rectangle has four integer properties (top, left, right, and bottom) that represent a coordinate of the rectangle, in pixels

Syntax

oRect = object.getBoundingClientRect()

g

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextRange, tfoot, th, thead, tr, tt, u, ul, var, xmp

getCharset()

Retrieves a Variant that specifies the character set of the specified font.

Syntax

charset = *Dialog Helper*.getCharset(*fontName*)

Parameter

<i>fontName</i>	Required. String that specifies the name of the font.
-----------------	---

Applies to

Dialog Helper

getClientRects() [ie]

Retrieves a collection of rectangles that describes the layout of the contents of an object or range within the client. Each rectangle describes a single line. Each rectangle has four integer properties (top, left, right, and bottom) that each represent a coordinate of the rectangle, in pixels.

Syntax

collRect = *object*.getClientRects()

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextRange, tfoot, th, thead, tr, tt, u, ul, var, xmp

getData() [ie]

Retrieves the data in the specified format from the clipboard through the dataTransfer or clipboardData objects. Returns the data in the format retrieved from clipboard through the dataTransfer or clipboardData object. Depending on the information contained in setData, this variable can retrieve a path to an image, text, or an anchor URL

Syntax

sRetrieveData = *object*.getData(*sDataFormat*)

Parameters

<i>sDataFormat</i>	Required. Stringthat specifies one of the following data format values: Text: Retrieves data formatted as text; URL: Retrieves data formatted as a URL.
--------------------	--

Applies to

clipboardData, dataTransfer

getDate() [javaScript]

The getDate() method returns the day of the month, a number between 1 and 31.

Syntax

dateObject.getDate()

Ex

```
<script type="text/javascript">
var d = new Date()
document.write(d.getDate())
</script>
```

Applies to

Date

getDay() [javaScript]

The getDay() method returns a number between 0 and 6 that represents the day of the week.

Syntax

dateObject.getDay()

Ex

```
<script type="text/javascript">
var d=new Date()
document.write(d.getDay())
</script>
```

Applies to

Date

getElementById()

Returns a reference to the first object with the specified value of the ID attribute.

Syntax

oElement = *document*.getElementById(*sIDValue*)

Parameters

<i>sIDValue</i>	Required. String that specifies the value of an ID attribute.
-----------------	---

Applies to

document

getElementsByName()

Retrieves a collection of objects based on the value of the NAME attribute.

Syntax

collObjects = *document*.getElementsByName(*sNameValue*)

Parameters

<i>sNameValue</i>	Required. String that specifies the value of a NAME attribute.
-------------------	--

Applies to

document

getElementsByTagName()

Retrieves a collection of objects based on the specified element name.

Syntax

collObjects = *document*.getElementsByTagName(*sNameValue*)

Parameter

<i>sTagName</i>	Required. String that specifies the name of an element.
-----------------	---

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, ins, kbd, label, legend, li, link, listing, map, marquee, menu, ol, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

getExpression() [ie]

Retrieves the expression for the given property.

Syntax

vExpression = *object*.getExpression(*sPropertyName*)

Parameter

<i>sPropertyName</i>	Required. Stringthat specifies the name of the property from which to retrieve the expression.
----------------------	--

Applies to

a, abbr, acronym, address, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, hr, i, iframe, img, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, VAR

getFullYear() [javaScript]

The getFullYear() method returns a four-digit number that represents a year.

Syntax

dateObject.getFullYear()

Applies to

Date

getHours() [JavaScript]

The getHours() method returns the hour of a time.

Syntax

dateObject.getHours()

Applies to

Date

getMilliseconds() [JavaScript]

The getHours() method returns the milliseconds of a time.

Syntax

dateObject.getMilliseconds()

Applies to

Date

getMinutes() [JavaScript]

The getMinutes() method returns the minutes of a time.

Syntax

dateObject.getMinutes()

Applies to

Date

getMonth() [JavaScript]

The getMonth() method returns the month, as a number between 0 and 11. January is 0, February is 1 and so on.

Syntax

dateObject.getMonth()

Ex

```
<script type="text/javascript">
var d=new Date()
document.write(d.getMonth())
</script>
```

Applies to

Date

getNamedItem()

Retrieves an attribute specified with the name property using the attributes collection.

Syntax

oGetName = *attributes*.getNamedItem(*sGetName*)

Parameter

<i>sGetName</i>	Required. String that specifies the name of the attribute to retrieve.
-----------------	--

Applies to

attributes

getResponseHeader()

Returns the specified response header.

Syntax

p = *XMLHttpRequest*.getResponseHeader(*bstrHeader*)

Parameter

<i>bstrHeader</i>	Required. String that specifies the response header name.
-------------------	---

Applies to

XMLHttpRequest

getYear() [JavaScript]

The getYear() method returns the year, as a two-digit OR a four-digit number.

Syntax

dateObject.getYear()

Applies to

Date

getSeconds() [JavaScript]

The getHours() method returns the seconds of a time.

Syntax

dateObject.getSeconds()

Applies to

Date

getTime() [JavaScript]

The getTime() method returns the number of milliseconds since midnight of January 1, 1970.

Syntax

dateObject.getTime()

Applies to

Date

getTimezoneOffset() [JavaScript]

The getTimezoneOffset() method returns the difference in minutes between local time and Greenwich Mean Time (GMT).

Syntax

dateObject.getTimezoneOffset()

Applies to

Date

getUTCDay() [JavaScript]

The getUTCDay() method returns a number that represents the day of the week according to the Universal Coordinated Time (UTC). Sunday is 0, Monday is 1 and so on.

Syntax

dateObject.getUTCDay()

Applies to

Date

getUTCFullYear() [JavaScript]

The getUTCFullYear() method returns a four-digit number that represents a year according to the Universal Coordinated Time (UTC).

Syntax

dateObject.getUTCFullYear()

Applies to

Date

getUTCHours() [JavaScript]

The getUTCHours() method returns the hour of a time according to the Universal Coordinated Time (UTC).

Syntax

dateObject.getUTCHours()

Applies to

Date

getUTCMinutes() [JavaScript]

The getUTCHours() method returns the minutes of a time according to the Universal Coordinated Time (UTC).

Syntax

dateObject.getUTCMinutes()

Applies to

Date

getUTCMilliseconds() [JavaScript]

The getUTCMilliseconds() method returns the milliseconds of a time according to the Universal Coordinated Time (UTC).

Syntax

dateObject.getUTCMilliseconds()

Applies to

Date

getUTCMonth() [JavaScript]

The getUTCMonth() method returns a number that represents a month according to the Universal Coordinated Time (UTC). January is 0, February is 1 and so on.

<h1>...<h6>

Renders text in heading style. From <h1>, the biggest, to <h6>, the smaller

Ex
<h1>This is header 1</h1>
<h2>This is header 2</h2>
<h3>This is header 3</h3>
<h4>This is header 4</h4>
<h5>This is header 5</h5>
<h6>This is header 6</h6>

Optional Attributes
align (deprecated in HTML 4.01 and not supported in XHTML1.0 Strict DDT)

Standard Attributes
id, class, title, style, dir, lang, xml:lang

Behaviors
clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection
all, attributes, behaviorUrns, childNodes, children

Events
onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods
addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects
currentStyle, runtimeStyle, style

Style Properties and attributes
:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zoom

hasChildNodes()

Returns a boolean value that indicates whether the object has children.

Syntax
bChildNodes = *object*.hasChildNodes()

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to
a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

hasFeature()

Returns a boolean value that indicates whether the object implements a specific Document Object Model (DOM) standard.

Syntax
bImplemented = *implementation*.hasFeature(*sFeature* [, *vVersion*])

Parameters

<i>sFeature</i>	Required. Name of the standard. XML DOM Core standard. HTML DOM HTML standard.
<i>vVersion</i>	Optional. Version number of the standard. 1.0 Level 1.

Applies to
implementation

hasFocus() [ie]

Retrieves the boolean value indicating whether the object currently has focus.

Syntax
bActive = *document*.hasFocus()

Applies to
document

hash [ie]

Sets or retrieves the subsection of the href property that follows the number sign (#).

Syntax
HTML
N/A
Scripting
object.hash [= *sHash*]

Applies To
a, area, location

hasLayout [ie]

Retrieves a value that indicates whether the object has layout.

Syntax
HTML
N/A
Scripting
[*sHasLayout* =] *object*.currentStyle.hasLayout

sHasLayout values

false	Default. Object does not have layout.
true	Object has layout.

Applies To
frame, frameset, layoutrect, table, td, th, tr

<head>

Provides an unordered collection of information about the document. The browser does not display the “head information” to the user. The following tags can be in the head section: <base>, <basefont>, <bgsound>, <link>, <meta>, <nextID>, <script>, <style>, and <title>.

Ex
<html>
<head>
<title>Title of the document</title>
</head>

<body>
...

...
</body>
</html>

Optional Attributes
profile
Standard Attributes
dir, lang, xml:lang
Behaviors
clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, appendChild, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeNode, replaceAdjacentText, replaceChild, replaceNode, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight, minWidth, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition, zoom

headers

Sets or retrieves a list of header cells (space-separeted cell ids) that provide information for the object.

Syntax

HTML

<ELEMENT HEADERS = *sHeaders...* >

Scripting

object.headers(*v*) [= *sHeaders*]

Applies To

td, th

height

(screen)

Retrieves the vertical resolution of the screen, in px.

Syntax

[*iHeight* =] *screen*.height

Applies To

screen

height

(css)

Sets or retrieves the height of the object.

Syntax

CSS

{ height : *sHeight* }

Scripting

object.height [= *sHeight*]

sHeight values

auto	Default
height	Integer that specifies the height of the object, in pixels
%	A percentage of the height of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, button, caption, center, cite, code, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tr, tt, u, ul, var, xmp

height

(embed)

Sets or retrieves the height of the object.

Syntax

HTML

<ELEMENT HEIGHT = *sHeight...* >

Scripting

object.height [= *sHeight*]

sHeight values

height	Integer that specifies the height of the object, in pixels
%	A percentage of the height of the parent object

Applies To

embed, frame, iframe, img, marquee, nobr, object, table, td, th, tr

hide()

Closes the pop-up window.

Syntax

popup.hide()

Applies to

popup

hidden [ie]

Sets or retrieves the value indicating whether the embedded object is invisible.

Syntax

HTML

N/A

Scripting

EMBED.hidden(*v*) [= *sHidden*]

sHidden values

false	Default. Object is not hidden
true	Object is hidden

Applies To

EMBED

hideFocus [ie]

Sets or retrieves the value indicating whether the object visibly indicates that it has focus.

Syntax

HTML

<ELEMENT BORDERCOLOR = *sColor...* >

Scripting

object.borderColor [= *sColor*]

bHidden values

false	Default. Focus is visibly indicated
true	Focus is not visibly indicated

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

history

Contains information about the URLs visited by the client.

Property	Description
length	Retrieves the number of elements in the History list.

Methods

back, forward, go

host [ie]

Sets or retrieves the hostname and port number of the location or URL.

Syntax

HTML

N/A

Scripting

object.host [= *sHost*]

Applies To

a, area, location

hostname [ie]

Sets or retrieves the host name part of the location or URL.

Syntax

HTML

N/A

Scripting

object.hostname [= *sHostname*]

Applies To
a, area, location

<hr>

Draws a horizontal rule.
In HTML the <hr> tag has no end tag, but in XHTML the <hr> tag must be properly closed.

Ex
This is some text <hr /> This is some text

Optional Attributes
align, noshade, size, width (deprecated, use CSS)

Standard Attributes
id, class, title, style, dir, lang, xml:lang

Behaviors
clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection
all, attributes, behaviorUrns, childNodes, children

Events
onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods
addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects
runtimeStyle, style

Style Properties and attributes
backgroundPositionX, backgroundPositionY, behavior, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, cursor, display, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, lineBreak, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textIndent, textJustify, textKashidaSpace, textUnderlinePosition, top, visibility, whiteSpace, width, wordBreak, wordWrap, writingMode, zoom

href

(base)

Sets or retrieves the baseline URL on which relative links will be based.

Syntax
<base HREF = *sURL*... >

Scripting
base.href [= *sURL*]

Applies To
base

href

(location)

Sets or retrieves the entire URL as a string.

Syntax
location.href [= *sURL*]

Applies To
location

href

(css)

Sets or retrieves the URL of the linked style sheet.

Syntax
styleSheet.href [= *sURL*]

sURL values

<i>sURL</i>	String that specifies or receives the URL.
-------------	--

Applies To
styleSheet

href

Sets or retrieves the destination URL or anchor point.

Syntax
HTML
<*ELEMENT* HREF = *sURL*... >

Scripting
object.href [= *sURL*]

Applies To
a, area, link

hreflang

Sets or retrieves the language code of the object.

Syntax
HTML
<*ELEMENT* HREFLANG = *sLang*... >

Scripting
object.hreflang(*v*) [= *sLang*]

Applies To
a, link

hspace

Sets or retrieves the horizontal margin for the object, in px.

Syntax
HTML
<*ELEMENT* HSPACE = *iMargin*... >

Scripting
object.hspace [= *iMargin*]

Applies To
applet, iframe, img, input, input type=image, marquee, object

<html>

Identifies the document as containing HTML elements.
The xmlns attribute is required in XHTML but not in HTML.
However, the validator on w3.org does not complain when this attribute is missing from the <html> tag in an XHTML document. This is because “xmlns=http://www.w3.org/1999/xhtml” is a fixed value and will be added to the <html> tag even if you do not include it.

Ex
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
<body>
...
</body>
</html>

Required Attribute
xmlns="http://www.w3.org/1999/xhtml" (Defines the XML namespace attribute)

Standard Attributes
dir, lang, xml:lang

Behaviors
clientCaps, download, homePage

Collection
all, attributes, behaviorUrns, childNodes, children

Methods
addBehavior, appendChild, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeNode, replaceAdjacentText, replaceChild, replaceNode, setAttribute, setAttributeNode, swapNode

Objects
currentStyle, runtimeStyle, style

Style Properties and attributes
:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundRepeat, behavior, color, cursor, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight,

minWidth, overflow, overflowX, overflowY, textAutospace, textDecoration, text-decoration-blink, text-decoration-line-through, text-decoration-none, text-decoration-overline, text-decoration-underline, textTransform, textUnderlinePosition, visibility, wordSpacing, wordWrap

htmlFor

(label)

Sets or retrieves the object to which the given label object is assigned.

Syntax

HTML

<label FOR = s/D... >

Scripting

label.htmlFor [= s/D]

Applies To

label

htmlFor

(script)

Sets or retrieves the object that is bound to the event script.

Syntax

HTML

<script FOR = oObject... >

Scripting

script.htmlFor [= oObject]

Applies To

script

htmlText [ie]

Retrieves the HTML source as a valid HTML fragment.

Syntax

HTML

[sTxt =] TextRange.htmlText

Applies To

TextRange

httpEquiv

Sets or retrieves information used to bind the meta tag's content to an HTTP response header.

Syntax

HTML

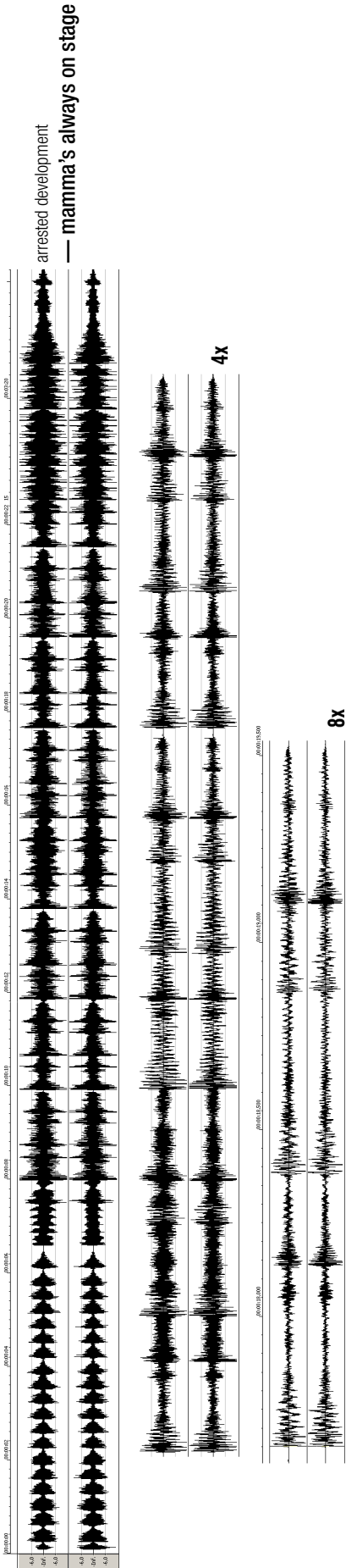
<meta HTTP-EQUIV = sInformation... >

Scripting

meta.httpEquiv [= sInformation]

Applies To

meta



<i>

Specifies that the text should be rendered in italic, where available.

Ex

<i>Italic text</i>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

id

Retrieves the string identifying the object.

Syntax

HTML

<ELEMENT ID = *sID*... >

Scripting

object.id [= *sID*]

Applies To

STYLE, a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

if ... else

The if ... else statement executes a statement if a specified condition is true. If the condition is false, another statement can be executed.

Syntax

if (*condition*)

statement1

[else

statement2]

or

if (*condition1*)

statement1

else if (*condition2*)

statement2

else if (*condition3*)

statement3

...

else

statementN

Parameters

| <i>condition</i> | An expression that evaluates to true or false. |
|-------------------|---|
| <i>statement1</i> | Statement that is executed if condition evaluates to true. Can be any statement, including further nested if statements. To execute multiple statements, use a block statement ({ ... }) to group those statements. |
| <i>statement2</i> | Statement that is executed if condition evaluates to false and the else clause exists. Can be any statement, including block statements and further nested if statements |

Ex

```
if (cipher_char == from_char) {
    result = result + to_char;
    x++;
} else
    result = result + clear_char;
```

<iframe>

Creates inline floating frames that can contain another document.

Ex

```
<iframe
src ="/default.asp"
width="100%">
</iframe>
```

Optional Attributes

align, frameborder, longdesc, marginheight, marginwidth, name, scrolling, src, width

Standard Attributes

id, class, title, style

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setExpression, swapNode

Objects

runtimeStyle, style

Style Properties and attributes

accelerator, backgroundAttachment, backgroundColor, backgroundPosition, backgroundPositionY, behavior, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, cursor, display, filter, height, layoutGrid, layoutGridMode, left, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflowX, overflowY, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, top, visibility, zIndex, zoom

imeMode [ie]

Sets or retrieves the state of an Input Method Editor (IME).

Syntax

CSS
{ ime-mode : *sMode* }
Scripting
object.style.imeMode [= *sMode*]
sMode values

| | |
|----------|--|
| auto | Default. IME is not affected. This is the same as not specifying the ime-mode attribute. |
| active | All characters are entered through the IME. Users can still deactivate the IME. |
| inactive | All characters are entered without IME. Users can still activate the IME. |
| disabled | IME is completely disabled. Users cannot activate the IME if the control has focus |

Applies To
currentStyle, input type=text, runtimeStyle, style, textarea

images

Retrieves a collection, in source order, of img objects in the document.

Syntax

[*oColl* =] document.images
[*oObject* =] document.images(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|------------------|---|
| <i>oColl</i> | Array of img objects. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> . |

Applies To
document

Embeds an image or a video clip in the document.
In HTML the tag has no end tag.
In XHTML the tag must be properly closed.

Ex

Required Attributes

alt, src

Optional Attributes

Attribute, border, height, hspace, ismap, longdesc, usemap, vspace, width (The "align", "border", "hspace", and "vspace" attributes of the image element were deprecated in HTML 4.01, and not supported in XHTML 1.0, use CSS)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, releaseCapture, removeAttribute,

removeAttributeNode, removeBehavior, removeExpression, replaceAdjacentText, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, cursor, direction, display, filter, font, fontFamily, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, msInterpolationMode, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, wordSpacing, wordWrap, zoom

implementation

Contains information about the modules supported by the object.

Method

hasFeature

Applies To

document

implementation

Retrieves the implementation object of the current document.

Syntax

[*oImplementation* =] *document.implementation*

Applies To

document

import

The import statement allows a script to import properties, functions, and objects from a signed script that has exported the information. This feature is not in ECMA 262, Edition 3.

Syntax

import *objectName.name1*, *objectName.name2*, ..., *objectName.nameN*;

import *objectName.**;

Parameters

| | |
|------------|--|
| objectName | Name of the object that will receive the imported names. |
| nameN | Property, function, or object to be imported. |

Ex

import obj.f, obj.p;

IMPORT

(processing instruction)

Imports a tag definition from an element behavior.

The value of a XMLNS attribute, defined in the html tag, must correspond to the value of the namespace attribute defined in the IMPORT processing instruction. Otherwise, the imported tag definition cannot be used.

Syntax

<?IMPORT
 namespace = "sNameSpace"
 implementation = "sImplementation"
>

Ex

The following sample illustrates the use of the IMPORT processing instruction. The XMLNS attribute defined in the html tag is defined to match the namespace attribute in the IMPORT processing instruction:

```
<HTML xmlns:games>  
<?IMPORT namespace="games" implementation="checkers.htc" >  
<body>
```

```
<games:checkers />
</body>
</HTML>
```

Required Attributes

| Attribute | Description |
|----------------|---|
| namespace | String that specifies the previously-defined namespace into which the tag definition is imported |
| implementation | One of the following values: <div> <div>#default</div> <div>An element behavior built into Microsoft Internet Explorer 5.5 or later.</div> <div>#objectID</div> <div>String that specifies the id attribute of an object tag.</div> <div>smplementation</div> <div>String that specifies the HTML Component (HTC) file that contains the tag definition.</div> </div> |

ImportExportFavorites() [ie]

Deprecated. Handles the importing and exporting of Microsoft Internet Explorer favorites.

Syntax

external.ImportExportFavorites(blmportExport, slmportExportPath)

Parameter

| | |
|--------------------------|--|
| <i>blmportExport</i> | Required. Boolean that specifies one of the following possible values.
true Import is requested.
false Export is requested. |
| <i>slmportExportPath</i> | Required. String that specifies the location (URL) to import or export, depending on blmportExport . If a value is an empty string, a file dialog box is opened. |

Applies to

external

imports [ie]

Retrieves a collection of all the imported style sheets defined for the respective styleSheet object.

Syntax

[*oColl* =] styleSheet.imports

[*oObject* =] styleSheet.imports(*iIndex*)

Possible Values

| | |
|----------------|---|
| <i>oColl</i> | Array of imported style sheets. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer that specifies the zero-based index of the item to be returned. |

Applies To

styleSheet

indeterminate [ie]

Sets or retrieves whether the user has changed the status of a check box.

Syntax

N/A

Scripting

input type=checkbox.indeterminate [= *bDim*]

bDim values

| | |
|-------|----------------------------------|
| false | Default. Check box is not dimmed |
| true | Check box is checked and dimmed |

Applies To

input type=checkbox

In

The in operator returns true if the specified property is in the specified object.

Syntax

propNameOrNumber in *objectName*

Parameters

| | |
|-------------------------|---|
| <i>propNameOrNumber</i> | A string or numeric expression representing a property name or array index. |
| <i>objectName</i> | Name of an object. |

Ex

The following examples show some uses of the in operator.

// Arrays

```
eles=new Array("Marx", "Lenin", "Trotsky", "Rosa")
```

```
0 in eles // returns true
```

```
3 in eles // returns true
```

```
6 in eles // returns false
```

```
"Marx" in eles // returns false (you must specify the index number,
// not the value at that index)
```

```
"length" in eles // returns true (length is an Array property)
```

```
// Predefined objects
```

```
"PI" in Math // returns true
```

```
// Custom objects
```

```
mycar = {make:"Honda",model:"Accord",year:1998}
```

```
"make" in mycar // returns true
```

```
"model" in mycar // returns true
```

You must specify an object on the right side of the in operator. For example, you can specify a string created with the String constructor, but you cannot specify a string literal.

```
color1=new String("green")
```

```
"length" in color1 // returns true
```

```
color2="coral"
```

```
"length" in color2 // generates an error (color is not a String object)
```

indexOf() [JavaScript]

The indexOf() method returns the position of the first occurrence of a specified string value in a string.

Syntax

stringObject.indexOf(searchvalue, fromindex)

Parameter

| | |
|--------------------|--|
| <i>searchvalue</i> | Required. Specifies a string value to search for |
| <i>fromindex</i> | Optional. Specifies where to start the search |

Ex

In this example we will do different searches within a "Hello world!" string:

```
<script type="text/javascript">
```

```
var str="Nem a pau"
```

```
document.write(str.indexOf("Nem") + "<br />")
```

```
document.write(str.indexOf("Pau") + "<br />")
```

```
document.write(str.indexOf("pau"))
```

```
</script>
```

The output of the code above will be:

```
0
```

```
-1
```

```
7
```

Applies to

String

Infinity [JavaScript]

The Infinity property is a numeric value that represents positive or negative infinity.

Infinity is displayed when a number is higher than 1.79769313 48623157E+10308, and -Infinity is displayed when a number is lower than -1.7976931348623157E+10308. Anything divided by Infinity is 0 and anything multiplied by Infinity is Infinity.

Syntax

Infinity

Ex

In this example we will show what happens if a number exceeds the infinity limit:

```
<script type="text/javascript">
```

```
var t1=1.7976931348623157E+10308
```

```
document.write(t1)
```

```
document.write("<br />")
```

```
var t2=-1.7976931348623157E+10308
```

```
document.write(t2)
```

```
</script>
```

The output of the code above will be:

Infinity

-Infinity

Applies To

JavaScript Functions

index

Sets or retrieves the ordinal position of an option in a list box.

Syntax

HTML

N/A

Scripting

option.index [= *iIndex*]

Applies To

option

innerHTML [ie]

Sets or retrieves the HTML between the start and end tags of the object.

Syntax

HTML

N/A

Scripting

object.innerHTML [= *sHTML*]

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, hn, HTML Comment, html, i, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, ol, option, p, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, STYLE, sub, sup, table, tbody, td, tfoot, th, thead, title, tr, tt, u, ul, VAR

innerText [ie]

Sets or retrieves the text between the start and end tags of the object.

Syntax

HTML

N/A

Scripting

object.innerText [= *sTxt*]

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, em, fieldset, font, form, hn, html, i, iframe, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

<input>

Creates a variety of form input controls.

In HTML the <input> tag has no end tag.

In XHTML the <input> tag must be properly closed.

Use the label element to define a label to a form control.

Ex

```
<form action="form_action.asp"
method="get">
```

First name:

```
<input type="text" name="fname" value="Mickey" />
<br />
```

Last name:

```
<input type="text" name="lname" value="Mouse" />
<br />
```

```
<input type="submit" value="Submit" />
</form>
```

<p>

If you click the "Submit" button, you will send your input to a new page called form_action.asp.

</p>

Properties

accept	Sets or retrieves a comma-separated list of content types.
align	Sets or retrieves how the object is aligned with adjacent text.

alt	Sets or retrieves a text alternative to the graphic.
complete	Retrieves whether the object is fully loaded.
dynsrc	Sets or retrieves the address of a video clip or VRML world to display in the window.
hspace	Sets or retrieves the horizontal margin for the object.
loop	Sets or retrieves the number of times a sound or video clip will loop when activated.
lowsrc	Sets or retrieves a lower resolution image to display.
start	Sets or retrieves when a video clip file should begin playing.
useMap	Sets or retrieves the URL, often with a bookmark extension (#name), to use as a client-side image map.
value	Sets or retrieves the default or selected value of the control.
vspace	Sets or retrieves the vertical margin for the object.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

mask

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Style Properties and attributes

:first-child, :hover, layoutFlow, textUnderlinePosition, wordWrap, writingMode, zoom

<input type=button>

Creates a button control.

Ex

```
<input TYPE=button ID=btnEmergency VALUE="In case of emergency,
push this button!"
onClick="alert('Aaaaaarrrrrgggghh!!!!')">
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, writingMode, zIndex, zoom,

<input type=checkbox>

Creates a check box control.

Ex

This example uses the input type=checkbox element to create two check boxes with explanatory text. The onclick events call two script functions. The first check box is checked:

```
<input TYPE=checkbox CHECKED ID=chk1 onclick="choosebox1()">Uncheck  
  this check box if you like  
<P><input TYPE=checkbox ID=chk2 onclick="choosebox2()">Or check  
  this if you understand  
<P ID=SampText>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, zIndex, zoom

<input type=file>

Creates a file upload object with a text box and Browse button.

For a file upload to take place:

The input type=file element must be enclosed within a FORM element; a value must be specified for the NAME attribute of the input type=file element; the METHOD attribute of the FORM element must be set to post; the ENCTYPE attribute of the FORM element must be set to multipart/form-data.

Ex

The following examples use the input type=file element to upload a file to a server. The first example requires Microsoft Posting Acceptor, which can be used with IIS or Personal Web Server.

This example uses HTML code to submit a file selected by the user to Cps host. dll, which is installed with Posting Acceptor.

```
<FORM NAME="oForm"  
  ACTION="repost.asp"  
  ENCTYPE="multipart/form-data"  
  METHOD="post">  
<input TYPE="file" NAME="oFile1"/>  
<input TYPE="submit" VALUE="Upload File">
```

</FORM>

This example shows the Active Server Page (ASP) content of Repost.asp. Notice that the properties of the uploaded file are accessible from the submitted form.

```
<%@ LANGUAGE = JScript %>  
<%  
  Response.buffer=true;  
%>  
<HTML>  
<title>Repost Example</title>  
<body>  
<H1>Upload Status</H1>  
<P>  
  Destination: <B><% Response.Write(Server.HtmlEncode(Request.  
Form("TargetURL"))) %></B>  
</P>  
<%  
  Response.write("<P>Name: " + Server.HtmlEncode(Request.  
Form("FileName")) + "</P>");  
  Response.write("<P>Size: " + Server.HtmlEncode(Request.Form("FileSize"))  
+ "</P>");  
  Response.write("<P>Path: " + Server.HtmlEncode(Request.Form("FilePath"))  
+ "</P>");  
%>  
</body>  
</HTML>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filter

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoratation, textDecoratationBlink, textDecoratationLineThrough, textDecoratationNone, textDecoratationOverline, textDecoratationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, writingMode, zIndex, zoom

<input type=hidden>

Transmits state information about client/server interaction.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filter

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick,

onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, writingMode, zIndex, zoom

<input type=

image>
Creates an image control that, when clicked, causes the form to be immediately submitted.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, createTextRange, detachEvent, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getExpression, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, replaceAdjacentText, replaceChild, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, zIndex, zoom

<input type=

password>
Creates a single-line text entry control similar to the input type=text control, except that text is not displayed as the user enters it.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, writingMode, zIndex, zoom

<input type=

radio>
Creates a radio button control.

Ex

<input type=radio name="radio" CHECKED>1-10 years old
<input type=radio name="radio">11 years old
<input type=radio name="radio">12-120 years old

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, zIndex, zoom

<input type=reset>

Creates a button that, when clicked, resets the form's controls to their initial values.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, zIndex, zoom

<input type=submit>

Creates a button that, when clicked, submits the form.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, filters

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout,

onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, zIndex, zoom

<input type=text>

Creates a single-line text entry control. The SIZE attribute sets the number of visible characters in the input type=text element. The MAXLENGTH attribute sets the maximum number of characters that can be entered.

Using this object incorrectly can compromise the security of your application. When submitting text through input type=text over an intranet or the Internet, validating the text string is recommended. For instance, you might validate the string for a restricted set of known, good values (like letters and numbers) and ignore the rest.

Ex

<input TYPE=text VALUE="" NAME="textbox" SIZE=15>

This example uses script to detect the content of the text box and display it in a dialog box:

```
<script>
function detectEntry()
{
    alert(textbox.value + "seu bobo!")
}
</script>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, imeMode, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, wordSpacing, wordWrap, writingMode, zIndex, zoom

inRange() [ie]

Returns a boolean value indicating whether one range is contained within another.

Syntax

bFound = *TextRange*.inRange(*oRange*)

Parameter

<i>oRange</i>	Required. TextRange object that might be contained.
---------------	---

Applies to

TextRange

<ins>

Specifies text that has been inserted into the document. Usually rendered underlined.

Ex

a dozen is 20 <ins>12</ins> pieces!

Optional Attributes

cite, datetime

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

insertAdjacentElement() [ie]

Inserts an element at the specified location. Returns an element object

Syntax

oElement = *object*.insertAdjacentElement(*sWhere*, *oElement*)

Parameters

<i>sWhere</i>	Required. String that specifies where to insert the HTML element, using one of the following values:
beforeBegin	Inserts <i>sText</i> immediately before the object.
afterBegin	Inserts <i>sText</i> after the start of the object but before all other content in the object.
beforeEnd	Inserts <i>sText</i> immediately before the end of the object but after all other content in the object.
afterEnd	Inserts <i>sText</i> immediately after the end of the object.
<i>sText</i>	Required. String that specifies the HTML text to insert. The string can be a combination of text and HTML tags. This must be well-formed, valid HTML or this method will fail.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

insertAdjacentHTML() [ie]

Inserts the given HTML text into the element at the location.

Syntax

object.insertAdjacentHTML(*sWhere*, *sText*)

Parameters

<i>sWhere</i>	Required. String that specifies where to insert the HTML element, using one of the following values:
beforeBegin	Inserts <i>sText</i> immediately before the object.
afterBegin	Inserts <i>sText</i> after the start of the object but before all other content in the object.
beforeEnd	Inserts <i>sText</i> immediately before the end of the object but after all other content in the object.
afterEnd	Inserts <i>sText</i> immediately after the end of the object.
<i>sText</i>	Required. String that specifies the HTML text to insert. The string can be a combination of text and HTML tags. This must be well-formed, valid HTML or this method will fail.

Applies to

a, address, area, b, basefont, big, blockquote, body, button, caption, center, cite, code, comment, custom, dd, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, ol, option, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, td, textarea, th, tt, u, ul, var, xmp

insertAdjacentText() [ie]

Inserts the given text into the element at the specified location.

Syntax

object.insertAdjacentText(*sWhere*, *sText*)

Parameters

<i>sWhere</i>	Required. String that specifies where to insert the HTML element, using one of the following values:
beforeBegin	Inserts <i>sText</i> immediately before the object.
afterBegin	Inserts <i>sText</i> after the start of the object but before all other content in the object.

beforeEnd	Inserts <i>sText</i> immediately before the end of the object but after all other content in the object.
afterEnd	Inserts <i>sText</i> immediately after the end of the object.
<i>sText</i>	Required. String that specifies the HTML text to insert. The string can be a combination of text and HTML tags. This must be well-formed, valid HTML or this method will fail.

Applies to

a, address, area, b, basefont, big, blockquote, body, button, caption, center, cite, code, comment, custom, dd, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, ol, option, p, pre, s, samp, select, small, span, strike, strong, sub, sup, td, textarea, th, tt, u, ul, VAR

insertBefore()

Inserts an element into the document hierarchy as a child node of a parent object.

Syntax

oElement = *object.insertBefore(oNewNode [,oChildNode])*

Parameters

<i>oNewNode</i>	Required. Object that specifies the new element to be inserted into the document hierarchy. Elements can be created with the createElement method.
<i>oChildNode</i>	Optional. Object that specifies the placement of the new element. If this parameter is specified, the new element will be inserted immediately before this existing child element.

Applies to

a, abbr, acronym, address, attribute, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

insertCell()

Creates a new cell in the table row (tr), and adds the cell to the cells collection.

Syntax

oTD = *tr.insertCell([iIndex])*

Parameters

<i>iIndex</i>	Optional. Integer that specifies where to insert the cell in the tr. The default value is -1, which appends the new cell to the end of the cells collection.
---------------	--

Applies to

tr

insertData()

Inserts a new character string in the object at a specified offset.

Syntax

object.insertData(nOffset, sString)

Parameters

<i>nOffset</i>	Required. Integerthat specifies the offset from which to start.
<i>sString</i>	Required. Stringthat specifies the new character string.

Applies to

comment, TextNode

insertRow()

Creates a new row (tr) in the table, and adds the row to the rows collection. Returns the tr element object if successful, or null otherwise

Syntax

oTR = *object.insertRow([iIndex])*

Parameter

<i>iIndex</i>	Optional. Integerthat specifies where to insert the row in the rows collection. The default value is -1, which appends the new row to the end of the rows collection.
---------------	---

Applies to

table, tbody, tfoot, thead

instanceof

The instanceof operator returns true if the specified object is of the specified object type.

Syntax

objectName instanceof objectType

Parameters

objectName	Name of the object to compare to objectType.
objectType	Object type.

Ex

You must specify an object on the right side of the instanceof operator. For example, you can specify a string created with the String constructor, but you cannot specify a string literal.

```
color1=new String("green")
color1 instanceof String // returns true
color2="coral"
color2 instanceof String // returns false (color is not a String object)
```

isContentEditable [ie]

Retrieves the value indicating whether the user can edit the contents of the object.

Syntax

HTML

N/A

Scripting

[*bCanEdit* =] *object.isContentEditable*

bCanEdit values

false	The content cannot be edited
true	The content can be edited

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

isDisabled [ie]

Retrieves the value indicating whether the user can interact with the object.

Syntax

HTML

N/A

Scripting

[*bDisabled* =] *object.isDisabled*

bDisabled values

false	User can interact with the object
true	User cannot interact with the object

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

isEqual() [ie]

Returns a boolean value indicating whether the specified range is equal to the current range.

Syntax

bEqual = *TextRange.isEqual(oCompareRange)*

Parameters

<i>oCompareRange</i>	Required. TextRange object to compare with the current TextRange object.
----------------------	--

Applies to

TextRange

isFinite() [JavaScript]

The isFinite() function is used to check if a value is a finite number.

Syntax

isFinite(*value*)

Ex

```
<script type="text/javascript">
document.write(isFinite(123)+ "<br />")
document.write(isFinite(-1.23)+ "<br />")
document.write(isFinite(5-2)+ "<br />")
document.write(isFinite(0)+ "<br />")
document.write(isFinite("Hello")+ "<br />")
document.write(isFinite("2005/12/12")+ "<br />")
</script>
```

The output of the code above will be:

true
true
true
true
false
false

Applies to

all of the built-in JavaScript objects

<isIndex>

Causes the browser to display a dialog window that prompts the user for a single line of input. Deprecated, use input.

isMap

Sets or retrieves whether the image is a server-side image map.

Syntax

HTML

Scripting

img.isMap [= *bMap*]

bMap values

false	Default. Image is not a server-side map
true	Image is a server-side map

Applies To

img

isMultiLine [ie]

Retrieves the value indicating whether the content of the object contains one or more lines.

Syntax

HTML

N/A

Scripting

[*bMulti* =] *object.isMultiLine*

bMulti values

false	The content contains exactly one line
true	The content contains more than one line

Applies To

defaults, a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image,

input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

isNaN() [JavaScript]

The isNaN() function is used to check if a value is not a number.

Syntax

isNaN(*number*)

Ex

```
<script type="text/javascript">
document.write(isNaN(123)+ "<br />")
document.write(isNaN(-1.23)+ "<br />")
document.write(isNaN(5-2)+ "<br />")
document.write(isNaN(0)+ "<br />")
document.write(isNaN("Hello")+ "<br />")
document.write(isNaN("2005/12/12")+ "<br />")
</script>
```

The output of the code above will be:

false
false
false
false
true
true

Applies to

all of the built-in JavaScript objects

isOpen [ie]

Retrieves a value indicating whether the popup window is open.

Syntax

[*bOpen* =] *popup.isOpen*

bOpen values

false	Pop-up window is closed
true	Pop-up window is open

Applies To

popup

isTextEdit [ie]

Retrieves whether a TextRange object can be created using the object.

Syntax

HTML

N/A

Scripting

[*bEdit* =] *object.isTextEdit*

bEdit values

false	TextRange object cannot be created
true	TextRange object can be created

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

IsSubscribed() [ie]

Retrieves a value indicating whether the client subscribes to the given channel.

Syntax

bSubscribed = *external.IsSubscribed(sURLToCDF)*

Parameters

<i>sURLToCDF</i>	Required. Stringthat specifies the URL of a Channel Definition Format (CDF) file to be checked for a subscription.
------------------	--

Applies to

external

italics() [JavaScript]

The italics() method is used to display a string in italic.

Syntax

stringObject.italics()

Ex

```
<script type="text/javascript">
var str="Nem a pau!"
document.write(str.italics())
</script>
```

Applies to

String

item()

(tags)

Retrieves an object from the all collection or various other collections.

Syntax

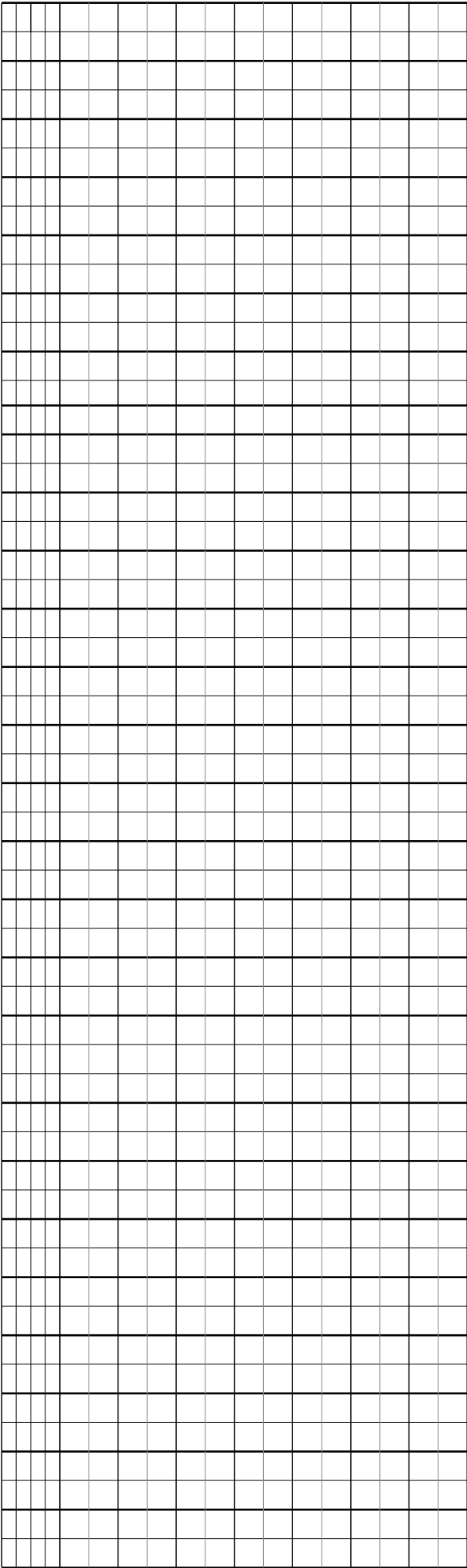
```
oltem = object.item(vIndex [,iSubindex])
oltem = attributes.item(vIndex)
oltem = pages.item(index)
```

Parameters

vIndex	Required. Integer or String that specifies the object or collection to retrieve. If this parameter is an integer, it is the zero-based index of the object. If this parameter is a string, all objects with matching name or id properties are retrieved, and a collection is returned if more than one match is made.
iSubindex	Optional. Integer that specifies the zero-based index of the object to retrieve when a collection is returned.

Applies to

all, anchors, applets, areas, boundElements, cells, elements, embeds, forms, images, links, mimeTypees, options, plugins, rows, scripts, tBodies, FORM attributes, behaviorUrns, blockFormats, fonts, bookmarks, childNodes, children, controlRange, filters, frames, imports, styleSheets, TextRange, TextRectangle, namespaces, pages, rules



javaEnabled() [ie]

Returns a boolean value indicating whether Java is enabled.

Syntax

bEnabled = *object*.javaEnabled()

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

clientInformation, navigator

join() [JavaScript]

The join() method is used to put all the elements of an array into a string.

The elements will be separated by a specified separator.

Syntax

arrayObject.join(separator)

Parameters

<i>separator</i>	Optional. Specifies the separator to be used
------------------	--

Ex

In this example we will create an array, and then put all the elements in a string:

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Marx"
arr[1] = "Lenin"
arr[2] = "Trotsky"
document.write(arr.join() + "<br />")
document.write(arr.join("."))
</script>
```

The output of the code above will be:

Marx, Lenin, Trotsky
Marx.Lenin.Trotsky

Applies to

Array

<kbd>

Renders text in a fixed-width font (keyboard text).

Ex

```
<kbd>Keyboard text</kbd>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

keyCode [ie]

Sets or retrieves the Unicode key code associated with the key that caused the event.

Syntax

event.keyCode [= *sKeyCode*]

Applies To

event

<label>

Specifies a label for another element on the page.

Ex

```
<p>Try clicking on the text labels:</p>
<form name="input" action="">
<input type="radio" name="sex" id="male" />
<label for="male">Male</label>
<br />
<input type="radio" name="sex" id="female" />
<label for="female">Female</label>
</form>
```

Optional Attributes

for (*id_of_another_field*)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

label

(statement)

Provides a statement with an identifier that you can refer to using a break or continue statement.

For example, you can use a label to identify a loop, and then use the break or continue statements to indicate whether a program should interrupt the loop or continue its execution.

Syntax

label :

statement

Parameters

label	Any JavaScript identifier that is not a reserved word.
statement	Statements. break can be used with any labeled statement, and continue can be used with looping labeled statements.

label

(optgroup)

Sets or retrieves the label that is displayed in the select element for the option group.

Syntax

HTML

<OPTGROUP label = sLabel... >

Scripting

OPTGROUP.label(v) [= sLabel]

Applies To

OPTGROUP

label

(option)

Sets or retrieves a value that you can use to implement your own label functionality for the object.

Syntax

HTML

<option label = sLabel... >

Scripting

option.label(v) [= sLabel]

Applies To

option

lang

Sets or retrieves the language to use.

Syntax

HTML

<ELEMENT LANG = sLanguage... >

Scripting

object.lang [= sLanguage

Language Code

af	Afrikaans	sq	Albanian
ar-sa	Arabic (Saudi Arabia)	ar-iq	Arabic (Iraq)
ar-eg	Arabic (Egypt)	ar-ly	Arabic (Libya)
ar-dz	Arabic (Algeria)	ar-ma	Arabic (Morocco)
ar-tn	Arabic (Tunisia)	ar-om	Arabic (Oman)
ar-ye	Arabic (Yemen)	ar-sy	Arabic (Syria)
ar-jo	Arabic (Jordan)	ar-lb	Arabic (Lebanon)
ar-kw	Arabic (Kuwait)	ar-ae	Arabic (U.A.E.)
ar-bh	Arabic (Bahrain)	ar-qa	Arabic (Qatar)
eu	Basque	bg	Bulgarian
be	Belarusian	ca	Catalan
zh-tw	Chinese (Taiwan)	zh-cn	Chinese (PRC)
zh-hk	Chinese (Hong Kong SAR)	zh-sg	Chinese (Singapore)
hr	Croatian	cs	Czech
da	Danish	nl	Dutch (Standard)
nl-be	Dutch (Belgium)	en	English
en-us	English (United States)	en-gb	English (United Kingdom)
en-au	English (Australia)	en-ca	English (Canada)
en-nz	English (New Zealand)	en-ie	English (Ireland)
en-za	English (South Africa)	en-jm	English (Jamaica)
en	English (Caribbean)	en-bz	English (Belize)
en-tt	English (Trinidad)	et	Estonian
fo	Faeroese	fa	Farsi
fi	Finnish	fr	French (Standard)
fr-be	French (Belgium)	fr-ca	French (Canada)
fr-ch	French (Switzerland)	fr-lu	French (Luxembourg)
gd	Gaelic (Scotland)	gd-ie	Gaelic (Ireland)
de	German (Standard)	de-ch	German (Switzerland)
de-at	German (Austria)	de-lu	German (Luxembourg)
de-li	German (Liechtenstein)	el	Greek
he	Hebrew	hi	Hindi

hu	Hungarian	is	Icelandic
id	Indonesian	it	Italian (Standard)
it-ch	Italian (Switzerland)	ja	Japanese
ko	Korean	ko	Korean (Johab)
lv	Latvian	lt	Lithuanian
Macedonian (FYROM)		ms	Malaysian
mt	Maltese	no	Norwegian (Bokmal)
no	Norwegian (Nynorsk)	pl	Polish
pt-br	Portuguese (Brazil)	pt	Portuguese (Portugal)
rm	Rhaeto-Romanic	ro	Romanian
ro-mo	Romanian (Moldavia)	ru	Russian
ru-mo	Russian (Moldavia)	sz	Sami (Lappish)
sr	Serbian (Cyrillic)	sr	Serbian (Latin)
sk	Slovak	sl	Slovenian
sb	Sorbian	es	Spanish (Spain ? Traditional)
es-mx	Spanish (Mexico)	es	Spanish (Spain ? Modern)
es-gt	Spanish (Guatemala)	es-cr	Spanish (Costa Rica)
es-pa	Spanish (Panama)	es-do	Spanish (Dominican Rep)
es-ve	Spanish (Venezuela)	es-co	Spanish (Colombia)
es-pe	Spanish (Peru)	es-ar	Spanish (Argentina)
es-ec	Spanish (Ecuador)	es-cl	Spanish (Chile)
es-uy	Spanish (Uruguay)	es-py	Spanish (Paraguay)
es-bo	Spanish (Bolivia)	es-sv	Spanish (El Salvador)
es-hn	Spanish (Honduras)	es-ni	Spanish (Nicaragua)
sx	Sutu	es-pr	Spanish (Puerto Rico)
sv	Swedish	sv-fi	Swedish (Finland)
th	Thai	ts	Tsonga
tn	Tswana	tr	Turkish
uk	Ukrainian	ur	Urdu
ve	Venda	vi	Vietnamese
xh	Xhosa	ji	Yiddish
zu	Zulu		

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

language

Sets or retrieves the language in which the current script is written.

Syntax

HTML

<ELEMENT LANGUAGE = sLanguage... >

Scripting

object.language [= sLanguage]

sLanguage values

JScript	Language is Microsoft JScript.
javascript	Script is JavaScript.
vbs	Language is Microsoft Visual Basic Scripting Edition (VBScript).
vbscript	Script is VBScript.
XML	Script is Extensible Markup Language (XML).
language	Browser-supported language.

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

lastChild

Retrieves a reference to the last child in the childNodes collection of an object.

Syntax

HTML

N/A

Scripting

[oElement =] object.lastChild

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

lastIndexOf() [javaScript]

The lastIndexOf() method returns the position of the last occurrence of a specified string value, searching backwards from the specified position in a string. The lastIndexOf() method is case sensitive and returns -1 if the string value to search for never occurs.

Syntax

stringObject.lastIndexOf(searchvalue, fromindex)

Parameters

searchvalue	Required. Specifies a string value to search for
fromindex	Optional. Specifies where to start the search

Ex

In this example we will do different searches within a "Hello world!" string:
<script type="text/javascript">
var str="Hello world!"
document.write(str.lastIndexOf("Hello") + "
")
document.write(str.lastIndexOf("World") + "
")
document.write(str.lastIndexOf("world"))
</script>

The output of the code above will be:

0
-1
6

Applies to

String

lastModified [ie]

Retrieves the date the page was last modified, if the page supplies one.

Syntax

[sModified =] document.lastModified

Applies To

document

lastPage() [ie]

Displays the last page of records in the data set to which the table is bound.

Syntax

table.lastPage()

Applies to

Introduction to Data Binding, firstPage, nextPage, previousPage

layoutFlow [ie]

Sets or retrieves the direction and flow of the content in the object.

Syntax

CSS

{ layout-flow : *sFlow* }

Scripting

object.style.layoutFlow(v) [= *sFlow*]

sFlow values

horizontal	Default. Content in the object flows from left to right, and the next horizontal line is positioned underneath the previous line. This layout is used in most Roman-based documents
vertical-ideographic	Content in the object flows from top to bottom, and the next vertical line appears to the left of the previous one. This layout is used in East Asian typography

Applies To

a, abbr, acronym, address, b, big, blockquote, button, caption, center, cite, code, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, hr, i, input, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, marquee, menu, ol, option, p, plaintext, pre, q, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, ul, VAR

layoutGrid [ie]

Sets or retrieves the composite document grid properties that specify the layout of text characters.

Syntax

CSS

{ layout-grid : *sLayout* }

Scripting

object.style.layoutGrid [= *sLayout*]

sLayout values

mode	The range of mode values available to the layoutGridMode property.
type	The range of type values available to the layoutGridType property.
line	The range of line values available to the layoutGridLine property.
char	The range of character values available to the layoutGridChar property.

Applies To

a, abbr, acronym, address, applet, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

layoutGridChar [ie]

Sets or retrieves the size of the character grid used for rendering the text content of an element.

Syntax

CSS

{ layout-grid-char : *sCharSize* }

Scripting

object.style.layoutGridChar [= *sCharSize*]

sCharSize values

none	Default. No character grid is set.
auto	Largest character in the font of the element is used to set the character grid.
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	The value is a percentage derived from the dimensions of the parent object.

Applies To

blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

layoutGridLine [ie]

Sets or retrieves the gridline value used for rendering the text content of an element.

Syntax

CSS

{ layout-grid-line : *sLineSpace* }

Scripting

object.style.layoutGridLine [= *sLineSpace*]

sLineSpace values

none	Default. No grid line is set.
auto	Largest character in the font of the element is used to set the grid line
<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	The value is a percentage derived from the dimensions of the parent object.

Applies To

blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

layoutGridMode [ie]

Sets or retrieves whether the text layout grid uses two dimensions.

Syntax

CSS

{ layout-grid-mode : *sMode* }

Scripting

object.style.layoutGridMode [= *sMode*]

sMode values

both	Default. Both the char and line grid modes are enabled. This setting is necessary to fully enable the layout grid on an element.
none	No grid is used.
line	Only a line grid is used. This is recommended for use with inline elements, such as a span, to disable the horizontal grid on runs of text that act as a single entity in the grid layout.
char	Only a character grid is used. This is recommended for use with block-level elements, such as a blockQuote, where the line grid is intended to be disabled.

Applies To

a, abbr, acronym, address, applet, b, base, basefont, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

layoutGridType [ie]

Sets or retrieves the type of grid used for rendering the text content of an element.

Syntax

CSS

{ layout-grid-type : *sType* }

Scripting

object.style.layoutGridType [= *sType*]

sType values

loose	Default. Grid used for Japanese and Korean characters. In this mode, a constant width increment is applied to characters as follows: Wide characters and narrow kana characters are incremented to obtain an exact grid fit, as specified by the layoutGridChar property. Other narrow characters, except connected and cursive characters, are incremented by half of the increment amount applied to wide characters. Other characters, including connected and cursive characters, are not incremented, and behave as if no character grid is set.
-------	---

strict	Grid used for Chinese, as well as Japanese (Genko) and Korean characters. Only the ideographs, kanas, and wide characters are snapped to the grid. Other characters are rendered as usual, as though the layout-grid-mode attribute is set to none or line for text spans containing these characters. This mode also disables special text justification and character width adjustments normally applied to the element. Finally, if there is no line-break opportunity in a text span that exceeds the line boundary, the text is pushed to the next line and the last part of the previous line is left blank.
fixed	Grid used for monospaced layout. The layout rules are as follows: All noncursive characters are treated as equal; every character is centered within a single grid space by default. Runs of cursive characters are treated as strips the same as in a strict grid. Justification or any other character-width changing behaviors are disabled.

Applies To

blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

left

(css)

Sets or retrieves the position of the object relative to the left edge of the next positioned object in the document hierarchy.

Syntax

CSS

{ left : *sPosition* }

Scripting

object.style.left [= *sPosition*]

sPosition values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	Integer, followed by a percent sign (%). The value is a percentage of the width or height of the object.
<i>auto</i>	Default. Default position, according to the regular HTML layout of the page.

Applies To

frame, frameset, layoutrect, table, td, th, tr

left [ie]

(TextRectangle)

Sets or retrieves the left coordinate of the rectangle surrounding the object content.

Syntax

TextRectangle.left [= *iCoord*]

Applies To

TextRectangle

leftMargin [ie]

Sets or retrieves the left margin for the entire body of the page, in px overriding the default margin. The default value is 10.

Syntax

HTML

<body LEFTMARGIN = *sMargin*... >

Scripting

body.leftMargin [= *sMargin*]

Applies To

frame, frameset, layoutrect, table, td, th, tr

<legend>

Inserts a caption into the box drawn by the fieldSet object.

Ex

<fieldset>

<legend>Health information:</legend>

Height <input type="text" size="3" />

Weight <input type="text" size="3" />

</fieldset>

Optional Attributes

align (deprectaed in HTML 4.01, not supported in XHTML1.0)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

accessKey, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineBreak, lineHeight, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textJustify, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

length

(TextNode)

Retrieves the number of characters in a TextNode object.

Syntax

[*iLength* =] *TextNode.length*

Applies To

TextNode

length

(history)

Retrieves the number of elements in the History list.

Syntax

[*iLength* =] *history.length*

Applies To

history

length [javascript]

The length property sets or returns the number of elements in an object.

Syntax

Object.length

Applies To

Array, String

length

(all, achors, applets ...)

Sets or retrieves the number of objects in a collection.

Syntax

HTML

N/A

Scripting

object.length [= *iLength*]

Applies To

all, anchors, applets, areas, attributes, behaviorUrns, bookmarks, boundElements, cells, childNodes, children, comment, controlRange, elements, embeds, filters, forms, frames, images, imports, links, namespaces, options, pages, plugins, rows, rules, scripts, select, styleSheets, tBodies, TextRectangle, TextRange, window, FORM

letterSpacing

Sets or retrieves the amount of additional space between letters in the object.

Syntax

CSS

{ letterSpacing : *sSpacing* }

Scripting

object.style.letterSpacing [= *sSpacing*]

sSpacing values

normal	Default spacing
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

Denotes one item in a list.

Ex

```
<ol>
<li>Coffee</li>
<li>Tea</li>
</ol>
```

```
<ul>
<li>Coffee</li>
<li>Tea</li>
</ul>
```

Optional Attributes

type, value (deprecated, use CSS)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyle, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft,

posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

lineBreak [ie]

Sets or retrieves line-breaking rules for Japanese text.

Syntax

CSS

{ line-break : *sBreak* }

Scripting

object.style.lineBreak [= *sBreak*]

sBreak values

normal	Default. Normal line-breaking mode is applied
strict	Stricter line-breaking rules are enforced

Applies To

address, blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

lineHeight [ie]

Sets or retrieves the distance between lines in the object.

Syntax

CSS

{ line-height : *sHeight* }

Scripting

object.style.lineHeight [= *sHeight*]

sHeight values

normal	Default. Default height.
height	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
percentage	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

<link>

Enables the current document to establish links to external documents.

In HTML the <link> tag has no end tag, in XHTML the <link> tag must be properly closed.

Ex

```
<head>
<link rel="stylesheet" type="text/css" href="theme.css" />
</head>
```

Optional Attributes

Charset, href, hreflang, media, rel, rev, target, type

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, media, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition

link

Sets or retrieves the color of the document links for the object.

Syntax

HTML

<ELEMENT BORDERCOLOR = *sColor*... >

Scripting

object.borderColor [= *sColor*]

Applies To

frame, frameset, layoutrect, table, td, th, tr

links

Retrieves a collection of all a objects that specify the HREF property and all area objects in the document.

Syntax

[*oColl* =] document.links

[*oObject* =] document.links(*iIndex*)

Possible Values

<i>oColl</i>	Array of a objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Ex

This example shows how to display the HREF attribute of the third link defined in the document:

alert(document.links(2).href);

Applies To

document

link() [javascript]

The link() method is used to display a string as a hyperlink.

Syntax

stringObject.link()

Ex

In this example "Free Porn!" will be displayed as a hyperlink:

```
<script type="text/javascript">
var str= "Free Porn!"
document.write(str.link("http://www.somexxx.com"))
</script>
```

Applies to

String

linkColor [ie]

Sets or retrieves the color of the document links.

Syntax

HTML

<body link = *sColor*... >

Scripting

body.link [= *sColor*]

Applies To

body

<listing>

Renders text in a fixed-width font. Deprecated, use <pre>

listStyle

Sets or retrieves up to three separate listStyle properties of the object.

Syntax

CSS

{ list-style : *sStyle* }

Scripting

object.style.listStyle [= *sStyle*]

sStyle values

type	Any of the range of type values available to the listStyleType property.
position	Any of the range of position values available to the listStylePosition property.
image	Any of the range of image values available to the listStyleImage property.

Applies To

li, ol, runtimeStyle, style, UL

listStyleImage

Sets or retrieves a value that indicates which image to use as a list-item marker for the object.

Syntax

CSS

{ list-style-image : *sLocation* }

Scripting

object.style.listStyleImage [= *sLocation*]

sLocation values

none	Default. No image is specified
url(<i>sURL</i>)	Location of the image, where sURL is an absolute or relative URL

Applies To

currentStyle, li, ol, runtimeStyle, style, UL

listStylePosition

Sets or retrieves a variable that indicates how the list-item marker is drawn relative to the content of the object.

Syntax

CSS

{ list-style-position : *sPosition* }

Scripting

object.style.listStylePosition [= *sPosition*]

sPosition values

outside	Default. Marker is placed outside the text, and any wrapping text is not aligned under the marker
inside	Marker is placed inside the text, and any wrapping text is aligned under the marker

Applies To

currentStyle, li, ol, runtimeStyle, style, UL

listStyleType

Sets or retrieves the predefined type of the line-item marker for the object.

Syntax

CSS

{ list-style-type : *sType* }

Scripting

object.style.listStyleType [= *sType*]

sType values

disc	Default. Solid circles.
circle	Outlined circles.
square	Solid squares.
decimal	1, 2, 3, 4, and so on.
lower-roman	i, ii, iii, iv, and so on.
upper-roman	i, II, III, IV, and so on.
lower-alpha	a, b, c, d, and so on.
upper-alpha	A, B, C, D, and so on.
none	No marker is shown.

Applies To

currentStyle, li, ol, runtimeStyle, style, UL

LN2 [javascript]

Returns the natural logarithm of 2 (approx. 0.693)

Applies To

Math

LN10 [javascript]

Returns the natural logarithm of 10 (approx. 2.302)

Applies To
Math

location

Contains information about the current URL. The default property for the location object is location.href. For example, setting location='http://microsoft.com' is equivalent to setting location.href='http://microsoft.com'.

Properties

Property	Description
hash	Sets or retrieves the subsection of the href property that follows the number sign (#).
host	Sets or retrieves the hostname and port number of the location or URL.
hostname	Sets or retrieves the host name part of the location or URL.
href	Sets or retrieves the entire URL as a string.
pathname	Sets or retrieves the file name or path specified by the object.
port	Sets or retrieves the port number associated with a URL.
protocol	Sets or retrieves the protocol portion of a URL.
search	Sets or retrieves the substring of the href property that follows the question mark

Methods

assign, reload, replace

log() [javaScript]

The log() method returns the natural logarithm (base E) of a number.

Syntax

Math.log(x)

Parameter
x

Ex

```
<script type="text/javascript">
document.write(Math.log(2.7183) + "<br />")
document.write(Math.log(2) + "<br />")
document.write(Math.log(1) + "<br />")
document.write(Math.log(0) + "<br />")
document.write(Math.log(-1))
</script>
```

The output of the code above will be:

1.0000066849139877
0.6931471805599453
0

-Infinity

NaN

Applies to
Math

LOG2E [javascript]

Returns the base-2 logarithm of E (approx. 1.442)

Applies To
Math

LOG10E [javascript]

Returns the base-10 logarithm of E (approx. 0.434)

Applies To
Math

logicalXDPI [ie]

Retrieves the normal number of horizontal dots per inch (DPI) of the system's screen.

Syntax

[*nNormDPI* =] screen.logicalXDPI

Applies To
screen

logicalYDPI [ie]

Retrieves the normal number of vertical dots per inch (DPI) of the system's screen.

Syntax

[*nNormDPI* =] screen.logicalYDPI

Applies To
screen

longDesc

Sets or retrieves a Uniform Resource Identifier (URI) to a long description of the object.

Syntax

HTML

N/A

Scripting

object.longDesc(v) [= *sDesc*]

Applies To
frame, iframe, img

loop [ie]

Sets or retrieves the number of times a marquee will play.

Syntax

HTML

<marquee LOOP = *iLoop*... >

Scripting

marquee.loop [= *iLoop*]

iLoop values

0, -1	Loops infinitely
<i>count</i>	Number of times to loop

Applies To
marquee

loop

Sets or retrieves the number of times a sound or video clip will loop when activated.

Syntax

HTML

<ELEMENT LOOP = *iLoop*... >

Scripting

object.loop [= *iLoop*]

iLoop values

-1	Loops infinitely
0	Loops one time.

count Number of times to loop.

Applies To
bgsound, img, input, input type=image

lowsrc

Sets or retrieves a lower resolution image to display.

Syntax

HTML

<ELEMENT LOWSRC = *sURL*... >

Scripting

object.lowsrc [= *sURL*]

Applies To
img, input, input type=image

<map>

Contains coordinate data for client-side image maps. An image-map is an image with clickable regions.

Ex

<p>Click on one of the planets:</p>

```
<img src = "planets.gif"
width = "145" height = "126"
alt = "Planets"
usemap = "#planetmap" />

<map id = "planetmap"
name = "planetmap">
<area shape = "rect" coords = "0, 0, 82, 126"
href = "sun.htm" target = "_blank"
alt = "Sun" />
<area shape = "circle" coords = "90, 58, 3"
href = "mercur.htm" target = "_blank"
alt = "Mercury" />
<area shape = "circle" coords = "124, 58, 8"
href = "venus.htm" target = "_blank"
alt = "Venus" />
</map>
```

Required Attributes

id

Optional Attributes

name

Standard Attributes

class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

tabindex, accesskey, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, onfocus, onblur

Methods

addBehavior, appendChild, applyElement, attachEvent, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setAttribute, setAttributeNode, setCapture, swapNode

Style Properties and attributes

behavior, maxHeight, maxWidth, minHeight, minWidth

margin

Sets or retrieves the width of the top, right, bottom, and left margins of the object. Margins are always transparent. This is a composite property that specifies up to four width values, in the following order: top, right, bottom, left. If one width value is specified, it is used for all four sides. If two width values are specified, the first is used for the top and bottom borders, and the second is used for left and right borders. If three width values are specified, they are used for the top, right/left, and bottom borders, respectively. Negative margins are supported except for top and bottom margins on inline objects.

Syntax

CSS

{ margin : *sMargin* }

Scripting

object.style.margin [= *sMargin*]

sMargin values

top	Any of the range of values available to the marginTop property.
right	Any of the range of values available to the marginRight property.
bottom	Any of the range of values available to the marginBottom property.
left	Any of the range of values available to the marginLeft property.

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

marginBottom

Sets or retrieves the height of the bottom margin of the object.

Syntax

CSS

{ margin-bottom : *sHeight* }

Scripting

object.style.marginBottom [= *sHeight*]

sHeight values

auto	Default. Bottom margin is set equal to the top margin.
<i>height</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>percentage</i>	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

marginHeight

Sets or retrieves the top and bottom margin heights before displaying the text in a frame.

CSS

{ margin-height : *sHeight* }

Scripting

object.style.marginHeight [= *sHeight*]

Applies To

frame, iframe, layoutrect

marginLeft

Sets or retrieves the width of the left margin of the object.

Syntax

CSS

{ margin-left : *sWidth* }

Scripting

object.style.marginLeft [= *sWidth*]

sWidth values

auto	Default. Left margin is set equal to the top margin.
<i>Width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>percentage</i>	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

marginRight

Sets or retrieves the width of the right margin of the object.

Syntax

CSS

{ margin-right : *sWidth* }



Scripting

object.style.marginRight [= *sWidth*]

sWidth values

auto	Default. Right margin is set equal to the top margin.
<i>Width</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>percentage</i>	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

marginTop

Sets or retrieves the height of the top margin of the object.

Syntax

CSS

{ margin-top : *sHeight* }

Scripting

object.style.marginTop [= *sHeight*]

sWidth values

auto	Default. Top margin is set equal to the top margin.
<i>Height</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>percentage</i>	Integer, followed by a percent sign (%). The value is a percentage of the height of the parent object

Applies To

a, abbr, acronym, b, bdo, big, blockquote, body, button, caption, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, li, listing, marquee, menu, nobr, object, ol, p, plaintext, pre, q, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

marginWidth

Sets or retrieves the left and right margin widths before displaying the text in a frame.

Syntax

HTML

<ELEMENT MARGINWIDTH = *iWidth*... >

Scripting

object.marginWidth [= *iWidth*]

Applies To

frame, iframe, layoutrect

<marquee> [ie]

Creates a scrolling text marquee. This object is a Microsoft extension to HTML

Ex

<marquee DIRECTION=RIGHT BEHAVIOR=SCROLL SCROLLAMOUNT=10 SCROLLELAY=200>
This is a scrolling marquee.
</marquee>

match() [JavaScript]

The match() method searches for a specified value in a string. This method is similar to indexOf() and lastIndexOf(), but it returns the specified value, instead of the position of the string. The match() method is case sensitive and returns null if the string value to search for never occurs.

Syntax

stringObject.match(searchvalue)

Parameter

<i>searchvalue</i>	Required. Specifies a string value to search for
--------------------	--

Ex

```
<script type="text/javascript">
var str="Hello world!"
document.write(str.match("world") + "<br />")
document.write(str.match("World") + "<br />")
document.write(str.match("world") + "<br />")
document.write(str.match("world!"))
</script>
```

The output of the code above will be:

world
null
null
world!

Applies to

String

Math Object

The JavaScript Math object allows you to perform common mathematical tasks. It includes several mathematical constants and functions.

Methods

abs(x), acos(x), asin(x), atan(x), atan2(y,x), ceil(x), cos(x), exp(x), floor(x), log(x), max(x,y), min(x,y), pow(x,y), random, round(x), sin(x), sqrt(x), tan(x), toSource, valueOf()

Properties

Property, constructor, E, LN2, LN10, LOG2E, LOG10E, PI, prototype, SQRT1_2, SQRT2

max() [JavaScript]

The max() method returns the number with the highest value of two specified numbers.

Syntax

Math.max(x, y)

Parameters

<i>x</i>	Required. A number
<i>y</i>	Required. A number

Ex

```
<script type="text/javascript">
document.write(Math.max(5, 7) + "<br />")
document.write(Math.max(-3, 5) + "<br />")
document.write(Math.max(-3, -5) + "<br />")
document.write(Math.max(7.25, 7.30))
</script>
```

The output of the code above will be:

7
5
-3
7.3

Applies to

Math

maxHeight

Sets or retrieves the maximum height for displayable block level elements.

Syntax

HTML

{ max-height : *sMaxHeight* }

Scripting

object.style.maxHeight(v) [= *sMaxHeight*]

sMaxHeight values

<i>length</i>	Floating-point number followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	Integer followed by a % that specifies a percentage of the containing block height to use as the maximum height of the element. If the height of the containing block is not explicitly set, then the element has no maximum height and the maxHeight property is interpreted as 0%

Applies To

currentStyle, style, runtimeStyle, a, abbr, acronym, address, applet, area,

b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

maxLength

Sets or retrieves the maximum number of characters that the user can enter in a text control.

Syntax

HTML

<ELEMENT MAXLENGTH = *iLength*... >

Scripting

object.maxLength(*v*) [= *iLength*]

Applies To

input type=text, input type=password

maxWidth

Sets or retrieves the maximum width for displayable block level elements.

Syntax

CSS

{ max-width : *sMaxWidth* }

Scripting

object.style.maxWidth(*v*) [= *sMaxWidth*]

sMaxWidth values

<i>length</i>	Floating-point number followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	Integer followed by a % that specifies a percentage of the containing block height to use as the maximum height of the element. If the height of the containing block is not explicitly set, then the element has no maximum width and the property is interpreted as 0%

Applies To

currentStyle, style, runtimeStyle, a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

media

Sets or retrieves the media type.

Syntax

HTML

<ELEMENT MEDIA = *sType*... >

Scripting

object.media [= *sType*]

sType values

screen	Output is intended for computer screens
print	Output is intended for printed material and for on-screen documents viewed in Print Preview mode
all	Default. Applies to all devices

Applies To

styleSheet, link, STYLE

<menu>

Creates an unordered list of items. Deprecated

Ex

```
<menu>
<li>html</li>
<li>xhtml</li>
</menu>
```

Optional Attributes

compact (deprecated)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onfocus, onblur, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecor, textDecorBlink, textDecorLineThrough, textDecorNone, textDecorOverline, textDecorUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

menuArguments [ie]

Returns the window object where the context menu item was executed.

Syntax

[*oWindow* =] *external*.menuArguments

Applies To

external

mergeAttributes() [ie]

Copies all read/write attributes to the specified element.

Syntax

object.mergeAttributes(*oSource* [, *bPreserve*])

Parameters

<i>oSource</i>	Required. Pointer to an Object that specifies the attributes copied to the object that invokes mergeAttributes.
<i>bPreserve</i>	Optional. Available as of Microsoft Internet Explorer 5.5. Pointer to a Boolean that specifies one of the following values: true: Default. Preserve the identity of the object to which attributes are being merged; false: Do not preserve the identity of the object to which attributes are being merged.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

<meta>

Conveys hidden information about the document to the server and the client, such as descriptions and keywords for search engines and refresh rates.
In HTML the <meta> tag has no end tag, in XHTML the <meta> tag must be properly closed.
The <meta> tag always goes inside the head element.

Ex
Define keywords for search engines:
<meta name="keywords" content="html, DHTML, CSS, xml, XHTML, JavaScript, VBScript" />
Define a description of your web page:
<meta name="description" content="Free Web tutorials on html, CSS, xml, and XHTML" />
Define the last revision of your page:
<meta name="revised" content="Ariane Stolfi, 7/07/06" />
Refresh page every 5 seconds:
<meta http-equiv="refresh" content="5" />

- Required Attributes**
content
- Optional Attributes**
http-equiv, name, scheme
- Standard Attributes**
dir, lang, xml:lang
- Behaviors**
anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData
- Collection**
all, attributes, behaviorUrns, childNodes, children
- Methods**
contains, dragDrop, getAttribute, removeAttribute, setAttribute
- Style Properties and attributes**
backgroundPositionX, backgroundPositionY, layoutGrid, layoutGridMode, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth

method

Sets or retrieves how to send the form data to the server.
Syntax
<FORM METHOD = *sMethod*... >
Scripting
FORM.method [= *sMethod*]
sMethod values

get	Append the arguments to the action URL and open it as if it were an anchor
post	Send the data through an HTTP post transaction.

Applies To
frame, frameset, layoutrect, table, td, th, tr

Methods [ie]

Sets or retrieves the list of HTTP methods supported by the object.
Syntax
HTML

Scripting
A.Methods [= *sMethod*]
Applies To
A

mimeTypes

Not currently implemented by IE

min() [JavaScript]

The min() method returns the number with the lowest value of two specified numbers.
Syntax
Math.min(x, y)
Parameters

x	Required. A number
---	--------------------

y	Required. A number
---	--------------------

Ex
<script type="text/javascript">
document.write(Math.min(5, 7) + "
")
document.write(Math.min(-3, 5) + "
")
document.write(Math.min(-3, -5) + "
")
document.write(Math.min(7.25, 7.30))
</script>

The output of the code above will be:
5
-3
-5
7.25
Applies to
Math

minHeight

Sets or retrieves the minimum height for displayable block level elements.
Syntax
CSS
{ min-height : *sMinHeight* }
Scripting
object.style.minHeight(v) [= *sMinHeight*]
sMinHeight values

<i>length</i>	Floating-point number followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	Integer followed by a % that specifies a percentage of the containing block height to use as the minimum height of the element. If the height of the containing block is not explicitly set, then the element has no minimum height and the minHeight property is interpreted as 0%

Applies To
currentStyle, style, runtimeStyle, a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

minHeight (table)

Sets or retrieves the minimum height for an element.
Syntax
CSS
{ min-height : *sMinHeight* }
Scripting
object.style.minHeight(v) [= *sMinHeight*]
sMinHeight values

<i>length</i>	Floating-point number followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	Integer followed by a % that specifies a percentage of the containing block height to use as the minimum height of the element. If the height of the containing block is not explicitly set, then the element has no minimum height and the minHeight property is interpreted as 0%

Applies To
currentStyle, style, runtimeStyle, td, th, tr

minWidth

Sets or retrieves the minimum width for displayable block level element.
Syntax
CSS
{ min-width : *sMinWidth* }
Scripting
object.style.minWidth(v) [= *sMinWidth*]

sMinWidth values

<i>length</i>	Floating-point number followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px).
%	Integer followed by a % that specifies a percentage of the containing block height to use as the minimum height of the element. If the height of the containing block is not explicitly set, then the element has no minimum width and the <code>minWidth</code> property is interpreted as 0%

Applies To

`currentStyle`, `style`, `runtimeStyle`, `a`, `abbr`, `acronym`, `address`, `applet`, `area`, `b`, `bdo`, `big`, `blockquote`, `body`, `br`, `button`, `caption`, `center`, `cite`, `code`, `col`, `colgroup`, `custom`, `dd`, `del`, `dfn`, `div`, `dl`, `dt`, `em`, `embed`, `fieldset`, `font`, `form`, `frame`, `frameset`, `head`, `hn`, `hr`, `html`, `i`, `iframe`, `img`, `input type=button`, `input type=checkbox`, `input type=file`, `input type=hidden`, `input type=image`, `input type=password`, `input type=radio`, `input type=reset`, `input type=submit`, `input type=text`, `ins`, `isindex`, `kbd`, `label`, `legend`, `li`, `listing`, `map`, `marquee`, `menu`, `nextID`, `nobr`, `object`, `ol`, `option`, `p`, `plaintext`, `pre`, `q`, `rt`, `ruby`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `table`, `tbody`, `td`, `textarea`, `tfoot`, `th`, `thead`, `tr`, `tt`, `u`, `ul`, `var`, `xmp`

move() [ie]

Collapses the given text range and moves the empty range by the given number of units.

Syntax

iMoved = *TextRange*.move(*sUnit* [, *iCount*])

Parameters

<i>sUnit</i>	Required. Stringt hat specifies the units to move, using one of the following values:
character	Moves one or more characters.
word	Moves one or more words. A word is a collection of characters terminated by a space or some other white-space character, such as a tab.
sentence	Moves one or more sentences. A sentence is a collection of words terminated by a punctuation character, such as a period.
textedit	Moves to the start or end of the original range.
<i>iCount</i>	Optional. Integer that specifies the number of units to move. This can be positive or negative. The default is 1.

Applies to

`TextRange`

moveBy() [ie]

Moves the screen position of the window by the specified x and y offset values.

Syntax

window.moveBy(*iX*, *iY*)

Parameters

<i>iX</i>	Required. Integer that specifies the horizontal scroll offset in pixels. The value can be either positive or negative.
<i>iY</i>	Required. Integer that specifies the vertical scroll offset in pixels. The value can be either positive or negative.

Applies to

`window`

moveEnd() [ie]

Changes the end position of the range.

Syntax

iMoved = *TextRange*.moveEnd(*sUnit* [, *iCount*])

Parameters

<i>sUnit</i>	Required. Stringt hat specifies the units to move, using one of the following values:
character	Moves one or more characters.
word	Moves one or more words. A word is a collection of characters terminated by a space or some other white-space character, such as a tab.

	sentence	Moves one or more sentences. A sentence is a collection of words terminated by a punctuation character, such as a period.
	textedit	Moves to the start or end of the original range.
<i>iCount</i>		Optional. Integer that specifies the number of units to move. This can be positive or negative. The default is 1.

Applies to

`TextRange`

moveRow() [ie]

Moves a table row to a new position.

Syntax

oRow = *object*.moveRow(*iSource*, *iTarget*)

Parameters

<i>iSource</i>	Required. Integer that specifies the index in the rows collection of the table row that is moved. -1 Default.
<i>iTarget</i>	Required. Integer that specifies where the row is moved within the rows collection. -1 Default.

Applies to

`table`, `tbody`, `tfoot`, `thead`

moveStart() [ie]

Changes the start position of the range.

Syntax

iMoved = *TextRange*.moveStart(*sUnit* [, *iCount*])

Parameter

<i>sUnit</i>	Required. Stringt hat specifies the units to move, using one of the following values:
character	Moves one or more characters.
word	Moves one or more words, a collection of characters terminated by a space or some other white-space character, such as a tab.
sentence	Moves one or more sentences. A sentence is a collection of words terminated by a punctuation character, such as a period.
textedit	Moves to the start or end of the original range.
<i>iCount</i>	Optional. Integer that specifies the number of units to move. This can be positive or negative. The default is 1.

Applies to

`TextRange`

moveTo() [ie]

Moves the screen position of the upper-left corner of the window to the specified x and y position.

Syntax

window.moveTo(*iX*, *iY*)

Parameters

<i>iX</i>	Required. Integer that specifies the horizontal scroll offset in pixels. The value can be either positive or negative.
<i>iY</i>	Required. Integer that specifies the vertical scroll offset in pixels. The value can be either positive or negative

Applies to

`window`

moveToBookmark() [ie]

Moves to a bookmark. Returns true if Successfully moved to the bookmark and false if failed.

Syntax

bSuccess = *TextRange*.moveToBookmark(*sBookmark*)

Parameter

<i>sBookmark</i>	Required. Stringthat specifies the bookmark to move to.
------------------	---

Applies to

`TextRange`

moveToElementText() [ie]

Moves the text range so that the start and end positions of the range encompass the text in the given element.

Syntax

TextRange.moveToElementText(*oElement*)

Parameter

<i>oElement</i>	Required. Object that specifies the element object to move to.
-----------------	--

Applies to

TextRange

moveToPoint() [ie]

Moves the start and end positions of the text range to the given point.

Syntax

TextRange.moveToPoint(*iX*, *iY*)

Parameters

<i>iX</i>	Required. Integer that specifies the horizontal offset relative to the upper-left corner of the window, in pixels.
<i>iY</i>	Required. Integer that specifies the vertical offset relative to the upper-left corner of the window, in pixels.

Applies to

window

msInterpolationMode

Sets or retrieves the interpolation (resampling) method used to stretch images. (preliminary)

Syntax

CSS

{ -ms-interpolation-mode : *sMode* }

Scripting

object.style.msInterpolationMode(*v*) [= *sMode*]

sMode values

nearest-neighbor	Always use nearest neighbor interpolation mode
bicubic	Always use high-quality bicubic interpolation mode

Applies To

currentStyle, img, runtimeStyle, style

multiple

Sets or retrieves the Boolean value indicating whether multiple items can be selected from a list.

Syntax

HTML

<select MULTIPLE ... >

Scripting

select.multiple [= *bMultiple*]

bMultiple values

false	Default. Multiple items cannot be selected
true	Multiple items can be selected

Applies To

select

John Stuart Mill, em seus Princípios da Economia Política, diz:

“E de se duvidar que todas as invenções mecânicas até agora feitas aliviaram a labuta diária de algum ser humano”.

Tal não é também de modo algum a finalidade da maquinaria utilizada como capital. Igual a qualquer outro desenvolvimento da força produtiva do trabalho, ela se destina a baratear mercadorias e a encurtar a parte da jornada de trabalho que o trabalhador precisa para si mesmo, a fim de encompridar a outra parte da sua jornada de trabalho que ele dá de graça para o capitalista. Ela é meio de produção de mais-valia.

(...)

“A máquina, da qual parte a Revolução Industrial, substitui o trabalhador, que maneja uma única ferramenta, por um mecanismo, que opera com uma massa de ferramentas iguais ou semelhantes de uma só vez, e que é movimentada por uma única força motriz, qualquer que seja sua força. Ai temos a máquina, mas apenas como elemento simples da produção mecanizada.”

Viu-se que as forças produtivas decorrentes da cooperação e da divisão do trabalho nada custam ao capital. São forças naturais do trabalho social. Forças naturais como vapor, água etc., que são apropriadas para seu uso em processos produtivos, também nada custam. Mas assim como o homem precisa de um pulmão para respirar, ele precisa de uma “criação da mão humana” para consumir produtivamente forças da Natureza. Uma roda-d’água é necessária para explorar a força motriz da água; uma máquina a vapor, para explorar a elasticidade do vapor. O que ocorre com as forças naturais ocorre com a ciência. Uma vez descobertas, a lei do desvio da agulha magnética no campo de ação de uma corrente elétrica ou a lei da indução de magnetismo no ferro em torno do qual circula uma corrente elétrica já não custam um único centavo. Mas, para a exploração dessas leis pela telegrafia etc., é preciso uma aparelhagem muito cara e extensa. A ferramenta, como vimos, não é suprimida pela máquina. De uma ferramenta anã do organismo humano, ela aumenta em tamanho e número, tornando-se ferramenta de um mecanismo criado pelo homem. Em vez de trabalhar com a ferramenta manual, o capital põe o operário a trabalhar agora com uma máquina, que conduz por si mesma suas ferramentas. Se, por isso, está claro à primeira vista que a grande indústria tem de aumentar extraordinariamente a produtividade do trabalho mediante a incorporação de monstruosas forças da Natureza e das ciências naturais ao processo de produção, não está de modo algum igualmente claro que essa força produtiva ampliada não é, por outro lado, conseguida à custa de maior dispêndio de trabalho. Como qualquer outro componente do capital constante, a maquinaria não cria valor, mas transfere seu próprio valor ao produto para cuja feitura ela serve. À medida que tem valor e, por isso, transfere valor ao produto, ela se constitui num componente de valor do mesmo. Ao invés de barateá-lo, encarece-o proporcionalmente a seu próprio valor. É evidente que máquina e maquinaria desenvolvidas sistematicamente, o meio característico de trabalho da grande indústria, contêm desproporcionalmente mais valor em comparação com os meios de trabalho do artesanato e da manufatura.

—MARX, o capital cap XIII

maquinaria e grande indústria

namespace

Dynamically imports an element behavior into a document.

Properties

Property	Description
name	Retrieves the name of the namespace.
readyState	Retrieves the current state of the object.
urn	Sets or retrieves a Uniform Resource Name (URN) for a target document.

Methods

attachEvent, detachEvent, dolmport

NaN [javaScript]

The NaN property indicates that a value is “Not a Number”.

Syntax

NaN

Ex

```
<script type="text/javascript">
var test1="300"
var test2="Hello World!"
document.write(Number(test1)+ " <br />")
document.write(Number(test2)+ " <br />")
document.write(isNaN(test1)+ " <br />")
document.write(isNaN(test2))
</script>
```

The output of the code above will be:

300
NaN
false
true

Applies To

JavaScript Functions

name

(meta)

Sets or retrieves the value specified in the CONTENT attribute of the meta object.

Syntax

HTML

<meta NAME = *sName*... >

Scripting

meta.name [= *sName*]

sName values

name	Arbitrary value.
Description	Associated CONTENT attribute describes the containing document. Some search engines use this to provide the user with a document summary in the result of a search.
DownloadOptions	Microsoft Internet Explorer 6 for Microsoft Windows XP Service Pack 2 (SP2) and later. Associated CONTENT attribute determines what buttons are visible on the File Download dialog box.
Note	If more than one meta tag is set with the DownloadOptions value, Internet Explorer uses the last one.
Generator	Associated CONTENT attribute identifies the name of the application used to create the document.
Keywords	Associated CONTENT attribute consists of comma-delimited words describing the document. Some search engines use this to allow the user to perform a keyword search.
ProgID	Associated CONTENT attribute contains the program-matic identifier of the document's default editor.
Robots	Associated CONTENT attribute indicates whether the containing document should be indexed by search engines that recognize the meta object. Possible values include the following: all Search engines can index the containing document; noindex Search engines cannot index the containing document.

Template

Associated CONTENT attribute specifies the location of the template used to edit the document. Use this in conjunction with the ProgIDmeta object if the editor supports document templates.

Applies To

meta

name [ie]

(namespace)

Retrieves the name of the namespace.

Syntax

[*sName* =] namespace.name

Applies To

namespace

name

Sets or retrieves the name of an input parameter which is passed to an applet, embed, or object element.

Syntax

HTML

<param NAME = *p*... >

Scripting

param.name(*v*) [= *p*]

p values

false	Default. Object is opaque. It cannot be transparent.
true	Object can be transparent.

Applies To

param

name

(objects)

Sets or retrieves the name of the object or frame.

Syntax

HTML

<ELEMENT NAME = *sName*... >

Scripting

object.name [= *sName*]

Applies To

a, applet, attribute, button, embed, form, frame, frameset, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, link, map, object, rt, ruby, select, textarea

name

(window)

Sets or retrieves a value that indicates the window name.

Syntax

window.name [= *sName*]

Applies To

window

namedItem()

Retrieves an object or a collection from the specified collection.

Syntax

oItem = *object*.namedItem(*sName*)

Parameter

<i>sName</i>	Required. String that specifies the name or id property of the object to retrieve. A collection is returned if more than one match is made.
--------------	---

Applies to

all, anchors, applets, areas, boundElements, cells, elements, embeds, forms, images, links, mimeTypes, options, plugins, rows, scripts, tBodies, form, filters, frames, imports, styleSheets, TextRange, TextRectangle

namedRecordset() [ie]

Retrieves the recordset object corresponding to the named data member from a data source object (DSO).

Syntax

oRecordset = *object*.namedRecordset(*sQualifier* [, *sSubChapter*])

Parameters

<i>sQualifier</i>	Required. String that specifies the name of the data member, or an empty string, which indicates the default data member.
<i>sSubChapter</i>	Optional. String that specifies a path to a hierarchical data set.

Applies to
applet, xml, object

nameProp [ie]

Retrieves the file name specified in the href or src property of the object.

Syntax
HTML
N/A
Scripting
[sFileName =] object.*nameProp*

Applies To
a, img

namespaces [ie]

Retrieves a collection of namespace objects.

Syntax
[*oNamespace* =] document.namespaces
[*oObject* =] document.namespaces(*iIndex*)
Possible Values

<i>oNamespace</i>	Array of namespace objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To
document

navigate() [ie]

Loads the specified URL to the current window.

Syntax
window.navigate(sURL)
Parameters

<i>sURL</i>	Required. String that specifies the URL to display.
-------------	---

Applies to
window

NavigateAndFind() [ie]

Navigates to the specified URL and selects the specified text.

Syntax
external.NavigateAndFind(sLocation, sQuery, sTargetFrame)
Parameters

<i>sLocation</i>	Required. String that specifies the URL of a Web page.
<i>sQuery</i>	Required. String that specifies the text to highlight on the Web page specified by sLocation.
<i>sTargetFrame</i>	Required. String that specifies the name of the target frame to query.

Applies to
external

navigator

Contains information about the browser.

Properties

Property	Description
appCodeName	Retrieves the code name of the browser.
appMinorVersion	Retrieves the application's minor version value.
appName	Retrieves the name of the browser.
appVersion	Retrieves the platform and version of the browser.
browserLanguage	Retrieves the current browser language.

cookieEnabled	Retrieves whether client-side persistent cookies are enabled in the browser. Persistent cookies are those that are stored on the client-side computer.
cpuClass	Retrieves a string denoting the CPU class.
onLine	Retrieves a value indicating whether the system is in global offline mode.
platform	Retrieves the name of the user's operating system.
systemLanguage	Retrieves the default language used by the operating system.
userAgent	Retrieves a string equivalent to the HTTP user-agent request header.
userLanguage	Retrieves the operating system's natural language setting.

Collection
plugins
Methods
javaEnabled, taintEnabled
Objects
userProfile

new

The new operator creates an instance of a user-defined object type or of one of the built-in object types that has a constructor function.

Syntax
objectName = new *objectType*(*param1* [,*param2*] ...[,*paramN*])
Parameters

<i>objectName</i>	Name of the new object instance.
<i>objectType</i>	Object type. It must be a function that defines an object type.
<i>param1...paramN</i>	Property values for the object. These properties are parameters defined for the objectType function.

Ex
Creating a user-defined object type requires two steps:
1. Define the object type by writing a function.
2. Create an instance of the object with new.

Car.prototype.color = null;
car1.color = "black";
birthday.descricao = "The day you were born";

<nextID>

Creates unique identifiers that text editing software can read.
Deprecated

nextPage [ie]

Retrieves the position of the next page within a print template.

Syntax
[*p* =] *event.nextPage*
p values

"left"	The next page appears to the left of the binding.
"right"	The next page appears to the right of the binding.
""	An @page rule was not specified.

Applies To
event

nextPage() [ie]

Displays the next page of records in the data set to which the table is bound.

Syntax
table.nextPage()
Applies to
table

nextSibling

Retrieves a reference to the next child of the parent for the object.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object*.nextSibling

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

<noBR> [ie]

Renders text without line breaks. This object is an microsoft extension to HTML

Ex

<noBr>Here's a line of text I don't want to be broken . . .
here's the end of the line.</noBr>

Standard Attributes

id, class, title, style, dir, lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, attachEvent, click, componentFromPoint, contains, detachEvent, fireEvent, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, insertAdjacentHTML, insertAdjacentText, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, scrollIntoView, setAttribute, setAttributeNode, setCapture, setExpression

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

background, backgroundColor, backgroundImage, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, color, direction, display, filter, font, hasLayout, height, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, visibility, width, wordWrap, zoom

<noFrames>

Contains HTML for browsers that do not support FRAMESET elements.

Ex

<frameset cols = "25%, 25%, *">
<noframes>
<body>Your browser does not handle frames!</body>
</noframes>
<frame src ="anus.htm" />
<frame src ="sun.htm" />
<frame src ="cu.htm" />
</frameset>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

behaviorUrns

Methods

addBehavior, componentFromPoint, fireEvent, getAttributeNode, normalize, removeAttributeNode, removeBehavior, setAttributeNode

Style Properties and attributes

behavior, textAutospace, textUnderlinePosition, zoom

nodeName

Retrieves the name of a particular type of node.

Syntax

HTML

N/A

Scripting

[*sName* =] *object*.nodeName

sName values

<i>tagName</i>	Name of the element, also available through the tagName property
<i>attrName</i>	Name of the attribute, where the node is an attribute object
<i>#text</i>	Node is a TextNode object.

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

nodeType

Retrieves the type of the requested node.

Syntax

HTML

N/A

Scripting

[*iType* =] *object*.nodeType

iType values

1	Element node
3	Text node

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

nodeValue

Sets or retrieves the value of a node.

Syntax

HTML

N/A

Scripting

object.nodeValue [= *sValue*]

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

noHref

Sets or retrieves whether clicks in this region cause action.

Syntax

HTML

<area NOHREF ... >

Scripting

area.noHref [= *bHref*]

bHref values

false	Default. Clicks cause action.
true	Clicks do not cause action.

Applies To

area

noResize

Sets or retrieves whether the user can resize the frame.

Syntax

HTML

<ELEMENT BORDERCOLOR = *sColor*... >

Scripting

object.borderColor [= *sColor*]

Applies To

frame, frameset, layoutrect, table, td, th, tr

normalize()

Merges adjacent TextNode objects to produce a normalized document object model.

Syntax

object.normalize()

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

<noScript>

Specifies HTML to be displayed in browsers that do not support scripting.

Ex

```
<body>
...
<script type="text/javascript">
<!--
document.write("Nem a pau!")
//-->
</script>
<noscript>Your browser does not support JavaScript!</noscript>
...
</body>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

behaviorUrns

Methods

addBehavior, componentFromPoint, fireEvent, getAttributeNode, normalize, removeAttributeNode, removeBehavior, setAttributeNode

Style Properties and attributes

behavior, textAutospace, textUnderlinePosition, zoom

noShade

Sets or retrieves whether the horizontal rule is drawn with 3-D shading. Deprecated.

Syntax

HTML

<hr NOSHADE ... >

Scripting

hr.noShade [= *bShade*]

bShade values

false	Default. Horizontal rule is drawn with 3-D shading
true	Horizontal rule is drawn without 3-D shading

Applies To

hr

noWrap

Sets or retrieves whether the browser automatically performs wordwrap.

Syntax

HTML

<ELEMENT NOWRAP ... >

Scripting

object.noWrap [= *bWrap*]

bWrap values

false	Default. Browser automatically wraps the text
true	Browser does not wrap the text

Applies To

body, dd, div, dt, td, th

Number() [JavaScript]

The Number() function converts the value of an object to a number.

If the parameter is a Date object, the Number() function returns the number of milliseconds since midnight January 1, 1970 UTC. If the object's value cannot be converted to a number, the Number() function returns NaN.

Syntax

Number(*object*)

Ex

```
<script type="text/javascript">
var test1= new Boolean(true)
var test2= new Boolean(false)
var test3= new Date()
var test4= new String("999")
var test5= new String("999 888")

document.write(Number(test1)+ "<br />")
document.write(Number(test2)+ "<br />")
document.write(Number(test3)+ "<br />")
document.write(Number(test4)+ "<br />")
document.write(Number(test5)+ "<br />")
</script>
```

The output of the code above will be:

```
1
0
1151522510390
999
NaN
```

Applies to

all of the built-in JavaScript objects

<object>

Inserts an embebbed object into the HTML page. Use this element to add multimedia to your XHTML page. This element allows you to specify the data and parameters for objects inserted into HTML documents, and the code that can be used to display/manipulate that data.

```
Ex
<object classid="clsid:F08DF954-8592-11D1-B16A-00C0F0283628"
id="Slider1" width="100" height="50">
  <param name="BorderStyle" value="1" />
  <param name="MousePointer" value="0" />
  <param name="Enabled" value="1" />
  <param name="Min" value="0" />
  <param name="Max" value="10" />
</object>
```

Optional Attributes

align, archive, border, classid, codebase, codetype, data, declare, height, hspace, name, standby, type, usemap, vspace, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns

Events

accesskey, tabindex, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, insertAdjacentElement, insertBefore, mergeAttributes, namedRecordset, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeExpression, removeNode, replaceAdjacentText, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, cursor, direction, display, hasLayout, height, layoutGrid, layoutGridMode, left, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAutospace, textUnderlinePosition, top, unicodeBidi, visibility, wordWrap, zoom

object

Retrieves the contained object.

Syntax

HTML

N/A

Scripting

[*oObject* =] *object*.object

Applies To

object, APPLET

offscreenBuffering [ie]

Sets or retrieves whether objects are drawn offscreen before being made visible to the user.

Syntax

HTML

window.offscreenBuffering [= *vBuffering*]

vBuffering values

false	Boolean that disables offscreen buffering
true	Boolean that enables offscreen buffering
auto	Default. String that allows Microsoft Internet Explorer to decide when offscreen buffering is used.

Applies To

window

offsetHeight [ie]

Retrieves the height of the object relative to the layout or coordinate parent, as specified by the offsetParent property.

Syntax

HTML

N/A

Scripting

[*iHeight* =] *object*.offsetHeight

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

offsetLeft [ie]

Retrieves the calculated left position of the object relative to the layout or coordinate parent, as specified by the offsetParent property.

Syntax

HTML

N/A

Scripting

[*iCoord* =] *object*.offsetLeft

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextRange, tfoot, th, thead, tr, tt, u, ul, var, xmp

offsetParent [ie]

Retrieves a reference to the container object that defines the offsetTop and offsetLeft properties of the object.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object*.offsetParent

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

offsetTop [ie]

Retrieves the calculated top position of the object relative to the layout or coordinate parent, as specified by the offsetParent property.

Syntax

HTML

N/A

Scripting

[*iCoord* =] *object.offsetTop*

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextRange, tfoot, th, thead, tr, tt, u, ul, var, xmp

offsetWidth [ie]

Retrieves the width of the object relative to the layout or coordinate parent, as specified by the offsetParent property.

Syntax

HTML

N/A

Scripting

[*iWidth* =] *object.offsetWidth*

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

offsetX [ie]

Sets or retrieves the x-coordinate of the mouse pointer's position relative to the object firing the event.

Syntax

event.offsetX [= *iCoord*]

Applies To

event

offsetY [ie]

Sets or retrieves the y-coordinate of the mouse pointer's position relative to the object firing the event.

Syntax

event.offsetY [= *iCoord*]

Applies To

event

Draws lines of text as a numbered list.

Ex

```
<ol>
<li>Breja</li>
<li>Pinga</li>
</ol>
```

Optional Attributes

compact, start, type (deprecated in HTML and not supported in XHTML)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression,

removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyle, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

onabort [ie]

Fires when the user aborts the download of an image.

Syntax

onabort="SomeJavaScriptCode"

Supported by

img, image

onactivate [ie]

Fires when the object is set as the active element.

onafterprint [ie]

Fires on the object immediately after its associated document prints or previews for printing.

onafterupdate [ie]

Fires on a databound object after successfully updating the associated data in the data source object.

onbeforeactivate [ie]

Fires immediately before the object is set as the active element.

onbeforecopy [ie]

Fires on the source object before the selection is copied to the system clipboard.

onbeforecut [ie]

Fires on the source object before the selection is deleted from the document.

onbeforedeactivate [ie]

Fires immediately before the activeElement is changed from the current object to another object in the parent document.

onbeforeeditfocus [ie]

Fires before an object contained in an editable element enters a UI-activated state or when an editable container object is control selected.

onbeforepaste [ie]

Fires on the target object before the selection is pasted from the system clipboard to the document.

onbeforeprint [ie]

Fires on the object before its associated document prints or previews for printing.

onbeforeunload [ie]

Fires prior to a page being unloaded.

onbeforeupdate [ie]

Fires on a databound object before updating the associated data in the data source object.

onblur

Fires when the object loses the input focus.

Syntax

onblur="SomeJavaScriptCode"

Supported by

a, acronym, address, area, b, bdo, big, blockquote, button, caption, cite, dd, del, dfn, div, dl, dt, em, fieldset, form, frame, frameset, h1 to h6, hr, i, iframe, img, input, ins, kbd, label, legend, li, object, ol, p, pre, q, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var
button, checkbox, fileUpload, layer, frame, password, radio, reset, submit, text, textarea, window

onbounce [ie]

Fires when the behavior property of the marquee object is set to "alternate" and the contents of the marquee reach one side of the window.

oncellchange [ie]

Fires when data changes in the data provider.

onchange

Fires when the contents of the object or selection have changed.

Syntax

onchange="SomeJavaScriptCode"

Supported by

<input type="text", select, textarea>
fileUpload, select, text, textarea

onclick

Fires when the user clicks the left mouse button on the object.

Syntax

onclick="SomeJavaScriptCode"

Ex

```
<html>
<body>
Field1: <input type="text" id="field1" value="Ai!">
<br />
Field2: <input type="text" id="field2">
<br /><br />
Click the button below to copy the content of Field1 to Field2.
<br />
<button onclick="document.getElementById('field2').value=
document.getElementById('field1').value">Copy Text</button>
</body>
</html>
```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, object, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var
button, document, checkbox, link, radio, reset, submit

oncontextmenu [ie]

Fires when the user clicks the right mouse button in the client area, opening the context menu.

oncontrolselect [ie]

Fires when the user is about to make a control selection of the object.

oncopy [ie]

Fires on the source element when the user copies the object or selection, adding it to the system clipboard.

oncut [ie]

Fires on the source element when the object or selection is removed from the document and added to the system clipboard.

ondataavailable [ie]

Fires periodically as data arrives from data source objects that asynchronously transmit their data.

ondatachanged [ie]

Fires when the data set exposed by a data source object changes.

ondatasetcomplete [ie]

Fires to indicate that all data is available from the data source object.

ondblclick

Fires when the user double-clicks the object.

Syntax

ondblclick="SomeJavaScriptCode"

Ex

```
<html>
<body>
Field1: <input type="text" id="field1" value="Ai!">
<br />
Field2: <input type="text" id="field2">
<br /><br />
Click the button below to copy the content of Field1 to Field2.
<br />
<button ondblclick="document.getElementById('field2').value=
document.getElementById('field1').value">Copy Text</button>
</body>
</html>
```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, object, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var
document, link

ondeactivate [ie]

Fires when the activeElement is changed from the current object to another object in the parent document.

ondrag [ie]

Fires on the source object continuously during a drag operation.

ondragend [ie]

Fires on the source object when the user releases the mouse at the close of a drag operation.

ondragenter [ie]

Fires on the target element when the user drags the object to a valid drop target.

ondragleave [ie]

Fires on the target object when the user moves the mouse out of a valid drop target during a drag operation.

ondragover [ie]

Fires on the target element continuously while the user drags the object over a valid drop target.

ondragstart [ie]

Fires on the source object when the user starts to drag a text selection or selected object.

ondrop [ie]

Fires on the target object when the mouse button is released during a drag-and-drop operation.

onerror

Fires when an error occurs during object loading.

Syntax

onerror="SomeJavaScriptCode"

Ex

```

```

Supported by

img, object, style
window, image

onerrorupdate [ie]

Fires on a databound object when an error occurs while updating the associated data in the data source object.

onfilterchange [ie]

Fires when a visual filter changes state or completes a transition.

onfinish [ie]

Fires when marquee looping is complete.

onfocus

Fires when the object receives focus.

Syntax

onfocus="SomeJavaScriptCode"

Ex

```
<html>
<head>
<script type="text/javascript">
function setStyle(x)
{
document.getElementById(x).style.background="red"
}
</script>
</head>
<body>
First name: <input type="text"
onfocus="setStyle(this.id)" id="fname">
<br />
Last name: <input type="text"
onfocus="setStyle(this.id)" id="lname">

</body>
</html>
```

Supported by

a, acronym, address, area, b, bdo, big, blockquote, button, caption, cite, dd, del, dfn, div, dl, dt, em, fieldset, form, frame, frameset, h1 to h6, hr, i, iframe, img, input, ins, kbd, label, legend, li, object, ol, p, pre, q, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var button, checkbox, fileUpload, layer, frame, password, radio, reset, select, submit, text, textarea, window

onfocusin [ie]

Fires for an element just prior to setting focus on that element.

onfocusout [ie]

Fires for the current element with focus immediately after moving focus to another element.

onhelp [ie]

Fires when the user presses the F1 key while the browser is the active window.

onkeydown

Fires when the user presses a key.

Syntax

onkeydown="SomeJavaScriptCode"

Ex

```
<html>
<body>
<script type="text/javascript">
function noNumbers(e)
```

```
{
var keynum
var keychar
var numcheck

if(window.event) // IE
{
keynum = e.keyCode
}
else if(e.which) // Netscape/Firefox/Opera
{
keynum = e.which
}
keychar = String.fromCharCode(keynum)
numcheck = /\d/
return !numcheck.test(keychar)
}
</script>
<form>
<input type="text" onkeydown="return noNumbers(event)" />
</form>
</html>
```

Supported by

a, acronym, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, del, dfn, div, dt, em, fieldset, form, h1 to h6, hr, i, input, kbd, label, legend, li, map, object, ol, p, pre, q, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var document, image, link, textarea

onkeypress

Fires when the user presses an alphanumeric key. Internet Explorer uses event.keyCode to retrieve the character that was pressed and Netscape/Firefox/Opera uses event.which.

Syntax

onkeypress="SomeJavaScriptCode"

Ex

In this example the user cannot type numbers into the input field:

```
<html>
<body>
<script type="text/javascript">
function noNumbers(e)
{
var keynum
var keychar
var numcheck
if(window.event) // IE
{keynum = e.keyCode}
else if(e.which) // Netscape/Firefox/Opera
{keynum = e.which}
keychar = String.fromCharCode(keynum)
numcheck = /\d/
return !numcheck.test(keychar)
}
</script>
<form>
<input type="text" onkeypress="return noNumbers(event)" />
</form>
</html>
```

Supported by

a, acronym, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, del, dfn, div, dt, em, fieldset, form, h1 to h6, hr, i, input, kbd, label, legend, li, map, object, ol, p, pre, q, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var document, image, link, textarea

onkeyup

Fires when the user releases a key.

Syntax

onkeyup="SomeJavaScriptCode"

Ex

```
<html><head>
<script type="text/javascript">
function upperCase(x)
{
var y=document.getElementById(x).value
document.getElementById(x).value=y.toUpperCase()
}
</script></head>
```

```
<body>
Enter your name: <input type="text"
id="fname" onkeyup="upperCase(this.id)">
</body>
</html>
```

Supported by

a, acronym, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, del, dfn, div, dt, em, fieldset, form, h1 to h6, hr, i, input, kbd, label, legend, li, map, object, ol, p, pre, q, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, vardocument, image, link, textarea

onlayoutcomplete [ie]

Fires when the print or print preview layout process finishes filling the current LayoutRect object with content from the source document.

onLine [ie]

Retrieves a value indicating whether the system is in global offline mode.

Syntax

[*bOnLine* =] *object*.onLine

bOnLine values

false	System is in global offline mode
true	System is not in global offline mode

Applies To

clientInformation, navigator

onload

Fires immediately after the browser loads the object.

Syntax

onload="SomeJavaScriptCode"

Ex

In this example the text "Page is loaded" will be displayed in the status bar:

```
<html><head>
<script type="text/javascript">
function load()
{
window.status="Page is loaded"
}
</script></head>
<body onload="load()">
</body></html>
```

Supported by

body, frame, frameset, iframe, img, link, script
image, layer, window

onlosecapture [ie]

Fires when the object loses the mouse capture.

onmousedown

Fires when the user clicks the object with either mouse button.

Syntax

onmousedown="SomeJavaScriptCode"

Ex

In this example an alert box is displayed when clicking on the picture:

```

```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, varbutton, document, link

onmouseenter [ie]

Fires when the user moves the mouse pointer into the object.

onmouseleave [ie]

Fires when the user moves the mouse pointer outside the boundaries of the object.

onmousemove

Fires when the user moves the mouse over the object.

Syntax

onmousemove="SomeJavaScriptCode"

Ex

In the following example we will display an alert box when the user moves the mouse pointer over the image:

```

```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var

onmouseout

Fires when the user moves the mouse pointer outside the boundaries of the object.

Syntax

onmouseout="SomeJavaScriptCode"

Ex

In the following example we will display an alert box when the user moves the mouse pointer away from the image:

```

```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var
layer, link

onmouseover

Fires when the user moves the mouse pointer into the object.

Syntax

onmouseover="SomeJavaScriptCode"

Ex

In the following example we will display an alert box when the user moves the mouse pointer over the image:

```

```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, var
layer, link

onmouseup

Fires when the user releases a mouse button while the mouse is over the object.

Syntax

onmouseup="SomeJavaScriptCode"

Ex

In this example an alert box is displayed when the mouse button is released after clicking the picture:

```

```

Supported by

a, address, area, b, bdo, big, blockquote, body, button, caption, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, h1 to h6, hr, i, img, input, kbd, label, legend, li, map, ol, p, pre, samp, select, small, span, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, ul, varbutton, document, link

onmousewheel [ie]

Fires when the wheel button is rotated.

onmove [ie]

Fires when the object moves.

onmoveend [ie]

Fires when the object stops moving.

onmovestart [ie]

Fires when the object starts to move.

onpaste [ie]

Fires on the target object when the user pastes data, transferring the data from the system clipboard to the document.

onpropertychange [ie]

Fires when a property changes on the object.

onreadystatechange [ie]

Fires when the state of the object has changed.

onreadystatechange

Sets the event handler for asynchronous requests.

Syntax

XMLHttpRequest.onreadystatechange(vHandler) [= vHandler]

Applies To

XMLHttpRequest

onreset

Fires when the user resets a form.

Syntax

onreset="SomeJavaScriptCode"

Ex

In this example the form changes back to the default values and displays an alert box when the reset button is clicked:

```
<form onreset="alert('The form will be reset')">
  Firstname: <input type="text" name="fname" value="John" />
<br />
  Lastname: <input type="text" name="lname" />
<br /><br />
  <input type="reset" value="Reset">
</form>
```

Supported by

form

onresize

Fires when the size of the object is about to change.

Syntax

onresize="SomeJavaScriptCode"

Ex

```
<body onresize="alert('You have changed the size of the window')">
</body>
```

Supported by

a, address, b, big, blockquote, body, button, cite, code, dd, dfn, div, dl, dt, em, fieldset, form, frame, h1 to <h6, hr, i, img, input, kbd, label, legend, li, object, ol, p, pre, samp, select, small, span, strong, sub, sup, table, textarea, tt, ul, var window

onresizeend [ie]

Fires when the user finishes changing the dimensions of the object in a control selection.

onresizestart [ie]

Fires when the user begins to change the dimensions of the object in a control selection.

onrowenter [ie]

Fires to indicate that the current row has changed in the data source and new data values are available on the object.

onrowexit [ie]

Fires just before the data source control changes the current row in the object.

onrowsdelete [ie]

Fires when rows are about to be deleted from the recordset.

onrowsinserted [ie]

Fires just after new rows are inserted in the current recordset.

onscroll [ie]

Fires when the user repositions the scroll box in the scroll bar on the object.

onselect [ie]

Fires when the current selection changes.

Syntax

onselect="SomeJavaScriptCode"

Ex

In this example an alert box will be displayed if some of the text is selected:

```
<form>
  Select text: <input type="text" value="Hello world!"
  onselect="alert('You have selected some of the text.')">
<br />
  Select text: <textarea cols="20" rows="5"
  onselect="alert('You have selected some of the text.')">
  Hello world!</textarea>
</form>
```

Supported by

input type="text", textarea
text, textarea

onselectionchange [ie]

Fires when the selection state of a document changes.

onselectstart [ie]

Fires when the object is being selected.

onstart [ie]

Fires at the beginning of every loop of the marquee object.

onstop [ie]

Fires when the user clicks the Stop button or leaves the Web page.

onsubmit

Fires when a FORM is about to be submitted.

Syntax

onsubmit="SomeJavaScriptCode"

Ex

In this example an alert box displays when a submit box is used:

```
<form name="testform" action="jsref_onsubmit.asp"
  onsubmit="alert('Hello ' + testform.fname.value + '!')">
  What is your name?<br />
  <input type="text" name="fname" />
  <input type="submit" value="Submit" />
</form>
```

Supported by

form

onunload

Fires immediately before the object is unloaded.

Syntax

onunload="SomeJavaScriptCode"

Ex

```
<body onunload="alert('The onunload event was triggered')">
</body>
```

Supported by

body, frameset
window

open()

(XMLHttpRequest)

This method works in two ways. It opens a document to collect the output of the write and writeln methods. In this case, only the first two parameters, url and name are used. When values for the additional parameters are specified, this method opens a window in the same way as the window.open method for the window object.

Syntax

XMLHttpRequest.open(sMethod, sUrl [,bAsync] [,sUser] [,sPassword])

Parameters

sMethod	Required. String that specifies the HTTP method used to open the connection: GET, POST, or head only. This parameter is not case-sensitive.
sUrl	Required. String that specifies either the absolute or a relative URL of the Extensible Markup Language (XML) data or server-side XML Web services.
bAsync	Optional. Variant that specifies true for asynchronous operation (the call returns immediately), or false otherwise. If true, assign a callback handler to the onreadystatechange property to determine when the call has completed. If not specified, the default is true.
sUser	Optional. Variant that specifies the name of the user for authentication. If this parameter is null ("") or missing and the site requires authentication, the component displays a logon window.
sPassword	Optional. Variant that specifies the password for authentication. This parameter is ignored if the user parameter is null ("") or missing.

Applies to
XMLHttpRequest

open() (document)

Opens a new window and loads the document specified by a given URL.

Syntax

XMLHttpRequest.open(sMethod, sUrl [,bAsync] [,sUser] [, sPassword])

Parameter

sMethod	Required. String that specifies the HTTP method used to open the connection: GET, POST, or head only. This parameter is not case-sensitive.
sUrl	Required. String that specifies either the absolute or a relative URL of the Extensible Markup Language (XML) data or server-side XML Web services.
bAsync	Optional. Variant that specifies true for asynchronous operation (the call returns immediately), or false otherwise. If true, assign a callback handler to the onreadystatechange property to determine when the call has completed. If not specified, the default is true.
sUser	Optional. Variant that specifies the name of the user for authentication. If this parameter is null ("") or missing and the site requires authentication, the component displays a logon window.
sPassword	Optional. Variant that specifies the password for authentication. This parameter is ignored if the user parameter is null ("") or missing.

Applies to
window

open() (window)

Assigns method, destination URL, and other optional attributes of a pending request.

Syntax

oNewWindow = window.open([sURL] [,sName] [,sFeatures] [, bReplace])

Parameter

sURL	Optional. String that specifies the URL of the document to display. If no URL is specified, a new window with about: blank is displayed.
sName	Optional. String that specifies the name of the window. This name is used as the value for the TARGET attribute on a form or an a element.
_blank	The sURL is loaded into a new, unnamed window.

_media	The sURL is loaded into the HTML content area of the Media Bar. Available in Microsoft Internet Explorer 6 or later.
_parent	The sURL is loaded into the current frame's parent. If the frame has no parent, this value acts as the value _self.
_search	Available in Internet Explorer 5 and later. The sURL is opened in the browser's search pane.
_self	The current document is replaced with the specified sURL.
_top	sURL replaces any framesets that may be loaded. If there are no framesets defined, this value acts as the value _self.

<i>sFeatures</i>	Optional. This String parameter is a list of items separated by commas. Each item consists of an option and a value, separated by an equals sign (for example, "fullscreen=yes, toolbar=yes"). The following features are supported.
channelmode = { yes no 1 0 }	Specifies whether to display the window in theater mode and show the channel band. The default is no.
directories = { yes no 1 0 }	Specifies whether to add directory buttons. The default is yes.
fullscreen = { yes no 1 0 }	Specifies whether to display the browser in full-screen mode. The default is no.
height = number	Specifies the height of the window, in pixels. The minimum value is 100.
left = number	Specifies the left position, in pixels. This value is relative to the upper-left corner of the screen. The value must be greater than or equal to 0.
location = { yes no 1 0 }	Specifies whether to display the input field for entering URLs directly into the browser. The default is yes.
menubar = { yes no 1 0 }	Specifies whether to display the menu bar. The default is yes.
resizable = { yes no 1 0 }	Specifies whether to display resize handles at the corners of the window. The default is yes.
scrollbars = { yes no 1 0 }	Specifies whether to display horizontal and vertical scroll bars. The default is yes.
status = { yes no 1 0 }	Specifies whether to add a status bar at the bottom of the window. The default is yes.
titlebar = { yes no 1 0 }	Specifies whether to display a title bar for the window. This parameter is ignored unless the calling application is an HTML Application or a trusted dialog box. The default is yes.
toolbar = { yes no 1 0 }	Specifies whether to display the browser toolbar, making buttons such as Back, Forward, and Stop available. The default is yes.
top = number	Specifies the top position, in pixels. This value is relative to the upper-left corner of the screen. The value must be greater than or equal to 0.
width = number	Sets the width of the window, in pixels. The minimum value is 100.

<i>bReplace</i>	Optional. When the sURL is loaded into the same window, this Boolean parameter specifies whether the sURL creates a new entry or replaces the current entry in the window's history list. true: sURL replaces the current document in the history list false: sURL creates a new entry in the history list.
-----------------	---

Applies to
window

opener [ie]

Sets or retrieves a reference to the window that created the current window.

Syntax

window.opener [= sWindow]

Applies To

window

<optgroup>

Allows authors to group choices logically in a select element.

Ex

```
<select>
<optgroup label="Swedish Cars">
<option value ="volvo">Volvo</option>
<option value ="saab">Saab</option>
</optgroup>
<optgroup label="German Cars">
<option value ="mercedes">Mercedes</option>
<option value ="audi">Audi</option>
</optgroup>
</select>
```

Required Attributes

label

Optional Attributes

disabled

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Events

tabindex, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

<option>

Denotes one choice in a select element.

In HTML the <option> tag has no end tag, in XHTML the

<option> tag must be properly closed.

The <option> tag can be used without any attributes, but you usually need the value attribute, which indicates what is sent to the server. Use this tag in conjunction with the select element, elsewhere it is meaningless.

Ex

```
<select>
<option value ="volvo">Volvo</option>
<option value ="saab">Saab</option>
<option value ="opel" selected="selected">Opel</option>
<option value ="audi">Audi</option>
</select>
```

Optional Attributes

disabled, label, selected, value

Standard Attributes

id, class, title, style, dir, lang, xml:lang, tabindex

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

backgroundAttachment, backgroundColor, backgroundPositionX, backgroundPositionY, behavior, clear, color, direction, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight,

minWidth, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition, unicodeBidi, width, wordWrap, writingMode, zoom

options

Retrieves a collection of the option objects in a select object.

Syntax

[oColl =] SELECT.options

[oObject =] SELECT.options(vIndex [, iSubIndex])

Possible Values

<i>oColl</i>	Array of option objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To

select

outerHTML [ie]

Sets or retrieves the object and its content in HTML.

Syntax

N/A

Scripting

object.outerHTML [= sHTML]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, bgsound, big, blockquote, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, hn, hr, HTML Comment, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, wbr, xmp

outerText [ie]

Sets or retrieves the text of the object.

Syntax

HTML

N/A

Scripting

object.outerText [= sTxt]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, bgsound, big, blockquote, br, button, center, cite, code, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, wbr, xmp

overflow

Sets or retrieves a value indicating how to manage the content of the object when the content exceeds the height or width of the object.

Syntax

CSS

{ overflow : sOverflow }

Scripting

object.style.overflow [= sOverflow]

sOverflow values

visible	Default. Content is not clipped and scroll bars are not added.
scroll	Content is clipped and scroll bars are added, even if the content does not exceed the dimensions of the object.
hidden	Content that exceeds the dimensions of the object is not shown.
auto	Content is clipped and scrolling is added only when necessary.

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, html, hn, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, menu, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, textarea, tt, u, ul, var, xmp

overflowX

Sets or retrieves how to manage the content of the object when the content exceeds the width of the object.

Syntax

CSS
{ overflow-x : *sOverflow* }
Scripting
object.style.overflowX [= *sOverflow*]

sOverflow values

visible	Default. Content is not clipped and scroll bars are not added. Elements are clipped to the size of the containing window or frame.
scroll	Content is clipped and scroll bars are added, even if the content does not exceed the dimensions of the object.
hidden	Content that exceeds the dimensions of the object is not shown.
auto	Content is clipped and scrolling is added only when necessary.

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, menu, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, textarea, tt, u, ul, var, xmp

overflowY

Sets or retrieves how to manage the content of the object when the content exceeds the height of the object.

Syntax

CSS
{ overflow-y : *sOverflow* }
Scripting
object.style.overflowY [= *sOverflow*]

sOverflow values

visible	Default. Content is not clipped and scroll bars are not added. Elements are clipped to the size of the containing window or frame.
scroll	Content is clipped and scroll bars are added, even if the content does not exceed the dimensions of the object.
hidden	Content that exceeds the dimensions of the object is not shown.
auto	Content is clipped and scrolling is added only when necessary.

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, center, cite, code, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, menu, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, textarea, tt, u, ul, var, xmp

ownerDocument

Retrieves the document object associated with the node.

Syntax

HTML
N/A

Scripting

[*oDoc* =] *object*.ownerDocument

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

owningElement [ie]

Retrieves the next object in the HTML hierarchy.

Syntax

[*oElement* =] *styleSheet*.owningElement

Applies To

styleSheet

<p>

Denotes a paragraph.

Ex

<p>This is some text in a very short paragraph</p>

Optional Attributes

align (deprectated in HTML and not supported in XHTML)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

Packages

Keyword used to creaaate a javaObject. The javaObject object is an instance of a Java class that is created in or passed to JavaScript. javaObject is a wrapper for the instance; all references to the class instance are made through the javaObject.

Syntax

new Packages.javaClass(parameterList)

Parameters

javaClass	the fully-specified name of the object's Java class
parameterList	An optional list of parameters, specified by the constructor of the Java class.

Ex

The following code creates the javaObject theString, which is an instance of the class java.lang.String:
var theString = new Packages.java.lang.String("Hello, world");

padding

Sets or retrieves the amount of space to insert between the object and its margin or, if there is a border, between the object and its border.

Syntax

CSS

{ padding : sPadding }

Scripting

object.style.padding [= sPadding]

sPadding values

length	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	A percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, td, textarea, th, tt, u, ul, var, xmp

paddingBottom

Sets or retrieves the amount of space to insert between the bottom border of the object and the content.

Syntax

CSS

{ padding-bottom : sPadding }

Scripting

object.style.paddingBottom [= sPadding]

sPadding values

length	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	A percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, ul, var, xmp

paddingLeft

Sets or retrieves the amount of space to insert between the left border of the object and the content.

Syntax

CSS

{ padding-left : sPadding }

Scripting

object.style.paddingLeft [= sPadding]

sPadding values

length	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	A percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, ul, var, xmp

paddingRight

Sets or retrieves the amount of space to insert between the right border of the object and the content.

Syntax

{ padding-right : sPadding }

Scripting

object.style.paddingRight [= sPadding]

p

sPadding values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>%</i>	A percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, ul, var, xmp

paddingTop

Sets or retrieves the amount of space to insert between the top border of the object and the content.

Syntax

CSS

{ padding-top : *sPadding* }

Scripting

object.style.paddingTop [= *sPadding*]

sPadding values

<i>length</i>	Floating-point number, followed by an absolute (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
<i>%</i>	A percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, ul, var, xmp

page

Represents an @page rule within a styleSheet.

Properties

Property	Description
pseudoClass	Retrieves a string that identifies the pseudo class of the page or pages an @page rule applies to.
selector	Retrieves a string that identifies which page or pages an @page rule applies to.

Applies To

pages, styleSheet

pageBreakAfter

Sets or retrieves a value indicating whether a page break occurs after the object.

Syntax

CSS

{ page-break-after : *sBreak* }

Scripting

object.style.pageBreakAfter [= *sBreak*]

sBreak values

always	Always insert a page break after the object.
auto	Default. Neither force nor forbid a page break after the object.
empty	string Do not insert a page break.
left	Currently works the same as always.
right	Currently works the same as always.

Applies To

blockquote, body, button, caption, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, isindex, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, tbody, tfoot, th, thead, tr, ul, xmp

pageBreakBefore

Sets or retrieves a string indicating whether a page break occurs before the object.

Syntax

CSS

{ page-break-before : *sBreak* }

Scripting

object.style.pageBreakBefore [= *sBreak*]

sBreak values

always	Always insert a page break before the object.
auto	Default. Neither force nor forbid a page break before the object.
empty	string Do not insert a page break.
left	Currently works the same as always.
right	Currently works the same as always.

Applies To

blockquote, body, button, caption, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, isindex, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, tbody, tfoot, th, thead, tr, ul, XM

pages [ie]

Retrieves a collection of page objects, which represent @page rules in a styleSheet.

Syntax

[*oColl* =] styleSheet.pages

[*oObject* =] styleSheet.pages(*iIndex*)

Possible Values

<i>oColl</i>	Array of page objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

styleSheet

palette [ie]

Retrieves the palette used for the embedded document.

Syntax

HTML

N/A

Scripting

[*sPalette* =] *EMBED*.palette

Applies To

EMBED

<param>

Sets the initial value of a property for an applet, embed, or object element.

In HTML the <param> tag has no end tag, in XHTML the <param> tag must be properly closed.

Ex

```
<object classid="clsid:F08DF954-8592-11D1-B16A-00C0F0283628"
id="Slider1" width="100" height="50">
  <param name="BorderStyle" value="1" />
  <param name="MousePointer" value="0" />
  <param name="Enabled" value="1" />
  <param name="Min" value="0" />
  <param name="Max" value="10" />
</object>
```

Required Attributes

name

Optional Attributes

type, value, valuetype

Standard Attributes

id

parent [ie]

Retrieves the parent of the window in the object hierarchy.

Syntax

[*oElement* =] *window*.parent

Applies To

window

parentElement [ie]

Retrieves the parent object in the object hierarchy.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object*.parentElement

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

parentElement() [ie]

Retrieves the parent element for the given text range.

Syntax

oElement = *TextRange*.parentElement()

Applies to

TextRange

parentNode

Retrieves the parent object in the document hierarchy.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object*.parentNode

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

parentStyleSheet [ie]

Retrieves the style sheet that imported the current style sheets.

Syntax

[*sStyleSheet* =] *styleSheet*.parentStyleSheet

Applies To

styleSheet

parentTextEdit [ie]

Retrieves the container object in the document hierarchy that can be used to create a TextRange containing the original object.

Syntax

HTML

N/A

Scripting

[*oObject* =] *object*.parentTextEdit

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

parentWindow [ie]

Retrieves a reference to the container object of the window.

Syntax

[*oParent* =] *document*.parentWindow

Applies To

document

parse() [JavaScript]

The parse() method takes a date string and returns the number of milliseconds since midnight of January 1, 1970.

Syntax

Date.parse(*datestring*)

Parameter

<i>datestring</i>	Required. A string representing a date
-------------------	--

Ex

In this example we will get how many milliseconds there are from 1970/01/01 to 2005/07/08:

```
<script type="text/javascript">
var d = Date.parse("Jul 8, 2005")
document.write(d)
</script>
```

The output of the code above will be:

1120773600000

Applies to

Date

parseFloat() [JavaScript]

The parseFloat() function parses a string and returns a floating point number.

This function determines if the first character in the specified string is a number. If it is, it parses the string until it reaches the end of the number, and returns the number as a number, not as a string.

Syntax

parseFloat(*string*)

Ex

```
<script type="text/javascript">
document.write(parseFloat("10") + " , ")
document.write(parseFloat("10.00") + " , ")
document.write(parseFloat("10.33") + " , ")
document.write(parseFloat("34 45 66") + " , ")
document.write(parseFloat(" 60 ") + " , ")
document.write(parseFloat("40 years") + " , ")
document.write(parseFloat("He was 40") )
</script>
```

The output of the code above will be:

10, 10, 10.33, 34, 60, 40, NaN

Applies to

all of the built-in JavaScript objects

parseInt() [JavaScript]

The parseInt() function parses a string and returns an integer.

The radix parameter is used to specify which numeral system to be used, for example, a radix of 16 (hexadecimal) indicates that the number in the string should be parsed from a hexadecimal number to a decimal number.

If the radix parameter is omitted, JavaScript assumes the following:

If the string begins with "0x", the radix is 16 (hexadecimal)

If the string begins with "0", the radix is 8 (octal). This feature is deprecated

If the string begins with any other value, the radix is 10 (decimal)

Syntax

parseInt(*string*, *radix*)

Parameters

<i>string</i>	Required. The string to be parsed
<i>radix</i>	Optional. A number (from 2 to 36) that represents the numeral system to be used

Ex

```
<script type="text/javascript">
```

```
document.write(parseInt("10") + " , ")
document.write(parseInt("10.00") + " , ")
document.write(parseInt("10.33") + " , ")
document.write(parseInt("34 45 66") + " , ")
document.write(parseInt(" 60 ") + " , ")
document.write(parseInt("40 years") + " , ")
document.write(parseInt("He was 40") + " , ")
document.write(parseInt("10")+ " , ")
document.write(parseInt("10", 10)+ " , ")
document.write(parseInt("010")+ " , ")
document.write(parseInt("10", 8)+ " , ")
document.write(parseInt("0x10")+ " , ")
document.write(parseInt("10", 16)+ " , ")
</script>
```

The output of the code above will be:
10, 10, 10, 34, 60, 40, NaN, 10, 10, 8, 8, 16, 16

Applies to
all of the built-in JavaScript objects

pasteHTML() [ie]

Pastes HTML text into the given text range, replacing any previous text and HTML elements in the range.

Syntax
TextRange.pasteHTML(*sHTMLText*)
Parameter

<i>sMessage</i>	Required. String that specifies the HTML text to paste. The string can contain text and any combination of the HTML tags described in HTML Elements.
-----------------	--

Applies to
TextRange

pathname [ie]

Sets or retrieves the file name or path specified by the object.

Syntax
HTML
N/A
Scripting
object.pathname [= *sName*]
Applies To
a, area, location

PhishingEnabled()

Returns a boolean value that determines whether Microsoft Phishing Filter is enabled.

Syntax
fEnabled = *window.external*.PhishingEnabled()
Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to
external

PI [javascript]

Returns PI (approx. 3.14159)
Applies To
Math

pixelBottom [ie]

Sets or retrieves the bottom position of the object.

Syntax
HTML
N/A
Scripting
object.style.pixelBottom [= *iBottom*]
Applies To
a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head,

hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

pixelHeight [ie]

Sets or retrieves the height of the object.

Syntax
HTML
N/A
Scripting
object.style.pixelHeight [= *iHeight*]
Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

pixelLeft [ie]

Sets or retrieves the left position of the object.

Syntax
HTML
N/A
Scripting
object.style.pixelLeft [= *iLeft*]
Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

pixelRight [ie]

Sets or retrieves the right position of the object

Syntax
HTML
N/A
Scripting
object.style.pixelRight [= *iRight*]
Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

pixelTop [ie]

Sets or retrieves the top position of the object.

Syntax
HTML
N/A
Scripting
object.style.pixelTop [= *iTop*]
Applies To
a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset,

input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

pixelWidth [ie]

Sets or retrieves the width of the object.

Syntax

HTML

N/A

Scripting

object.pixelWidth [= *iWidth*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

plaintext

Not currently supported.

platform [ie]

Retrieves the name of the user's operating system.

Syntax

[*sPlatform* =] *object*.platform

sPlatform values

HP-UX	HP Unix-based computers.
MacPPC	Macintosh PowerPC-based computers.
Mac68K	Macintosh 68K-based computers.
SunOS	Solaris-based computers.
Win32	Microsoft Windows 32-bit platform.
Win16	Windows 16-bit platform.
WinCE	Windows CE platform.

Applies To

clientInformation, navigator

plugins [ie]

Retrieves a collection of all embed objects within the document.

Syntax

[*oColl* =] object.plugins

[*oObject* =] object.plugins(*iIndex*)

Possible Values

<i>oColl</i>	Array that is empty.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer that specifies the zero-based index of the item to be returned.

Applies To

clientInformation, navigator

pluginspage [ie]

Retrieves the URL of the plug-in used to view an embedded document.

Syntax

HTML

<EMBED PLUGINSPAGE = *sURL*... >

Scripting

[*sURL* =] *EMBED*.pluginspage

Applies To

EMBED

pop() [JavaScript]

The pop() method is used to remove and return the last element of an array, changing the length of the array.

Syntax

arrayObject.pop()

Ex

In this example we will create an array, and then remove the last element of the array. Note that this will also change the length of the array:

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Marx"
arr[1] = "Lenin"
arr[2] = "Trotsky"
document.write(arr + "<br />")
document.write(arr.pop() + "<br />")
document.write(arr)
</script>
```

The output of the code above will be:

Marx, Lenin, Trotsky

Trotsky

Marx, Lenin

Applies to

Array

popup [ie]

A special type of overlapped window typically used for dialog boxes, message boxes, and other temporary windows that appear separate from an application's main window.

Ex

```
<script LANGUAGE="JScript">
var oPopup = window.createPopup();
var oPopupBody = oPopup.document.body;
```

```
oPopupBody.innerHTML = "Display some <B>HTML</B> here.";
oPopup.show(100, 100, 200, 50, document.body);
```

</script>

Properties

document	Retrieves the HTML document in a given popup window.
isOpen	Retrieves a value indicating whether the popup window is open.

Methods

hide, show

pow() [JavaScript]

The pow() method returns the value of x to the power of y (x^y).

Syntax

Math.min(x, y)

Parameters

<i>x</i>	Required. A number
<i>y</i>	Required. A number

Ex

```
<script type="text/javascript">
document.write(Math.pow(0, 0) + " ")
document.write(Math.pow(0, 1) + " ")
document.write(Math.pow(1, 1) + " ")
document.write(Math.pow(1, 10) + " ")
document.write(Math.pow(2, 3) + " ")
document.write(Math.pow(-2, 3) + " ")
document.write(Math.pow(2, 4) + " ")
document.write(Math.pow(-2, 4) )
</script>
```

The output of the code above will be:

1, 0, 1, 1, 8, -8, 16, 16

Applies to

Math

port [ie]

Sets or retrieves the port number associated with a URL.

Syntax

HTML - N/A

Scripting

object.port [= *sPort*]

Applies To
a, area, location

posBottom [ie]

Sets or retrieves the bottom position of the object in the units specified by the bottom attribute.

Syntax

HTML

N/A

Scripting

object.style.posBottom [= *iBottom*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

posHeight [ie]

Sets or retrieves the height of the object in the units specified by the height attribute.

Syntax

HTML

N/A

Scripting

object.style.posHeight [= *iHeight*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

position

Sets or retrieves the type of positioning used for the object.

Syntax

CSS

{ position : *sPosition* }

Scripting

object.style.position [= *sPosition*]

sPosition values

static	Default. Object has no special positioning; it follows the layout rules of HTML
absolute	Object is positioned relative to parent element's position or to the body object if its parent element is not positioned using the top and left properties
relative	Object is positioned according to the normal flow, and then offset by the top and left properties
fixed	Object has a fixed position. May not work at some browsers

Applies To

frame, frameset, layoutrect, table, td, th, tr

posLeft [ie]

Sets or retrieves the left position of the object in the units specified by the left attribute.

Syntax

HTML

N/A

Scripting

object.style.posLeft [= *iLeft*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults,

del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

posRight [ie]

Sets or retrieves the right position of the object in the units specified by the right attribute.

Syntax

HTML

N/A

Scripting

object.style.posRight [= *iRight*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

posTop [ie]

Sets or retrieves the top position of the object in the units specified by the top attribute.

Syntax

HTML

N/A

Scripting

object.style.posTop [= *iTop*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

posWidth [ie]

Sets or retrieves the width of the object in the units specified by the width attribute.

Syntax

HTML - N/A

Scripting

object.style.posWidth [= *iWidth*]

Applies To

a, abbr, acronym, address, applet, b, base, basefont, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

<pre>

The pre element defines preformatted text. The text enclosed in the pre element usually preserves spaces and line breaks. The text renders in a fixed-pitch font.

Ex

<pre>

This text is in a fixed-pitch font,
and it preserves
both spaces and
line breaks</pre>

Optional Attributes

width (the maximum number of characters per line (usually 40, 80, or 132)

Standard Attributes

id, class, title, style, dir, lang, xml:lang, xml:space

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :first-letter, :first-line, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

previousPage() [ie]

Displays the previous page of records in the data set to which the table is bound.

Syntax

table.previousPage()

Applies to

table

previousSibling

Retrieves a reference to the previous child of the parent for the object.

Syntax

HTML

N/A

Scripting

[*oElement* =] *object*.previousSibling

Applies To

a, abbr, acronym, address, applet, area, attribute, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextNode, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

print() [ie]

Prints the document associated with the window.

When the user clicks the OK button, the following sequence of events occurs:

- 1. The onbeforeprint event fires.
- 2. The document prints.
- 3. The onafterprint event fires.

Syntax

window.print()

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

window

profile

Sets or retrieves one or more URL(s) in which the object's properties and legal values for those properties are defined.

<head PROFILE = *sProfile*... >

Scripting

head.profile(*v*) [= *sProfile*]

Applies To

head

prompt() [ie]

Displays a dialog box that prompts the user with a message and an input field.

Syntax

vTextData = window.prompt([sMessage] [,sDefaultValue])

Parameters

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box. By default, this parameter is set to "".
<i>sDefaultValue</i>	Optional. String that specifies the default value of the input field. By default, this parameter is set to "undefined".

Applies to

window

propertyName [ie]

Sets or retrieves the name of the property that changes on the object.

Syntax

event.propertyName [= *sProperty*]

Applies To

event

protocol [ie]

Sets or retrieves the protocol portion of a URL.

Syntax

HTML

N/A

Scripting

object.protocol [= *sProtocol*]

Applies To

a, area, document, img, location

prototype [JavaScript]

The prototype property allows you to add properties and methods to an object.

Syntax

object.prototype.*name*=*value*

Ex

```
<script type="text/javascript">
function employee(name, jobtitle, born)
{
  this.name=name
  this.jobtitle=jobtitle
  this.born=born
}
```

```
var fred=new employee("Fred Flintstone", "Caveman", 1970)
employee.prototype.salary=null
fred.salary=20000
document.write(fred.salary)
</script>
```

The output of the code above will be:
20000

Applies To
Array, Boolean, Date, Math, String

pseudoClass [ie]

Retrieves a string that identifies the pseudo class of the page or pages an @page rule applies to.

Syntax

[*sPseudoClass* =] *page.pseudoClass*

sPseudoClass values

:first Rule	applies to the first page.
:left Rule	applies to pages on the left side of the bindingverso pages.
:right Rule	applies to pages on the right side of the bindingrecto pages.

Applies To

frame, frameset, layoutrect, table, td, th, tr

push() [JavaScript]

The push() method adds one or more elements to the end of an array and returns the new length.

Syntax

arrayObject.push(newelement1, newelement2,, newelementX)

Parameters

<i>newelement1</i>	Required. The first element to add to the array
<i>newelement2</i>	Optional. The second element to add to the array
<i>newelementX</i>	Optional. Several elements may be added

Ex

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Azul"
arr[1] = "Amarelo"
arr[2] = "Vermelho"
document.write(arr + "<br />")
document.write(arr.push("preto e branco") + "<br />")
document.write(arr)
</script>
```

The output of the code above will be:

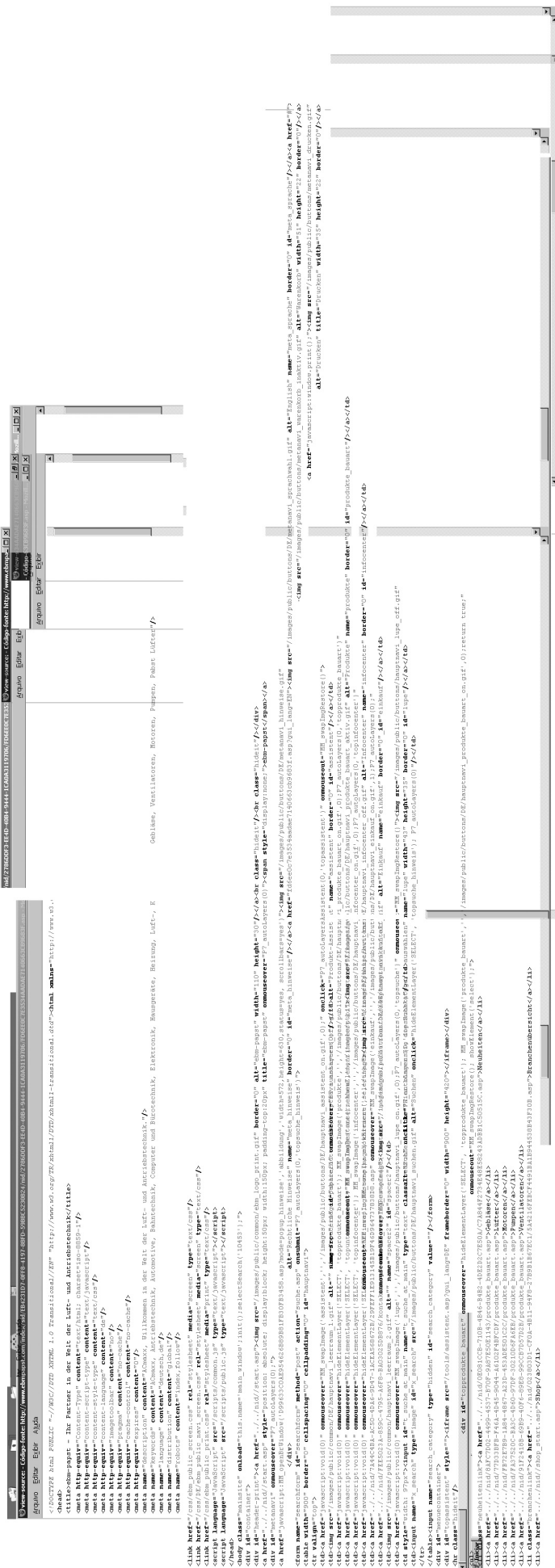
Azul, Amarelo, Vermelho

4

Azul, Amarelo, Vermelho, preto e branco

Applies to

Array



<q>

Sets apart a quotation in text.

Ex

Here comes a short quotation: <q>here is a short quotation here is a short quotation</q>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

qualifier [ie]

Sets or retrieves the name of the data member provided by a data source object.

Syntax

event.qualifier [= *sQualifier*]

Applies To

event

queryCommandEnabled() [ie]

Returns a Boolean value that indicates whether a specified command can be successfully executed using execCommand, given the current state of the document.

Syntax

bEnabled = *object*.queryCommandEnabled(*sCmdID*)

Parameter

<i>sCmdID</i>	Required. String that specifies a command identifier.
---------------	---

Applies to

document, controlRange, TextRange

queryCommandIndeterm() [ie]

Returns a Boolean value that indicates whether the specified command is in the indeterminate state.

Syntax

bIndeterminate = *object*.queryCommandIndeterm(*sCmdID*)

Parameter

<i>sCmdID</i>	Required. String that specifies a command identifier.
---------------	---

Applies to

document, controlRange, TextRange

queryCommandState() [ie]

Returns a Boolean value that indicates the current state of the command.

Syntax

bDone = *object*.queryCommandState(*sCmdID*)

Parameter

<i>sCmdID</i>	Required. String that specifies a command identifier.
---------------	---

Applies to

document, controlRange, TextRange

queryCommandSupported() [ie]

Returns a Boolean value that indicates whether the current command is supported on the current range.

Syntax

bSupported = *object*.queryCommandSupported(*sCmdID*)

Parameter

<i>sCmdID</i>	Required. String that specifies a command identifier.
---------------	---

Applies to

document, controlRange, TextRange

queryCommandValue() [ie]

Returns the current value of the document, range, or current selection for the given command.

Syntax

vCmdValue = *object*.queryCommandValue(*sCmdID*)

Parameter

<i>sCmdID</i>	Required. String that specifies a command identifier.
---------------	---

Applies to

document, controlRange, TextRange

random() [JavaScript]

The random() method returns a random number between 0 and 1.

Syntax

Math.random()

Ex

In this example we will get a random number between 0 and 1:

```
<script type="text/javascript">
document.write(Math.random())
</script>
```

The output of the code above can be:

0.19458011666887554

Applies to

Math

readOnly

(input, textArea)

Retrieves whether the rule or style sheet is defined on the page or is imported.

Syntax

HTML

<ELEMENT READONLY ... >

Scripting

object.readOnly [= bRead]

bReadOnly values

false	Default. Property is not set on the object
true	Object is set to read-only

Applies To

input type=text, input type=password, textarea

readOnly

(styleSheet, rule)

Sets or retrieves the value indicated whether the content of the object is read-only.

Syntax

[bReadOnly =] object.readOnly

bReadOnly values

false	Default. Rule or style sheet is obtained through a link object or the @import rule
true	Rule or style sheet is defined on the page

Applies To

styleSheet, rule

readyState [ie]

(document, a, abbr, acronym ...)

Retrieves the current state of the object.

Syntax

HTML

N/A

Scripting

[vState =] object.readyState

vState values

uninitialized	Object is not initialized with data.
loading	Object is loading its data.
loaded	Object has finished loading its data.
interactive	User can interact with the object even though it is not fully loaded.
complete	Object is completely initialized.

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, head, hn, hr, html, i, iframe, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, namespace, nextID, nobr, noframes, noscript, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, textarea, title, tt, u, ul, var, xml, xmp, document

readyState [ie]

Retrieves a value that indicates the current state of the object.

Syntax

[lState =] object.readyState

lState values

0	Object is not initialized with data.
1	Object is loading its data.
2	Object has finished loading its data.
3	User can interact with the object even though it is not fully loaded.
4	Object is completely initialized.

Applies To

object

readyState [ie]

(XMLHttpRequest)

Retrieves the current state of the request operation.

Syntax

[nState =] XMLHttpRequest.readyState

nState values

0	The object has been created, but not initialized (the open method has not been called).
1	The object has been created, but the send method has not been called.
2	The send method has been called, but the status and headers are not yet available.
3	Some data has been received. Calling the responseBody and responseText properties at this state to obtain partial results will return an error, because status and response headers are not fully available.
4	All the data has been received, and the complete data is available.

Applies To

XMLHttpRequest

reason [ie]

Sets or retrieves the result of the data transfer for a data source object.

Syntax

event.reason [= iReason]

iReason values

0	Object is not initialized with data.
1	Object is loading its data.
2	Object has finished loading its data.

Applies To

event

recalc() [ie]

Recalculates all dynamic properties in the current document.

Syntax

document.recalc([bForceAll])

Parameter

bForceAll	Optional. Boolean that specifies one of the following values: false: Default. Recalculates only those expressions that have changed since the last recalculation; true: Recalculates all expressions in the document.
-----------	--

Applies to

document

recordNumber [ie]

Retrieves the ordinal record from the data set that generated the object.

Syntax

Scripting

[iNumber =] object.recordNumber

Applies To

a, button, div, frame, iframe, img, input type=checkbox, input type=hidden, input type=password, input type=radio, input type=text, label, legend, marquee, select, span, textarea



recordset [ie]

Sets or retrieves from a data source object a reference to the default record set.

Syntax

HTML

N/A

Scripting

object.recordset [= *oRecordset*]

Applies To

event, object, XML

referrer

Retrieves the URL of the location that referred the user to the current page.

Syntax

[*sUrl* =] *document*.referrer

Applies To

document

refresh() [ie]

Refreshes the content of the table. This might be necessary after a call to a method such as removeRule, when the page does not automatically reflow.

Syntax

table.refresh()

Applies to

table

rel

Sets or retrieves the relationship between the object and the destination of the link.

Syntax

HTML

<ELEMENT BORDERCOLOR = *sColor*... >

Scripting

object.borderColor [= *sColor*]

Alternate	Substitute version of the file that contains the link.
Appendix	Page that is an appendix for the set of pages.
Bookmark	Bookmark.
Chapter	Page that is a chapter for a set of pages.
Contents	Table of contents document.
Copyright	Copyright notice for the current page.
Glossary	Glossary for the current page.
Help	Help document.
Index	Index document for the current page.
Next	Next document in a sequence.
Offline	href that contains a path to the CDF file to be used for an offline favorite.
Prev	Previous document in a sequence.
Section	Page that is a section for a set of pages.
Shortcut Icon	href that contains a path to an icon file to be used for the favorite or link.
Start	First document of a set.
Stylesheet	Style sheet.
Subsection	Page that is a subsection for a set of pages.

Applies To

a, link

releaseCapture() [ie]

Removes mouse capture from the object in the current document.

Syntax

object.releaseCapture()

Applies to

a, address, applet, area, b, big, blockquote, body, br, button, caption, center, cite, code, custom, dd, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, hn, hr, i, img, input type=submit, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

reload() [ie]

Reloads the current page.

Syntax

location.reload([*bReloadSource*])

Parameters

<i>bReloadSource</i>	Optional. Boolean that specifies one of the following possible values: false: Default. Reloads the page from the browser cache; true: Reloads the page from the server.
----------------------	---

Applies to

location

remove()

Removes an element from the collection.

Syntax

object.remove(*iIndex*)

Parameter

<i>iIndex</i>	Required. Integer that specifies the zero-based index of the element to remove from the collection.
---------------	---

Applies to

areas, controlRange, options, select

removeAttribute()

Removes the given attribute from the object and returns a boolean value indicating if the attribute was successfully removed.

Syntax

bSuccess = *object*.removeAttribute(*sName* [, *iCaseSensitive*])

Parameters

<i>sName</i>	Required. Stringthat specifies the attribute name.
<i>iCaseSensitive</i>	ptional. Integerthat specifies whether to use a case-sensi-tive search to locate the attribute. By default, this value is set to 1 to indicate that the uppercase and lowercase letters in the specified <i>sName</i> parameter must exactly match those in the attribute name. If there are multiple attributes specified with different case sensitivity, the attribute returned might vary across platforms.

Applies to

a, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xmp

removeAttributeNode()

Removes an attribute object from the object and returns a boolean value indicating if the attribute was successfully removed.

Syntax

oRetAttribute = *object*.removeAttributeNode(*oSrcAttribute*)

Parameter

<i>oSrcAttribute</i>	Required. attribute object to be removed.
----------------------	---

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input

type=, text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

removeBehavior() [ie]

Detaches a behavior from the element and returns a boolean value indicating if the behavior was successfully removed.

Syntax

bSuccess = *object.removeBehavior(iID)*

Parameter

<i>iID</i>	Required. Integer that specifies the identifier returned from a previous addBehavior call.
------------	--

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, STYLE, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

removeChild()

Removes a child node from the object and returns a reference to the object that is removed.

Syntax

oRemove = *object.removeChild(oNode)*

Parameter

<i>oNode</i>	Required. Object that specifies the element to be removed from the document.
--------------	--

Applies to

a, abbr, acronym, address, attribute, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

removeExpression() [ie]

Removes the expression from the specified property.

Syntax

bSuccess = *object.removeExpression(sPropertyName)*

Parameter

<i>sPropertyName</i>	Required. String that specifies the name of the property from which to remove an expression.
----------------------	--

Applies to

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, VAR

removeNamedItem()

Removes an attribute specified with the name property from an element using the attributes collection. Returns an attribute removed from the document if successful, otherwise null.

Syntax

newretNode = *attributes.removeNamedItem(sName)*

Parameter

<i>sName</i>	Required. String that specifies the name of an attribute to remove.
--------------	---

Applies to

attributes

removeNode() [ie]

Removes the object from the document hierarchy and returns a reference to the object that is removed.

Syntax

oRemoved = *object.removeNode([bRemoveChildren])*

Parameter

<i>bRemoveChildren</i>	Optional. Boolean that specifies one of the following values: false: Default. childNodes collection of the object is not removed; true: childNodes collection of the object is removed.
------------------------	--

Applies to

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, STYLE, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

removeRule() [ie]

Deletes an existing style rule for the styleSheet object, and adjusts the index of the rules collection accordingly.

Syntax

styleSheet.removeRule([iIndex])

Parameter

<i>iIndex</i>	Optional. Integer that specifies the index value of the rule to be deleted from the style sheet. If an index is not provided, the first rule in the rules collection is removed.
---------------	--

Applies to

styleSheet

repeat [ie]

Retrieves whether the onkeydown event is being repeated.

Syntax

event.repeat [= *bRepeat*]

bRepeat values

false	Event fires once
true	Event fires two or more times

Applies To

event

replace() [ie]

Replaces the current document by loading another document at the specified URL.

Syntax

location.replace(sURL)

Parameter

<i>sURL</i>	Required. String that specifies the URL to insert into the session history.
-------------	---

Applies to

location

replace() [JavaScript]

The replace() method is used to replace some characters with some other characters in a string.

Syntax

stringObject.replace(*findstring*, *newstring*)

Parameters

<i>findstring</i>	Required. Specifies a string value to find. To perform a global search add a 'g' flag to this parameter and to perform a case-insensitive search add an 'i' flag
<i>newstring</i>	Required. Specifies the string to replace the found value from findstring

Ex

```
<script type="text/javascript">
var str="Visit Microsoft!"
document.write(str.replace(/Microsoft/, "Me"))
```

</script>
The output of the code above will be:
Visit Me!

In the following example we will perform a case-insensitive search, and the word Microsoft will be replaced:

```
<script type="text/javascript">
var str="Visit Microsoft!"
document.write(str.replace(/microsoft/i, "W3Schools"))
</script>
```

The output of the code above will be:
Visit W3Schools!

Applies to
String

replaceAdjacentText() [ie]

Replaces the text adjacent to the element and returns a Pointer to a String returning the old text.

Syntax

alert([*sMessage*])

Parameters

<i>sWhere</i>	Required. String that specifies where the text is located using one of the following values:
beforeBegin	Text is returned immediately before the element.
afterBegin	Text is returned after the start of the element but before all other content in the element.
beforeEnd	Text is returned immediately before the end of the element but after all other content in the element.
afterEnd	Text is returned immediately after the end of the element.
sReplaceText	Required. Stringthat specifies the replacement text.

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

replaceChild()

Replaces an existing child element with a new child element. and returns a reference to the object that is replaced.

Syntax

oReplace = *object*.replaceChild(*oNewNode*, *oChildNode*)

Parameters

<i>oNewNode</i>	Required. Object that specifies the new element to be inserted into the document.
<i>oChildNode</i>	Required. Object that specifies the existing element to be replaced.

Applies to

a, abbr, acronym, address, attribute, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

replaceData()

Replaces a specified range of characters in the object with a new character string.

Syntax

object.replaceData(*nOffset*, *nCount*, *sString*)

Parameters

<i>nOffset</i>	Optional. String that specifies the message to display in the dialog box.
nCount	Required. Integer that specifies the number of characters to replace.
sString	Required. String that specifies the new character string.

Applies to

comment, TextNode

replaceNode() [ie]

Replaces the object with another element.

Syntax

oReplace = *object*.replaceNode(*oNewNode*)

Parameter

<i>oNewNode</i>	Required. Object that specifies the new element to replace the object.
-----------------	--

Applies to

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frameset, head, hn, html, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

reset()

Simulates a mouse click on a reset button for the calling form.

Syntax

FORM.reset()

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to

FORM

resizeBy() [ie]

Changes the current size of the window by the specified x- and y-offset.

Syntax

window.resizeBy(*iX*, *iY*)

Parameters

<i>iX</i>	Required. Integer that specifies the horizontal offset in pixels. The value can be either positive or negative.
<i>iY</i>	Required. Integer that specifies the vertical offset in pixels. The value can be either positive or negative.

Applies to

window

resizeTo() [ie]

Sets the size of the window to the specified width and height values.

Syntax

window.resizeTo(*iWidth*, *iHeight*)

Parameter

<i>iWidth</i>	Required. Integer that specifies the width of the window in pixels. The value can be either positive or negative.
<i>iHeight</i>	Required. Integer that specifies the height of the window in pixels. The value can be either positive or negative.

Applies to

window

responseBody

Retrieves the response body as an array of unsigned bytes.

Preliminary.

Syntax

[*vBody* =] *XMLHttpRequest*.responseBody

Applies To

XMLHttpRequest

responseText

Retrieves the response body as a string. Preliminary.

Syntax

[*sBody* =] *XMLHttpRequest.responseText*

Applies To

XMLHttpRequest

responseXML

Retrieves the response body as an Extensible Markup Language (XML) Document Object Model (DOM) object.

Syntax

[*oBody* =] *XMLHttpRequest.responseXML*

Applies To

responseBody, responseText

return

The return statement specifies the value to be returned by a function.

Syntax

return [expression];

Ex

The following function returns the square of its argument, x, where x is a number.

```
function square(x) {  
    return x * x; }
```

returnValue [ie]

(window)

Sets or retrieves the value returned from the modal dialog window.

Syntax

window.returnValue [= *vValue*]

Applies To

window

returnValue [ie]

(event)

Sets or retrieves the return value from the event.

Syntax

event.returnValue [= *bValue*]

bValue values

false	Default action of the event on the source object is canceled
true	Default. Value from the event is returned

Applies To

event

rev

Sets or retrieves the relationship between the object and the destination of the link.

Syntax

HTML

<*ELEMENT* REV = *sRelation*... >

Scripting

object.rev [= *sRelation*]

sRelation values

Alternate	Substitute version of the file that contains the link.
Appendix	Page that is an appendix for the set of pages.
Bookmark	Bookmark.
Chapter	Page that is a chapter for a set of pages.
Contents	Table of contents document.
Copyright	Copyright notice for the current page.
Glossary	Glossary for the current page.
Help	Help document.
Index	Index document for the current page.

Next	Next document in a sequence.
Prev	Previous document in a sequence.
Section	Page that is a section for a set of pages.
Start	First document of a set.
Stylesheet	Style sheet.
Subsection	Page that is a subsection for a set of pages.

Applies To

a, link

reverse() [JavaScript]

The reverse() method is used to reverse the order of the elements in an array.

Syntax

arrayObject.reverse()

Ex

In this example we will create an array, and then reverse the order of it:

```
<script type="text/javascript">  
var arr = new Array(3)  
arr[0] = "Azul"  
arr[1] = "Amarelo"  
arr[2] = "Vermelho"  
document.write(arr + "<br />")  
document.write(arr.reverse())  
</script>
```

The output of the code above will be:

Azul, Amarelo, Vermelho
Vermelho, Amarelo, Azul

Applies to

Array

right

(textRectangle)

Sets or retrieves the right coordinate of the rectangle surrounding the object content.

Syntax

TextRectangle.right [= *iCoord*]

Applies To

TextRectangle

right

(css)

Sets or retrieves the position of the object relative to the right edge of the next positioned object in the document hierarchy.

Syntax

{ right : *sPosition* }

Scripting

object.style.right [= *sPosition*]

sPosition values

auto	Default. Position is determined by the regular HTML layout of the page
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	Integer, followed by a percent sign (%). The value is a percentage of the width of the parent object

Applies To

a, address, applet, b, big, blockquote, button, center, cite, code, currentStyle, dd, dfn, dir, div, dl, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, layoutrect, legend, li, listing, marquee, menu, object, ol, p, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

rightMargin [ie]

Sets or retrieves the right margin for the entire body of the page.

Syntax

HTML

<body RIGHTMARGIN = *sMargin*... >

Scripting

body.rightMargin [= *sMargin*]

Applies To
body

round() [JavaScript]

The round() method rounds a number to the nearest integer.

Syntax

Math.round(x)

Parameter

x	Required. A number
---	--------------------

Ex

In this example we will round different numbers with the round() method:

```
<script type="text/javascript">
document.write(Math.round(0.60) + ", ")
document.write(Math.round(0.50) + ", ")
document.write(Math.round(0.49) + ", ")
document.write(Math.round(-4.40) + ", ")
document.write(Math.round(-4.60))
</script>
```

The output of the code above will be:

1, 1, 0, -4, -5

Applies to
Math

rowIndex

Retrieves the position of the object in the rows collection for the table.

Syntax

HTML

N/A

Scripting

[*iIndex* =] *tr*.rowIndex

Applies To
tr

ROWS

(frameset)

Sets or retrieves the frame heights of the object.

Syntax

HTML

<FRAMESET ROWS = *sHeight*... >

Scripting

FRAMESET.rows [= *sHeight*]

sHeight values

<i>height</i>	Integer that specifies the frame height, in pixels
%	Integer, followed by a %. The value is a percentage of total available height
<i>height</i> *	Integer that specifies the frame height as a relative value. After allocating pixel or percentage values, the remaining space is divided among all relative-sized frames

Applies To
FRAMESET

ROWS

(textarea)

Sets or retrieves the number of horizontal rows contained in the object.

Syntax

HTML

<textarea ROWS = *iRows*... >

Scripting

textarea.rows [= *iRows*]

Applies To
textarea

ROWS

Retrieves a collection of tr (table row) objects from a table object.

Syntax

[*oColl* =] *object*.rows

[*oObject* =] *object*.rows(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|------------------|---|
| <i>oColl</i> | Array of tr objects. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> . |

Applies To
table, tbody, tfoot, thead

rowSpan

Sets or retrieves how many rows in a table the cell should span.

Syntax

HTML

<ELEMENT ROWSPAN = *iRows*... >

Scripting

object.rowSpan [= *iRows*]

Applies To
td, th

<rt> [ie]

Designates the ruby text for the ruby element. This object is part of a proposed addition to html, Not supported by some browsers.

Ex

```
<ruby>
  Base Text
  <rt>Ruby Text
</ruby>
```

Behaviors

clientCaps, download, homePage

Collection

behaviorUrns, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, blur, componentFromPoint, fireEvent, focus, getAttributeNode, getExpression, normalize, removeAttributeNode, removeBehavior, removeExpression, setActive, setAttributeNode, setExpression

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, cursor, direction, display, filter, height, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, styleFloat, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, width, wordWrap, writingMode, zoom

<ruby> [ie]

Designates an annotation or pronunciation guide to be placed above or inline with a string of text. This object is part of a proposed addition to html, Not supported by some browsers.

Ex

```
<ruby>
Base Text
<rt>Ruby Text
</ruby>
```

Behaviors

clientCaps, download, homePage

Collection

behaviorUrns, children, filters

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, blur, componentFromPoint, fireEvent, focus, getAttributeNode, getExpression, normalize, removeAttributeNode, removeBehavior, removeExpression, setActive, setAttributeNode, setExpression

Style Properties and attributes

accelerator, backgroundPositionX, backgroundPositionY, behavior, cursor, direction, display, filter, height, layoutGrid, layoutGridMode, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, styleFloat, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, width, wordWrap, writingMode, zoom

rubyAlign [ie]

Sets or retrieves the position of the ruby text specified by the rt object.

Syntax

CSS

{ ruby-align : *sRubyAlign* }

Scripting

object.style.rubyAlign [= *sRubyAlign*]

sRuby values

| | |
|--------|--|
| auto | Default. Browser determines how the ruby text is aligned. The recommended behavior for an ideographic (Asian character) ruby is to be aligned in the distribute-space mode. The recommended behavior for a Latin character ruby is to be aligned in the center mode. |
| left | Ruby text is left-aligned with the base. |
| center | Ruby text is centered within the width of the base. If the length of the base is smaller than the length of the ruby text, the base is centered within the width of the ruby text. |
| right | Ruby text is right-aligned with the base. |

Applies To

currentStyle, ruby, runtimeStyle, style

rubyOverhang [ie]

Sets or retrieves the position of the ruby text specified by the rt object.

Syntax

CSS

{ ruby-overhang : *sRubyOverhang* }

Scripting

object.style.rubyOverhang [= *sRubyOverhang*]

sRubyOverhang values

| | |
|------------|--|
| auto | Default. Ruby text overhangs any other text adjacent to the base text. |
| whitespace | Ruby text overhangs only white-space characters. |
| none | Ruby text overhangs only text adjacent to its base. |

Applies To

currentStyle, ruby, runtimeStyle, style

rubyPosition [ie]

Sets or retrieves the position of the ruby text specified by the rt object.

Syntax

CSS

{ ruby-position : *sRubyPlacement* }

Scripting

object.style.rubyPosition [= *sRubyPlacement*]

sRubyPlacement values

| | |
|--------|--|
| above | Default. Ruby text is positioned above the base text |
| inline | Ruby text is positioned inline with the base text |

Applies To

currentStyle, ruby, runtimeStyle, style

rule

Represents a style within a Cascading Style Sheets (CSS) that consists of a selector and one or more declarations.

Properties

| | |
|--------------|--|
| readOnly | Retrieves whether the rule or style sheet is defined on the page or is imported. |
| selectorText | Retrieves a string that identifies which elements the corresponding style sheet rule applies to. |

Objects

style

rules

Sets or retrieves which dividing lines (inner borders) are displayed.

Syntax

HTML

<table RULES = *sRule*... >

Scripting

table.rules [= *sRule*]

sRule values

| | |
|--------|--|
| all | Borders are displayed on all rows and columns. |
| cols | Borders are displayed between all table columns. |
| groups | Horizontal borders are displayed between all tHead, tBody, and tFoot objects; vertical borders are displayed between all colGroup objects. |
| none | All interior table borders are removed. |
| rows | Horizontal borders are displayed between all table rows. |

Applies To

table

rules

Retrieves a collection of rules defined in a style sheet.

Syntax

[*oColl* =] styleSheet.rules

[*oObject* =] styleSheet.rules(*iIndex*)

Possible Values

| | |
|----------------|---|
| <i>oColl</i> | Array of rules. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer that specifies the zero-based index of the item to be returned. |

Applies To

styleSheet

RunOnceShown()

Indicates that the “first run” page has been shown.

Syntax

window.external.RunOnceShown()

Applies to

external

runtimeStyle

Represents the cascaded format and style of the object that overrides the format and style specified in global style sheets, inline styles, and HTML attributes.

Ex

To change or clear multiple style properties at once, use this object with the cssText property. For example, to change the font color and background color of a DIV element, you could use the following code:

<DIV onclick="this.runtimeStyle.cssText = 'color:red;background-color:blue;border:5px solid black;'">
Click on this DIV to change style properties.</DIV>

Property

onOffBehavior

Methods

getAttribute, getExpression, removeAttribute, removeExpression, setAttribute,

setExpression

Style Properties and attributes

accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderCollapse, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cssText, cursor, direction, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, height, imeMode, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyle, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, msInterpolationMode, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, rubyAlign, rubyOverhang, rubyPosition, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, tableLayout, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

Applies To

a, abbr, acronym, address, applet, b, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, cssText, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

sobre o segredo comercial

O capitalismo liberal, baseado sobre a livre concorrência e a liberdade de comércio, já desapareceu.

O capitalismo monopolista, que o substituiu, não somente foi incapaz de controlar a anarquia do mercado, como também, ao contrário, conferiu a esta última um caráter particularmente convulsivo. A necessidade de um controle sobre a economia, de uma «direção» estatal, de uma planificação» é, atualmente, reconhecida, pelo menos em palavras, por quase todas as correntes do pensamento burguês e pequeno-burguês, do fascismo à social-democracia. Para os fascistas, trata-se, sobretudo, de uma pilhagem planificada do povo com fins militares. Os sociais-democratas procuram esvaziar o oceano da anarquia com a colher de uma «planificação» burocrática. Os engenheiros e os professores escrevem artigos sobre a tecnocracia». Os governos democráticos chocam-se, nas suas mesquinhas tentativas de regulamentação», à sabotagem intransponível do grande capital.

A verdadeira relação entre exploradores e controladores democráticos é caracterizada do melhor modo pelo fato de que os senhores «reformadores», tomados de santa emoção, param ao limiar dos trustes com seus segredos industriais e comerciais. Nesse terreno reina o princípio da «não-intervenção». As contas entre o capitalista isolado e a sociedade constituem um segredo do capitalista: a sociedade nada tem que ver com isto. O segredo comercial é sempre justificado, como na época do capitalismo liberal pelas exigências da concorrência». Os trustes, porém, não guardam segredos entre si. O segredo comercial, na época atual, é um complot constante do capital monopolista

contra a sociedade. Os projetos de limitação do absolutismo dos padrões pelo direito divino» permanecerão lamentáveis farsas, enquanto os proprietários privados dos meios sociais de produção puderem esconder aos produtores e aos consumidores as maquinações da exploração, da pilhagem, do engano.

A abolição do segredo comercial» é o primeiro passo em direção a um verdadeiro controle da indústria.

— TROTSKY, Leon, o programa de transição

<S>

Renders text in strike-through type.

Ex

A new version is <s>not yet available.</s> now available!

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

<samp>

Specifies a code sample.

Ex

<samp>Sample computer code text</samp>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor,

borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

saveType [ie]

Retrieves the clipboard type when oncontentsave fires.

Syntax

[sSaveType =] event.saveType

sSaveType values

| | |
|------|---|
| TEXT | Data saved to the clipboard is in text format |
| HTML | Data saved to the clipboard is in HTML format |

Applies To

event

scheme

Sets or retrieves a scheme to be used in interpreting the value of a property specified for the object.

Syntax

HTML

<meta SCHEME = sScheme... >

Scripting

meta.scheme(v) [= sScheme]

Applies To

meta

scope

Sets or retrieves the group of cells in a table to which the object's information applies.

Syntax

HTML

<ELEMENT SCOPE = sScope... >

Scripting

object.scope(v) [= sScope]

Applies To

td, th

scopeName [ie]

Retrieves the namespace defined for the element.

Syntax

HTML

N/A

Scripting

[sName =] object.scopeName

sName values

| | |
|-------|--|
| HTML | Default. All standard HTML tags |
| scope | Namespace prefix used with the custom tags. This namespace is declared in the document using the XMLNS attribute of the html element |

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp



screen [ie]

Contains information about the client's screen and rendering capabilities.

Properties

| | |
|----------------------|---|
| availHeight | Retrieves the height of the working area of the system's screen, excluding the Microsoft Windows taskbar. |
| availWidth | Retrieves the width of the working area of the system's screen, excluding the Windows taskbar. |
| bufferDepth | Sets or retrieves the number of bits per pixel used for colors in the off-screen bitmap buffer. |
| colorDepth | Retrieves the number of bits per pixel used for colors on the destination device or buffer. |
| deviceXDPI | Retrieves the actual number of horizontal dots per inch (DPI) of the system's screen. |
| deviceYDPI | Retrieves the actual number of vertical dots per inch (DPI) of the system's screen. |
| fontSmoothingEnabled | Retrieves whether the user has enabled font smoothing in the Display control panel. |
| height | Retrieves the vertical resolution of the screen. |
| logicalXDPI | Retrieves the normal number of horizontal dots per inch (DPI) of the system's screen. |
| logicalYDPI | Retrieves the normal number of vertical dots per inch (DPI) of the system's screen. |
| updateInterval | Sets or retrieves the update interval for the screen. |
| width | Retrieves the horizontal resolution of the screen. |

Applies To

window

screenLeft [ie]

Retrieves the x-coordinate of the upper left-hand corner of the browser's client area, relative to the upper left-hand corner of the screen.

Syntax

[*iPos* =] *window.screenLeft*

Applies To

window

screenTop [ie]

Retrieves the y-coordinate of the top corner of the browser's client area, relative to the top corner of the screen.

Syntax

[*iPos* =] *window.screenTop*

Applies To

window

screenX [ie]

Retrieves the x-coordinate of the mouse pointer's position relative to the user's screen.

Syntax

event.screenX [= *iSize*]

Applies To

event

screenY [ie]

Sets or retrieves the y-coordinate of the mouse pointer's position relative to the user's screen.

Syntax

event.screenY [= *iSize*]

Applies To

event

<script>

Specifies a script for the page that is interpreted by a script engine.

Ex

```
<script type="text/javascript">
document.write("Hello World!")
</script>
```

Required Attributes

type (text/ecmascript, text/javascript, text/jscript, text/vbscript, text/vbs, text/xml)

Optional Attributes

charset, defer, language, src

Standard Attributes

xml:space

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onload, onpropertychange, onreadystatechange

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, layoutGrid, layoutGridMode, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition

scripts [ie]

Retrieves a collection of all script objects in the document.

Syntax

[*oColl* =] document.scripts

[*oObject* =] document.scripts(*vIndex* [, *iSubIndex*])

Possible Values

| | |
|------------------|---|
| <i>oColl</i> | Array of script objects. |
| <i>oObject</i> | Reference to an individual item in the array of elements contained by the object. |
| <i>iIndex</i> | Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements. |
| <i>iSubIndex</i> | Optional. Position of an element to retrieve. This parameter is used when <i>vindex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> . |

Applies To

document

scroll [ie]

Sets or retrieves a value that indicates whether the scroll bars are turned on or off.

Syntax

<ELEMENT SCROLL = *sScroll*... >

Scripting

object.scroll [= *sScroll*]

sScroll values

| | |
|------|---|
| yes | IE Default. Scroll bars are turned on |
| no | Scroll bars are turned off |
| auto | Scroll bars are shown when the page content exceeds the client area |

Applies To

body, HTML

scroll() [ie]

Causes the window to scroll to the specified x- and y-offset at the upper-left corner of the window.

Syntax

*window.scroll(*iX*, *iY*)*

Parameters

| | |
|-----------|---|
| <i>iX</i> | Required. Integer that specifies the horizontal scroll offset, in pixels. |
| <i>iY</i> | Required. Integer that specifies the vertical scroll offset, in pixels. |

Applies to

window

scrollAmount [ie]

Sets or retrieves the number of pixels the text scrolls between each subsequent drawing of the marquee.

Syntax

HTML

<marquee SCROLLAMOUNT = *iAmount*... >

Scripting

marquee.scrollAmount [= *iAmount*]

Applies To

marquee

scrollbar3dLightColor [ie]

Sets or retrieves the color of the top and left edges of the scroll box and scroll arrows of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-3dlight-color : *vColor* }

Scripting

object.style.scrollbar3dLightColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarArrowColor [ie]

Sets or retrieves the color of the arrow elements of a scroll arrow. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-arrow-color : *vColor* }

Scripting

object.style.scrollbarArrowColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarBaseColor [ie]

Sets or retrieves the color of the main elements of a scroll bar, which include the scroll box, track, and scroll arrows. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-base-color : *vColor* }

Scripting

object.style.scrollbarBaseColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarDarkShadowColor [ie]

Sets or retrieves the color of the gutter of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-darkshadow-color : *vColor* }

Scripting

object.style.scrollbarDarkShadowColor [= *vColor*]

Applies To

frame, frameset, layoutrect, table, td, th, tr

scrollbarFaceColor [ie]

Sets or retrieves the color of the scroll box and scroll arrows of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-face-color : *vColor* }

Scripting

object.style.scrollbarFaceColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarHighlightColor [ie]

Sets or retrieves the color of the top and left edges of the scroll box and scroll arrows of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-highlight-color : *vColor* }

Scripting

object.style.scrollbarHighlightColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarShadowColor [ie]

Sets or retrieves the color of the bottom and right edges of the scroll box and scroll arrows of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-shadow-color : *vColor* }

Scripting

object.style.scrollbarShadowColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollbarTrackColor [ie]

Sets or retrieves the color of the track element of a scroll bar. This property is a Microsoft extension to CSS.

Syntax

CSS

{ scrollbar-track-color : *vColor* }

Scripting

object.style.scrollbarTrackColor [= *vColor*]

Applies To

applet, bdo, body, currentStyle, custom, defaults, div, embed, object, runtimeStyle, style, textarea

scrollBy() [ie]

Causes the window to scroll relative to the current scrolled position by the specified x- and y-pixel offset.

Syntax

*window.scrollBy(*iX*, *iY*)*

Parameters

| | |
|-----------|--|
| <i>iX</i> | Required. Integer that specifies the horizontal scroll offset, in pixels. Positive values scroll the window right, and negative values scroll it left. |
| <i>iY</i> | Required. Integer that specifies the vertical scroll offset, in pixels. Positive values scroll the window down, and negative values scroll it up. |

Applies to

window

scrollDelay [ie]

Sets or retrieves the speed of the marquee scroll, in milliseconds.

Syntax

HTML

<marquee SCROLLDELAY = *iDelay*... >

Scripting

marquee.scrollDelay [= *iDelay*]

Applies To

marquee

scrollHeight [ie]

Retrieves the scrolling height of the object, in px.

Syntax

HTML

N/A

Scripting

[*iHeight* =] *object*.scrollHeight

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, head, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

scrollIntoView() [ie]

Causes the object to scroll into view, aligning it either at the top or bottom of the window.

Syntax

object.scrollIntoView([*bAlignToTop*])

Parameter

| | |
|--------------------|---|
| <i>bAlignToTop</i> | Optional. Boolean that specifies one of the following values: true: Default. Scrolls the object so that top of the object is visible at the top of the window. false: Scrolls the object so that the bottom of the object is visible at the bottom of the window. |
|--------------------|---|

Applies to

a, address, applet, area, b, big, blockquote, br, button, caption, center, cite, code, col, colgroup, comment, controlRange, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, TextRange, tfoot, th, thead, tr, tt, u, ul, var, wbr, xmp

scrolling

Sets or retrieves whether the frame can be scrolled.

Syntax

HTML

<ELEMENT SCROLLING = *sScrolling*... >

Scripting

object.scrolling [= *sScrolling*]

sScrolling values

| | |
|------|---|
| auto | Default. Browser determines whether scroll bars are necessary |
| no | Frame cannot be scrolled |
| yes | Frame can be scrolled |

Applies To

frame, iframe

scrollLeft [ie]

Sets or retrieves the distance between the left edge of the object and the leftmost portion of the content currently visible in the window.

Syntax

HTML

N/A

Scripting

object.scrollLeft [= *iDistance*]

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, head, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

scrollTo() [ie]

Scrolls the window to the specified x- and y-offset.

Syntax

window.scrollTo(*iX*, *iY*)

Parameters

| | |
|-----------|---|
| <i>iX</i> | Required. Integer that specifies the horizontal scroll offset, in pixels. |
| <i>iY</i> | Required. Integer that specifies the vertical scroll offset, in pixels. |

Applies to

window

scrollTop [ie]

Sets or retrieves the distance between the top of the object and the topmost portion of the content currently visible in the window.

Syntax

HTML

N/A

Scripting

object.scrollTop [= *iDistance*]

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, head, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

scrollWidth [ie]

Retrieves the scrolling width of the object.

Syntax

HTML

N/A

Scripting

[*iWidth* =] *object*.scrollWidth

Applies To

a, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, form, head, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, menu, meta, nobr, object, ol, option, p, plaintext, pre, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

search [ie]

Sets or retrieves the substring of the href property that follows the question mark.

Syntax

HTML

N/A

Scripting

object.search [= *sSearch*]

Applies To

a, area, location

search() [JavaScript]

The search() method is used to search a string for a specified value.

Syntax

stringObject.search(*searchstring*)

Parameter

| | |
|---------------------|---|
| <i>searchstring</i> | Required. The value to search for in a string. To perform a case-insensitive search add an 'i' flag |
|---------------------|---|

Ex

In the following example we will perform a case-insensitive search:

```
<script type="text/javascript">
var str="Visit Me!"
document.write(str.search(/me/i))
</script>
```

The output of the code above will be:

6

Applies to

String

sectionRowIndex

Retrieves the position of the object in the tBody, tHead, tFoot, or rows collection.

Syntax

HTML

N/A

Scripting

[*iIndex* =] *tr*.sectionRowIndex

Applies To

tr

SECURITY [ie]

Sets the value indicating whether the source file of a frame or iframe has specific security restrictions applied.

Syntax

HTML

<ELEMENT SECURITY=sSecure ... >

Scripting

N/A

sSecure values

| | |
|------------|---|
| restricted | Applies security settings to frame or iframe source files |
|------------|---|

Applies To

frame, iframe

<select>

Denotes a list box or drop-down list.

Ex

```
<select>
  <option value ="deeon">Dj Deeon</option>
  <option value ="assault">Dj Assault</option>
  <option value ="slugo">Dj Slugo</option>
  <option value ="eu">Dj Eu</option>
</select>
```

Optional Attributes

disabled, multiple, name, size

Standard Attributes

id, class, title, style, dir, lang, xml:lang, accesskey, tabindex

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onchange

Methods

add, addBehavior, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, remove, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView,

setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode, urns

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundColor, behavior, bottom, clear, clip, color, direction, display, font, fontFamily, fontSize, fontStyle, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, width, wordSpacing, wordWrap, zoom

select()

(controlRange, textRange)

Highlights the input area of a form element.

Syntax

object.select()

Applies to

TextRange, controlRange

select()

(input, textarea)

Makes the selection equal to the current object.

Syntax

object.select()

Applies to

input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, textarea

selected

Sets or retrieves whether the option in the list box is the default item.

Syntax

HTML

<option SELECTED ... >

Scripting

option.selected [= *bSelected*]

bSelected values

false	Default. Item is not selected as the default
true	Item is selected as the default

Applies To

option

selectedIndex

Sets or retrieves the index of the selected option in a select object.

Syntax

HTML

N/A

Scripting

select.selectedIndex [= *iIndex*]

Applies To

select

selection [ie]

Represents the active selection, which is a highlighted block of text, and/or other elements in the document on which a user or a script can carry out some action.

Parameters

type	Retrieves the type of selection.
typeDetail	Retrieves the name of the selection type.

Behaviors

time2

Collection

TextRange

Events

ontimeerror

Methods

clear, createRange, createRangeCollection, empty

selector [ie]

Retrieves a string that identifies which page or pages an @page rule applies to.

Syntax

[*sSelector* =] *page.selector*

Applies To

page

selectorText [ie]

Retrieves a string that identifies which elements the corresponding style sheet rule applies to.

Syntax

rule.selectorText [= *sSelectorText*]

Applies To

rule

self [ie]

Retrieves a reference to the current window or frame.

Syntax

HTML

N/A

Scripting

[*oSelf* =] *object.self*

Applies To

frame, window

send()

Sends an HTTP request to the server and receives a response.

Syntax

XMLHttpRequest.send([*varBody*])

Parameter

<i>varBody</i>	Optional. Variant that specifies the body of the message being sent with the request.
----------------	---

Applies to

XMLHttpRequest

setActive() [ie]

Sets the object as active without setting focus to the object.

Syntax

object.setActive()

Applies to

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, window, xmp

setAttribute()

Sets the value of the specified attribute.

Syntax

object.setAttribute(*sName*, *vValue* [, *iFlags*])

Parameters

<i>sName</i>	Required. String that specifies the name of the attribute.
<i>vValue</i>	Required. Variant that specifies the string, number, or Boolean to assign to the attribute.
<i>iFlags</i>	Optional. Integer that specifies one of the following flags: 0: When the attribute is set, it overwrites any attributes with the same name, regardless of their case. 1: Default. The case of the attribute that you set is respected when it is set on the object.

Applies to

a, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom,

dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, userProfile, var, wbr, xmp

setAttributeNode()

Sets an attribute objectnode as part of the object.

Syntax

alert([*sMessage*])

Parameter

<i>oSrcAttribute</i>	Required. attribute object to be assigned.
----------------------	--

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

setCapture() [ie]

Sets the mouse capture to the object belonging to the current document.

Syntax

object.setCapture([*bContainerCapture*])

Parameter

<i>bContainerCapture</i>	Optional. Boolean that specifies one of the following values: true: Default. Events originating within a container are captured by the container. false: Events originating within a container are not captured by the container.
--------------------------	--

Applies to

a, address, applet, area, b, big, blockquote, body, br, button, caption, center, cite, code, custom, dd, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, kbd, label, legend, li, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

setData() [ie]

Assigns data in a specified format to the dataTransfer or clipboardData object. And returns a boolean value indicating if the data was successfully added.

Syntax

bSuccess = *object.setData*(*sDataFormat*, *sData*)

Parameters

<i>sDataFormat</i>	Required. String that specifies the format of the data to be transferred, using one of the following values: Text: Transfers data formatted as text. URL: Transfers data formatted as a URL.
--------------------	--

Applies to

clipboardData, dataTransfer

setDate() [JavaScript]

The setDate() method is used to set the day of the month.

Syntax

dateObject.setDate(*day*)

Parameter

<i>day</i>	Required. A number from 1 to 31 that represents a day in a month
------------	--

Ex

```
<script type="text/javascript">
var d = new Date()
d.setDate(15)
document.write(d)
</script>
```

Applies to
Date

setEndPoint() [ie]

Sets the endpoint of one range based on the endpoint of another range.

Syntax

TextRange.setEndPoint(sType, oTextRange)

Parameter

<i>sMessage</i>	Optional. String that specifies the message to display in the dialog box.
-----------------	---

Applies to
window

setExpression() [ie]

Sets an expression for the specified object.

Syntax

alert([sMessage])

Parameters

<i>sType</i>	Required. String that specifies the endpoint to transfer using one of the following values.
StartToEnd	Move the start of the TextRange object to the end of the specified oTextRange parameter.
StartToStart	Move the start of the TextRange object to the start of the specified oTextRange parameter.
EndToStart	Move the end of the TextRange object to the start of the specified oTextRange parameter.
EndToEnd	Move the end of the TextRange object to the end of the specified oTextRange parameter.
<i>oTextRange</i>	Required. TextRange object from which the source endpoint is to be taken.

Applies to
TextRange

setHours() [JavaScript]

The setHours() method is used to set the hour of a specified time.

Syntax

dateObject.setHours(hour, min, sec, millisec)

Parameter

<i>hour</i>	Required. A number between 0 and 23 representing the hour
<i>min</i>	Optional. A number between 0 and 59 representing the minutes
<i>sec</i>	Optional. A number between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A number between 0 and 999 representing the milliseconds

Applies to
Date

setInterval() [ie]

Evaluates an expression each time a specified number of milliseconds has elapsed.

Syntax

iTimerID = window.setInterval(vCode, iMilliSeconds [,sLanguage])

Parameter

<i>vCode</i>	Required. Variant that specifies a function pointer or string that indicates the code to be executed when the specified interval has elapsed.
<i>iMilliSeconds</i>	Required. Integer that specifies the number of milliseconds.
<i>sLanguage</i>	Optional. String that specifies any one of the possible values for the LANGUAGE attribute.

Applies to
window

setMonth() [JavaScript]

The setMonth() method is used to set the month. The value set by setMonth() is a number between 0 and 11. January is 0, February is 1 and so on.

Syntax

dateObject.setMonth(month, day)

Parameter

<i>day</i>	Optional. A numeric value (from 1 to 31) that represents a day in a month
<i>month</i>	Required. A numeric value between 0 and 11 representing the month

Applies to
Date

setMinutes() [JavaScript]

The setMinutes() method is used to set the minutes of a specified time.

Syntax

dateObject.setMinutes(min, sec, millisec)

Parameter

<i>min</i>	Required. A numeric value between 0 and 59 representing the minutes
<i>sec</i>	Optional. A numeric value between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A numeric value between 0 and 999 representing the milliseconds

Applies to
Date

setMilliseconds() [JavaScript]

The setSeconds() method is used to set the milliseconds of a specified time.

Syntax

dateObject.setMilliseconds(millisec)

Parameter

<i>millisec</i>	Required. A numeric value between 0 and 999 representing the milliseconds
-----------------	---

Applies to
Date

setMonth() [JavaScript]

The setFullYear() method is used to set the year.

Syntax

dateObject.setFullYear(year, month, day)

Parameters

<i>day</i>	Optional. A numeric value (from 1 to 31) that represents a day in a month
<i>month</i>	Optional. A numeric value between 0 and 11 representing the month
<i>year</i>	Required. A four-digit value representing the year

Applies to
Date

setNamedItem()

Adds an attribute to an element using an attributes collection.

Syntax

newretNode = attributes.setNamedItem(oNamedItem)

Parameter

<i>oNamedItem</i>	Required. An attribute.
-------------------	-------------------------

Applies to
attributes

setRequestHeader() [ie]

Adds custom HTTP headers to the request.

Syntax

XMLHttpRequest.setRequestHeader(sName, sValue)

Parameters

<i>sName</i>	Required. String that specifies the header name.
<i>sValue</i>	Required. String that specifies the header value.

Applies to
XMLHttpRequest

setSeconds() [javaScript]

The setSeconds() method is used to set the seconds of a specified time.

Syntax

dateObject.setSeconds(sec, millisec)

Parameters

<i>sec</i>	Required. A numeric value between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A numeric value between 0 and 999 representing the milliseconds

Applies to
Date

setTime() [javaScript]

The setTime() method is used to calculate a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970.

Syntax

dateObject.setTime(*millisec*)

Parameter

<i>millisec</i>	Required. A numeric value representing the milliseconds since midnight January 1, 1970. Can be a negative number
-----------------	--

Applies to
Date

setTimeout() [ie]

Evaluates an expression after a specified number of milliseconds has elapsed. Returns an identifier that cancels the evaluation with the clearTimeout method.

Syntax

iTimerID = window.setTimeout(*vCode*, *iMilliSeconds* [, *sLanguage*])

Parameters

<i>vCode</i>	Required. Variant that specifies the function pointer or string that indicates the code to be executed when the specified interval has elapsed.
<i>iMilliSeconds</i>	Required. Integerthat specifies the number of milliseconds.
<i>sLanguage</i>	Optional. Stringthat specifies one of the following values: JScript Language is JScript. VBScript Language is VBScript. JavaScript Language is JavaScript.

Applies to
window

setUTCDate() [javaScript]

The setUTCDate() method is used to set the day of the month according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCDate(day)

Parameter

<i>day</i>	Required. A numeric value (from 1 to 31) that represents a day in a month
------------	---

Applies to
Date

setUTCFullYear() [javaScript]

The setUTCFullYear() method is used to set the year according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCFullYear(year, month, day)

Parameter

<i>day</i>	Optional. A numeric value (from 1 to 31) that represents a day in a month
<i>month</i>	Optional. A numeric value between 0 and 11 representing the month
<i>year</i>	Required. A four-digit value representing the year

Applies to
Date

setUTCHours() [javaScript]

The setUTCHours() method is used to set the hour of a specified time according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCHours(hour, min, sec, millisec)

Parameter

<i>hour</i>	Required. A numeric value between 0 and 23 representing the hour
<i>min</i>	Optional. A numeric value between 0 and 59 representing the minutes
<i>sec</i>	Optional. A numeric value between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A numeric value between 0 and 999 representing the milliseconds

Applies to
Date

setUTCMilliseconds() [javaScript]

The setUTCMilliseconds() method is used to set the milliseconds of a specified time according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCMilliseconds(millisec)

Parameter

<i>millisec</i>	Optional. A number between 0 and 999 representing the milliseconds
-----------------	--

Applies to
Date

setUTCMinutes() [javaScript]

The setUTCMinutes() method is used to set the minutes of a specified time according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCMinutes(min, sec, millisec)

Parameter

<i>min</i>	Required. A numeric value between 0 and 59 representing the minutes
<i>sec</i>	Optional. A numeric value between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A numeric value between 0 and 999 representing the milliseconds

Applies to
Date

setUTCMonth() [javaScript]

The setUTCMonth() method is used to set the month according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCMonth(month, day)

Parameter

<i>day</i>	Optional. A numeric value (from 1 to 31) that represents a day in a month
<i>month</i>	Required. A numeric value between 0 and 11 representing the month

Applies to
Date

setUTCSeconds() [javaScript]

The setUTCSeconds() method is used to set the seconds of a specified time according to Universal Coordinated Time (UTC).

Syntax

dateObject.setUTCSeconds(sec, millisec)

Parameters

<i>sec</i>	Optional. A numeric value between 0 and 59 representing the seconds
<i>millisec</i>	Optional. A numeric value between 0 and 999 representing the milliseconds

Applies to
Date

setYear() [JavaScript]

The setYear() method is used to set the year.

Syntax

dateObject.setYear(year)

Parameter

<i>year</i>	Required. A two or four digit number that indicates the year
-------------	--

Applies to
Date

shape

Sets or retrieves the shape of the object.

Syntax

HTML

<ELEMENT SHAPE = *sShape* ... COORDS=... >

Scripting

object.shape [= *sShape*]

sShape values

circ	Circle (COORDS= "x1, y1, r" Where x1, y2 are the coordinates of the center of the circle, and r is the radius of the circle)
circle	Circle (COORDS= "x1, y1, x2, y2...xn, yn" Where each x, y pair contains the coordinates of one vertex of the polygon)
poly	Polygon (COORDS= "x1, y1, x2, y2...xn, yn" Where each x, y pair contains the coordinates of one vertex of the polygon)
polygon	Polygon (COORDS= "x1, y1, x2, y2...xn, yn" Where each x, y pair contains the coordinates of one vertex of the polygon)
rect	Rectangle (COORDS= "x1, y1, x2, y2" Where x1, y1 are the co-ordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the lower-right coordinates of the rectangle)
rectangle	Rectangle (COORDS= "x1, y1, x2, y2" Where x1, y1 are the co-ordinates of the upper-left corner of the rectangle and x2, y2 are the coordinates of the lower-right coordinates of the rectangle)

Applies To
a, area

shift() [JavaScript]

The shift() method is used to remove and return the first element of an array.

Syntax

arrayObject.shift()

Ex

In this example we will create an array, and then remove the first element of the array. Note that this will also change the length of the array:

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Azul"
arr[1] = "Amarelo"
arr[2] = "Vermelho"
document.write(arr + "<br />")
document.write(arr.shift() + "<br />")
document.write(arr)
</script>
```

The output of the code above will be:

Azul, Amarelo, Vermelho

Azul

Amarelo, Vermelho

Applies to

Array

shiftKey [ie]

Sets or retrieves the state of the SHIFT key.

Syntax

[*bKey* =] *event*.shiftKey

bkey values

false	SHIFT key is not pressed
true	SHIFT key is pressed

Applies To
event

shiftLeft [ie]

Retrieves the state of the left SHIFT key.

Syntax

[*bPressed* =] *event*.shiftLeft

bPressed values

false	Left SHIFT key is not pressed
true	Left SHIFT key is pressed

Applies To
event

show() [ie]

Displays the pop-up window on the screen.

Syntax

popup.show(*iX*, *iY*, *iWidth*, *iHeight* [, *oElement*])

Parameter

<i>iX</i>	Required. Integer that specifies the x-coordinate of the pop-up window, in pixels.
<i>iY</i>	Required. Integer that specifies the y-coordinate of the pop-up window, in pixels.
<i>iWidth</i>	Required. Integer that specifies the width of the pop-up window, in pixels.
<i>iHeight</i>	Required. Integer that specifies the height of the pop-up window, in pixels.
<i>oElement</i>	Optional. Object that specifies the element to which the x, y coordinates are relative. If none is given, the x, y coordinates are relative to the desktop, where (0, 0) is the upper left corner.

Applies to
popup

ShowBrowserUI() [ie]

Opens the specified browser dialog box.

Syntax

vReturn = *external*.ShowBrowserUI(*sUI*, *null*)

Parameters

<i>sUI</i>	Required. String that specifies a browser dialog box, using one of the following values.	
LanguageDialog	Opens the Language Preference dialog box.	
OrganizeFavorites	Opens the Organize Favorites dialog box.	
PrivacySettings	Microsoft Internet Explorer 6 and later. Opens the Privacy Preferences dialog box.	
ProgramAccessAndDefaults	Microsoft Windows XP Service Pack 1 (SP1) and later. Opens the Set Program Access and Defaults dialog box.	

Applies to
external

showHelp() [ie]

Displays a Help file. This method can be used with Microsoft HTML Help.

Syntax

window.showHelp(*sURL* [, *vContextID*])

Parameters

<i>sURL</i>	Required. String that specifies the URL of the file to be displayed as help.
<i>vContextID</i>	Optional. Variant that specifies a string or integer that indicates a context identifier in a Help file.

Applies to
window

showModalDialog() [ie]

Creates a modal dialog box that displays the specified HTML document.

Syntax

vReturnValue = *window*.showModalDialog(*sURL* [, *vArguments*] [, *sFeatures*])

Parameters

<i>sURL</i>	Required. String that specifies the URL of the document to load and display.
<i>vArguments</i>	Optional. Variant that specifies the arguments to use when displaying the document. Use this parameter to pass a value of any type, including an array of values. The dialog box can extract the values passed by the caller from the dialogArguments property of the window object.
<i>sFeatures</i>	Optional. String that specifies the window ornaments for the dialog box, using one or more of the following semicolon-delimited values: dialogHeight: <i>sHeight</i> Sets the height of the dialog window

Applies to
window

showModelessDialog() [ie]

Creates a modeless dialog box that displays the specified HTML document.

Syntax

vReturnValue = *window*.showModelessDialog(*sURL* [, *vArguments*] [, *sFeatures*])

Parameters

<i>sURL</i>	Required. String that specifies the URL of the document to load and display.
<i>vArguments</i>	Optional. Variant that specifies the arguments to use when displaying the document. Use this parameter to pass a value of any type, including an array of values. The dialog box can extract the values passed by the caller from the dialogArguments property of the window object.
<i>sFeatures</i>	Optional. Variant of type String that specifies the window ornaments for the dialog box, using one or more of the following semicolon-delimited values: dialogHeight: <i>sHeight</i> Sets the height of the dialog window (see Remarks for default unit of measure). dialogLeft: <i>sXPos</i> Sets the left position of the dialog window relative to the upper-left corner of the desktop. dialogTop: <i>sYPos</i> Sets the top position of the dialog window relative to the upper-left corner of the desktop. dialogWidth: <i>sWidth</i> Sets the width of the dialog window (see Remarks for default unit of measure). center: { yes no 1 0 on off } Specifies whether to center the dialog window within the desktop. The default is yes. dialogHide: { yes no 1 0 on off } Specifies whether the dialog window is hidden when printing or using print preview. This feature is only available when a dialog box is opened from a trusted application. The default is no. edge: { sunken raised } Specifies the edge style of the dialog window. The default is raised. help: { yes no 1 0 on off } Specifies whether the dialog window displays the context-sensitive Help icon. The default is yes.

resizable: { yes no 1 0 on off }	Specifies whether the dialog window has fixed dimensions. The default is no.
scroll: { yes no 1 0 on off }	Specifies whether the dialog window displays scrollbars. The default is yes.
status: { yes no 1 0 on off }	Specifies whether the dialog window displays a status bar. The default is yes for untrusted dialog windows and no for trusted dialog windows.
unadorned: { yes no 1 0 on off }	Specifies whether the dialog window displays the border window chrome. This feature is only available when a dialog box is opened from a trusted application. The default is no.

Applies to
window

sin() [JavaScript]

The sin() method returns the sine of a number.

Syntax

Math.sin(*x*)

Parameters

<i>x</i>	Required. A number
----------	--------------------

Ex

```
<script type="text/javascript">
document.write(Math.sin(3) + "<br />")
document.write(Math.sin(-3) + "<br />")
document.write(Math.sin(0) + "<br />")
document.write(Math.sin(Math.PI) + "<br />")
document.write(Math.sin(Math.PI/2))
</script>
```

The output of the code above will be:

```
0.1411200080598672
-0.1411200080598672
0
1.2246063538223772e-16
1
```

Applies to
Math

size (hr)

Sets or retrieves the height of the hr object. Deprecated.

Syntax

HTML

<hr SIZE = *iSize*... >

Scripting

hr.size [= *iSize*]

Applies To
hr

size (select)

Sets or retrieves the number of rows in the list box.

Syntax

HTML

<select SIZE = *iRows*... >

Scripting

select.size [= *iRows*]

Applies To
select

size (basefont)

Sets or retrieves the font size of the object.

Syntax

HTML

<ELEMENT SIZE = *iSize*... >

Scripting

object.size [= *iSize*]

Applies To

basefont, font

size

(input)

Sets or retrieves the size of the control.

Syntax

HTML

<ELEMENT SIZE = *iRows*... >

Scripting

ELEMENT.size [= *iRows*]

Applies To

input

SkipRunOnce() [ie]

Allows the user to select “first run” settings at a later time.

Syntax

window.external.SkipRunOnce()

Applies to

external

slice() [javaScript]

The slice() method returns selected elements from an existing array or a part from a string, returning the the extracted part in a new string or array.

You can use negative numbers to select from the end of the array.

Syntax

arrayObject.slice(iStart, iend)

stringObject.slice(start, end)

Parameters

<i>iStart</i>	Required. Specify where to start the selection. Must be a number
<i>iEnd</i>	Optional. Specify where to end the selection. Must be a number

Ex

In this example we will create an array, and then display selected elements from it:

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Marx"
arr[1] = "Lenin"
arr[2] = "Trotsky"
document.write(arr + "<br />")
document.write(arr.slice(1) + "<br />")
document.write(arr)
</script>
```

The output of the code above will be:

Marx, Lenin, Trotsky

Lenin, Trotsky

Marx, Lenin, Trotsky

In this example we will extract all characters from a string, starting at position 6:

```
<script type="text/javascript">
var str="Hello happy world!"
document.write(str.slice(6))
</script>
```

The output of the code above will be:

happy world!

Applies to

Array, String

<small>

Specifies that the enclosed text should be displayed in a smaller font.

Ex

```
<small>Small text</small>
```

Standard Attributes

id, class, title, style, dir, lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

small() [javaScript]

The small() method is used to display a string in a small font.

Syntax

stringObject.small()

Ex

In this example “argh!” will be displayed in a small font:

```
<script type="text/javascript">
var str="argh!"
document.write(str.small())
</script>
```

Applies to

String

sort() [javaScript]

The sort() method is used to sort the elements of an array. W

ll sort the elements alphabetically by default. However, this means that numbers will not be sorted correctly (40 comes before 5). To sort numbers, you must create a function that compare numbers.

Syntax

arrayObject.sort(*sortby*)

Parameter

<i>sortby</i>	Optional. Specifies the sort order. Must be a function
---------------	--

Ex

```
<script type="text/javascript">
var arr = new Array(6)
arr[0] = "Sinatra"
arr[1] = "Brubeck"
arr[2] = "Miles"
arr[3] = "Ella"
arr[4] = "Duke"
arr[5] = "Coltrane"
document.write(arr + "<br />")
document.write(arr.sort())
</script>
```

The output of the code above will be:

Sinatra, Brubeck, Miles, Ella, Duke, Coltrane

Brubeck, Coltrane, Duke, Ella, Miles, Sinatra

Applies to

Array

sourceIndex [ie]

Retrieves the ordinal position of the object, in source order, as the object appears in the document’s all collection.

Syntax

HTML

N/A

Scripting

[*lIndex* =] *object*.sourceIndex

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

Specifies an inline text container.

Ex

<P>This paragraph contains a single blue word.

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children, filters

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, doScroll, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

span

Sets or retrieves the number of columns in the group.

Syntax

HTML

<ELEMENT span = *iSpan*... >

Scripting

object.span [= *iSpan*]

Applies To

col, COLGROUP

specified

Retrieves whether an attribute has been specified.

Syntax

[*bSpecified* =] *attribute*.specified

bSpecified values

false	Attribute is not specified.
true	Attribute is specified.

Applies To

attribute

splice() [JavaScript]

The splice() method is used to remove and add new elements to an array.

Syntax

arrayObject.splice(*index*, *howmany*, *element1*,, *elementX*)

Parameters

<i>index</i>	Required. Specify where to add/remove elements. Must be a number
<i>howmany</i>	Required Specify how many elements should be removed. Must be a number, but can be "0"
<i>element1</i>	Optional. Specify a new element to add to the array
<i>elementX</i>	Optional. Several elements can be added

Ex

In this example we will create an array and add an element to it:

```
<script type="text/javascript">
var arr = new Array(5)
arr[0] = "Sinatra"
arr[1] = "Brubeck"
arr[2] = "Miles"
arr[3] = "Ella"
arr[4] = "Duke"
document.write(arr + "<br />")
arr.splice(2, 0, "Coltrane")
document.write(arr + "<br />")
</script>
```

The output of the code above will be:

Sinatra, Brubeck, Miles, Ella, Duk

Sinatra, Brubeck, Miles, Ella, Duke, Coltrane

Applies to

Array

split() [JavaScript]

The split() method is used to split a string into an array of strings.

Syntax

stringObject.split(separator, howmany)

Parameters

<i>separator</i>	Required. Specifies the character, regular expression, or substring that is used to determine where to split the string
<i>howmany</i>	Optional. Specify how many times split should occur. Must be a numeric value

Ex

In this example we will split up a string in different ways:

```
<script type="text/javascript">
var str="How are you doing today?"
document.write(str.split(" ") + "<br />")
document.write(str.split("") + "<br />")
document.write(str.split(" ", 3))
</script>
```

The output of the code above will be:

How, are, you, doing, today?

H, o, w, , a, r, e, , y, o, u, , d, o, i, n, g, , t, o, d, a, y, ?

How, are, you

Applies to

String

splitText()

Divides a text node at the specified index. Returns a text node object.

Syntax

oSplitNode = *TextNode*.splitText([*lIndex*])

Parameter	
<i>iIndex</i>	Optional. Integer that specifies the index of the string that indicates where the separation occurs. If a value is not provided, a new text node with no value is created.
Applies to	TextNode

SqmEnabled()

Returns a boolean value that determines whether Software Quality Monitoring (SQM) is enabled.

Syntax
fEnabled = *window.external.SqmEnabled()*

Applies to
external

sqrt() [javaScript]

The sqrt() method returns the square root of a number.

Syntax
Math.sqrt(x)
Parameter

<i>x</i>	Required. A number
----------	--------------------

Ex
<script type="text/javascript">
document.write(Math.sqrt(0) + " , ")
document.write(Math.sqrt(1) + " , ")
document.write(Math.sqrt(9) + " , ")
document.write(Math.sqrt(0.64) + " , ")
document.write(Math.sqrt(-9))
</script>
The output of the code above will be:
0, 1, 3, 0.8, NaN

Applies to
Math

SQRT1_2 [javascript]

Returns the square root of 1/2 (approx. 0.707)

Applies To
Math

SQRT2 [javascript]

Returns the square root of 2 (approx. 1.414)

Applies To
Math

src

(bgsound)
Sets or retrieves the URL of a sound to play.

Syntax
HTML
<ELEMENT SRC = *sURL*... >
Scripting
object.src [= *sURL*]
Applies To
bgsound

src

(script)
Retrieves the URL to an external file that contains the source code or data.

Syntax
HTML
<script SRC = *sURL*... >
Scripting
script.src [= *sURL*]
Applies To
script

src

(applet, embed, ...)
Sets or retrieves a URL to be loaded by the object.

Syntax

HTML
<ELEMENT SRC = *sURL*... >
Scripting

object.src [= *sURL*]
Applies To
applet, embed, frame, iframe, img, input type=image, XML

srcElement [ie]

Sets or retrieves the object that fired the event.

Syntax
event.srcElement [= *oObject*]
Applies To
event

srcFilter [ie]

Sets or retrieves the filter object that caused the onfilterchange event to fire.

Syntax
event.srcFilter [= *sSrcFilter*]
Applies To
event

srcUrn [ie]

Retrieves the Uniform Resource Name (URN) of the behavior that fired the event. Null is the default value for IE.

Syntax
event.srcUrn [= *sUrn*]
Applies To
event

standby

Sets or retrieves a message to be displayed while an object is loading.

Syntax
HTML
<object STANDBY = *sStand*... >
Scripting
object.standby(v) [= *sStand*]
Applies To
object

start

(ol)
Sets or retrieves the starting number for an ordered list.

Syntax
HTML
<ol START = *iStart*... >
Scripting
ol.start [= *iStart*]
Applies To
ol

start

(img, input)
Sets or retrieves when a video clip file should begin playing.

Syntax
HTML
<ELEMENT start = *sStart*... >
Scripting
object.start [= *sStart*]
sStart values

| | |
|-----------|---|
| fileopen | Default. Video begins as soon as it finishes loading |
| mouseover | Video begins when the user moves the mouse over the animation |

Applies To
img, input, input type=image

start() [ie]

Starts scrolling the marquee.

Syntax

marquee.start()

Applies to

marquee

status [ie]

(input)

Sets or retrieves the value indicating whether the control is selected

Syntax

HTML

N/A

Scripting

object.status [= *bStatus*]

bStatus values

| | |
|-------|----------------------------------|
| false | Default. Control is not selected |
| true | Control is selected |
| null | Control is not initialized |

Applies To

input type=checkbox, input type=radio, textarea

status [ie]

(window)

Sets or retrieves the value indicating whether the control is selected.

Syntax

window.status [= *sStatus*]

Applies To

window

status

(XMLHttpRequest)

Retrieves the HTTP status code of the request.

Syntax

[*nStatus* =] *XMLHttpRequest.status*

Applies To

XMLHttpRequest

statusText

Retrieves the friendly HTTP status of the request.

Syntax

[*sStatus* =] *XMLHttpRequest.statusText*

sStatus values

| | |
|-------|--|
| false | Default. Object is opaque. It cannot be transparent. |
| true | Object can be transparent. |

Applies To

XMLHttpRequest

stop() [ie]

Stops the marquee from scrolling.

Syntax

marquee.stop()

Applies to

marquee

<strike>

Renders text in strike-through type.

Ex

A new version is <strike>not yet available.</strike> now available!

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

strike() [javaScript]

The strike() method is used to display a string with a strikethrough.

Syntax

stringObject.strike()

Applies to

String

String [javascript]

Os objetos string são utilizados para manipular um pedaço armazenado de texto

Methods

anchor, big, blink, bold, charAt, charCodeAt, concat, fixed, fontcolor, fontsize, fromCharCode, indexOf, italics, lastIndexOf, link, match, replace, search, slice, small, split, strike, sub, substr, substring, sup, toLowerCase, toUpperCase, toSource, valueOf()

Properties

constructor, length, prototype

String() [javaScript]

The String() function converts the value of an object to a string.

Syntax

String(*object*)

Ex

```
<script type="text/javascript">
var test1= new Boolean(1)
var test2= new Boolean(0)
var test3= new Boolean(true)
var test4= new Boolean(false)
var test5= new Date()
var test6= new String("999 888")
var test7=12345
document.write(String(test1)+ " , ")
document.write(String(test2)+ " , ")
document.write(String(test3)+ " , ")
document.write(String(test4)+ "<br />")
document.write(String(test5)+ "<br />")
document.write(String(test6)+ " , ")
document.write(String(test7))
</script>
```

The output of the code above will be:
true, false, true, false
Wed Jun 28 2006 16:27:37 GMT-0300 (Hora oficial do Brasil)
999 888, 12345

Applies to

all of the built-in JavaScript objects

Renders text in bold.

Ex

Strong text

Standard Attributes

id, class, title, style, dir, lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, msInterpolationMode, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

<style>

Specifies a style sheet for the page. The style element goes in the head section. If you want to include a style sheet in your page, you should define the style sheet externally, and link to it using <link>.

Ex

```
<head>
<style type="text/css">
h1 {color: red}
h3 {color: blue}
</style>
</head>
```

Required Attributes

type (text/css)

Optional Attributes

media (screen, tty, tv, projection, handheld, print, braille, aural, all)

Standard Attributes

title, dir, lang, xml:space

Collection

behaviorUrns

Events

onerror, onreadystatechange

Methods

addBehavior, dragDrop, removebehavior, removeNode

Style Properties and attributes

media, type

style

(object)

Represents the current settings of all possible inline styles for a given element.

Property

onOffBehavior

Methods

getAttributeNode, getExpression, normalize, removeAttribute, removeAttributeNode, removeExpression, setAttribute, setAttributeNode, setExpression

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderCollapse, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cssText, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, height, imeMode, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyle, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minHeight, minWidth, msInterpolationMode, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, rubyAlign, rubyOverhang, rubyPosition, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, tableLayout, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, verticalAlign, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

style

Sets an inline style for the element.

Syntax

HTML

<ELEMENT STYLE=*sStyle* ... >

Scripting

N/A

Applies To

a, abbr, acronym, address, applet, area, b, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, map, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

styleFloat

Sets or retrieves on which side of the object the text will flow.

Syntax

CSS

{ float : *sFloat* }

Scripting

object.style.styleFloat [= *sFloat*]

sFloat values

none	Default. Object displays where it appears in the text
left	Text flows to the right of the object
right	Text flows to the left of the object

Applies To

a, address, applet, b, big, blockquote, button, center, cite, code, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee,

menu, object, ol, p, pre, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

styleSheet

Represents a single style sheet in the document.

Standard Attributes

id, title

Collection

imports, pages, rules

Methods

addImport, addPageRule, addRule, fireEvent, removeRule

Object

page

Style Properties and attributes

cssText, media, textAutospace

styleSheets [ie]

Retrieves a collection of styleSheet objects representing the style sheets that correspond to each instance of a link or style object in the document.

Syntax

[*oColl* =] document.styleSheets

[*oObject* =] document.styleSheets(*vIndex* [, *iSubIndex*])

Possible Values

<i>oColl</i>	Array of styleSheet objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>iIndex</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when <i>vIndex</i> is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by <i>iSubIndex</i> .

Applies To

document

<sub>

Specifies that the enclosed text should be displayed in subscript, using a smaller font than the current font.

Ex

This text contains _{subscript}

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight,

borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

sub() [JavaScript]

The sub() method is used to display a string as subscript.

Syntax

stringObject.sub()

Applies to

String

submit()

Submits the form.

Syntax

FORM.submit()

Applies to

FORM

substr() [JavaScript]

The substring() method extracts the characters in a string between two specified indices. To extract characters from the end of the string, use a negative start number, the start index starts at 0.

Syntax

stringObject.substring(*start*, *stop*)

Parameters

<i>start</i>	Required. Where to start the extraction. Must be a numeric value
<i>stop</i>	Optional. Where to stop the extraction. Must be a numeric value

Ex

```
<script type="text/javascript">
var str="Hello world!"
document.write(str.substring(3))
</script>
```

Applies to

String

substringData()

Extracts a range of characters from the object. Returns a reference to the substring.

Syntax

sSubstring = object.substringData(*nOffset*, *nCount*)

Parameters

<i>nOffset</i>	Required. Integer that specifies the offset from which to start.
<i>nCount</i>	Required. Integer that specifies the number of characters to extract.

Applies to

comment, TextNode

summary

Sets or retrieves a description and/or structure of the object.

Syntax

HTML

<table SUMMARY = *sSummary*... >

Scripting

table.summary(*v*) [= *sSummary*]

Applies To

table

<sup>

Specifies that the enclosed text should be displayed in superscript, using a smaller font than the current font.

Ex

This text contains ^{superscript}

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

sup() [JavaScript]

The sup() method is used to display a string as superscript.

Syntax

stringObject.sup()

Ex

<script type="text/javascript">
var str="Hello world!"
document.write(str.sup())
</script>

Applies to

String

swapNode() [ie]

Exchanges the location of two objects in the document hierarchy.

Syntax

oSwapped = object.swapNode(oNode)

Parameter

oNode	Required. Object that specifies the existing element.
-------	---

Applies to

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

switch

The switch statement evaluates an expression, matching the expression's value to a case label, and executes statements associated with that case.

If a match is found, the program executes the associated statements. If multiple cases match the provided value, the first case that matches is selected, even if the cases are not equal to each other.

Syntax

```
switch (expression) {  
    case label1:  
        statements1  
        [break;]  
    case label2:  
        statements2  
        [break;]  
    ...  
    case labelN:  
        statementsN  
        [break;]  
    default:  
        statements_def  
        [break;]  
}
```

Parameters

expression	An expression matched against each label.
labelN	Identifier used to match against expression.
statementsN	Statements that are executed if expression matches the associated label.
statements_def	Statements that are executed if expression does not match any label.

Ex

In the following example, if expression evaluates to "Bananas", the program matches the value with case "Bananas" and executes the associated statement. When break is encountered, the program breaks out of switch and executes the statement following switch. If break were omitted, the statement for case "Cherries" would also be executed.

```
switch (expr) {  
    case "Oranges":  
        document.write("Oranges are $0.59 a pound.<br>");  
        break;  
    case "Apples":  
        document.write("Apples are $0.32 a pound.<br>");  
        break;  
    case "Bananas":  
        document.write("Bananas are $0.48 a pound.<br>");  
        break;  
    case "Cherries":  
        document.write("Cherries are $3.00 a pound.<br>");  
        break;  
    case "Mangoes":  
    case "Papayas":  
        document.write("Mangoes and papayas are $2.79 a pound.<br>");  
        break;  
    default:  
        document.write("Sorry, we are out of " + i + ".<br>");  
}  
document.write("Is there anything else you'd like?<br>");
```

systemLanguage [ie]

Retrieves the default language used by the operating system.

Syntax

[sLanguage =] object.systemLanguage

Applies To

clientInformation, navigator

tabIndex

Sets or retrieves the index that defines the tab order for the object. Negative numbers removes objects from the tab order

Syntax

HTML

<ELEMENT TABINDEX = *i*l*ndex*... >

Scripting

object.tabIndex [= *i*l*ndex*]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

<table>

Specifies that the contained content is organized into a table with rows and columns. Inside a <table> tag you can put table headers, table rows, table cells, and other tables.

Ex

<table border = "1">
<tr>
<td>Cell A</td>
<td>Cell B</td>
</tr>
</table>

Optional Attributes

align(deprecated), bgcolor (depracated), border, cellpadding, cellspacing, frame, rules, summary, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

tableLayout

Sets or retrieves a string that indicates whether the table layout is fixed.

Syntax

CSS

{ table-layout : *sLayout* }

Scripting

object.style.tableLayout [= *sLayout*]

sLayout values

auto	Default. Column width is set by the widest unbreakable content in the column cells
fixed	Table and column widths are set either by the sum of the widths on the col objects or, if these are not specified, by the width of the first row of cells. If no width is specified for the table, it renders by default with width=100%

Applies To

currentStyle, runtimeStyle, style, table

tabStop [ie]

Sets or retrieves whether an element behavior can receive focus and participate in the tabbing sequence.

Syntax

defaults.tabStop(*v*) [= *bTabEnabled*]

bTabEnabled values

false	Default. Element cannot participate in tabbing
true	Element can participate in tabbing

Applies To

defaults

tags() [ie]

Retrieves a collection of objects that have the specified HTML tag name. Returns a collection of element objects if successful, or null otherwise.

Syntax

colElements = *object*.tags(*sTag*)

Parameters

<i>sTag</i>	Required. Variant of type String that specifies an HTML tag. It can be any one of the objects exposed by the DHTML Object Model.
-------------	--

Applies to

all, anchors, applets, areas, boundElements, cells, children, elements, embeds, forms, images, links, options, plugins, rows, scripts, tBodies

tagName

Retrieves the tag name of the object.

Syntax

Scripting

[*sName* =] *object*.tagName

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, custom, dd, del, dfn, dir, div, dl, !DOCTYPE, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, HTML Comment, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, meta, nextID, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

tagUrn [ie]

Sets or retrieves the URN specified in the namespace declaration. If the namespace declaration is omitted, the value will be null.

Syntax

Scripting

object.tagUrn [= *sUrn*]

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup,



comment, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, map, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, wbr, xml, xmp

taintEnabled() [ie]

Returns a boolean value indicating whether data tainting is enabled.

Syntax

bEnabled = object.taintEnabled()

Applies to

clientInformation, navigator

tan() [JavaScript]

The tan() method returns a number that represents the tangent of an angle.

Syntax

Math.tan(x)

Parameter

x	Required. A number
---	--------------------

Ex

```
<script type="text/javascript">
document.write(Math.tan(0.50) + " ");
document.write(Math.tan(-0.50) + " ");
document.write(Math.tan(5) + "<br />");
document.write(Math.tan(10) + " ");
document.write(Math.tan(-5) + " ");
document.write(Math.tan(-10))
</script>
```

The output of the code above will be:

0.5463024898437905, -0.5463024898437905, -3.380515006246586
0.6483608274590866, 3.380515006246586, -0.6483608274590866

Applies to

Math

target

Sets or retrieves the window or frame at which to target content.

Syntax

HTML

<ELEMENT TARGET = sTarget...>

Scripting

object.target [= sTarget]

sTarget values

name	The name of the target window or frame
_blank	Load the linked document into a new blank window. This window is not named
_media	Load the linked document into the HTML content area of the Media Bar. Available in Microsoft Internet Explorer 6 or later
_parent	Load the linked document into the immediate parent of the document the link is in
_search	Load the linked document into the browser search pane. Available in Internet Explorer 5 or later
_self	Default. Load the linked document into the window in which the link was clicked (the active window)
_top	Load the linked document into the topmost window

Applies To

a, area, base, form, link

<tBody>

Designates rows as the body of the table.

Ex

```
<table border = "1">
<thead>
<tr>
<td>This text is in the thead</td>
</tr>
```

```
</thead>
<tfoot>
<tr>
<td>This text is in the tfoot</td>
</tr>
</tfoot>
<tbody>
<tr>
<td> This text is in the tbody</td>
</tr>
</tbody>
</table>
```

Optional Attributes

align, char, charoff, valign

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, deleteRow, deleteTFoot, deleteTHead, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, insertRow, mergeAttributes, moveRow, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollToView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, layoutGrid, layoutGridMode, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordSpacing, zIndex, zoom

tBodies

Retrieves a collection of all tBody objects in the table. Objects in this collection are in source order.

Syntax

[oColl =] TABLE.tBodies

[oObject =] TABLE.tBodies(vIndex [,iSubIndex])

Possible Values

oColl	Array of tBody objects.
oObject	Reference to an individual item in the array of elements contained by the object.
iIndex	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
iSubIndex	Optional. Position of an element to retrieve. This parameter is used when vIndex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To

table

<td>

Specifies a cell in a table.

Ex

```
<table border = "1">
<tr>
<td>Cell A</td>
<td>Cell B</td>
</tr>
</table>
```

Optional Attributes

abbr, align, axis, bgcolor, char, charoff, colspan, headers, height, nowrap, scope, valign, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children, filters

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minheight, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

text

(body)

Sets or retrieves the text (foreground) color for the document body.

Syntax

HTML

```
<body TEXT = sColor... >
```

Scripting

```
body.text [ = sColor ]
```

Applies To

body

text

(comment)

Retrieves or sets the text of the object as a string.

Syntax

```
object.text [ = sTxt ]
```

Applies To

script, title, comment, HTML Comment

text

(option)

Sets or retrieves the text string specified by the option tag.

Syntax

```
option.text [ = sTxt ]
```

Applies To

option

text

(TextRange)

Sets or retrieves the text contained within the range.

Syntax

```
TextRange.text [ = sTxt ]
```

Applies To

extRange

textAlign

Sets or retrieves whether the text in the object is left-aligned, right-aligned, centered, or justified.

Syntax

CSS

```
{ text-align : sAlign }
```

Scripting

```
object.style.textAlign [ = sAlign ]
```

sAlign values

left	Default. Text is aligned to the left
right	Text is aligned to the right
center	Text is centered
justify	Text is justified

Applies To

blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, input type=password, input type=text, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, textarea, th, tr, ul, xmp

textAlignLast [ie]

Sets or retrieves how to align the last line or only line of text in the object.

Syntax

CSS

```
{ text-align-last : sAlignLast }
```

Scripting

```
object.style.textAlignLast(v) [ = sAlignLast ]
```

sAlignLast values

auto	Default. Text is aligned like the other lines in the object, using the value of the textAlign property
center	Text is centered
inherit	Text is aligned like the text in the parent object
justify	Text is justified
left	Text is aligned to the left
right	Text is aligned to the right

Applies To

address, blockquote, body, center, currentStyle, custom, dd, div, dl, dt, fieldset, form, hn, hr, isindex, li, listing, menu, ol, p, plaintext, pre, runtimeStyle, style, ul, xmp

<textarea>

Specifies a multiline text input control.

Ex

```
<textarea rows="2" cols="20">
```

The cat was playing in the garden. Suddenly a dog showed up.....

```
</textarea>
```

Required Attributes

cols, rows

Optional Attributes

disables, name, readonly

Standard Attributes

id, class, title, style, dir, lang, xml:lang, tabindex, accesskey

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, createTextRange, detachEvent, doScroll, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, select, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, imeMode, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, scrollbar3dLightColor, scrollbarArrowColor, scrollbarBaseColor, scrollbarDarkShadowColor, scrollbarFaceColor, scrollbarHighlightColor, scrollbarShadowColor, scrollbarTrackColor, styleFloat, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

textAutospace

Sets or retrieves the autospacing and narrow space width adjustment of text.

Syntax

CSS

{ text-autospace : *sIdeograph* }

Scripting

object.style.textAutospace [= *sIdeograph*]

sIdeograph values

none	Default. No effect takes place that is, no extra space is added
ideograph-alpha	Creates extra spacing between runs of ideographic and non-ideographic text, such as Latin-based, Cyrillic, Greek, Arabic, or Hebrew text
ideograph-numeric	Creates extra spacing between runs of ideographic text and numeric characters
ideograph-parenthesis	Creates extra spacing between a normal (non-wide) parenthesis and an ideograph
ideograph-space	Extends the width of the space character when it is adjacent to ideographs

Applies To

a, abbr, acronym, address, applet, b, base, basefont, bdo, bgsound, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, styleSheet, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xml, xmp

textDecoration

Sets or retrieves a value that indicates whether the text in the object has blink, line-through, overline, or underline decorations.

Syntax

CSS

{ text-decoration : *sDecoration* }

Scripting

object.style.textDecoration [= *sDecoration*]

sDecoration values

none	Default. Text has no decoration.
underline	Text is underlined.
overline	Text has a line over it.
line-through	Text has a line drawn through it.
blink	Blinking. Not implemented by some browsers like IE

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=file, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textDecorationBlink [ie]

Sets or retrieves a Boolean value that indicates whether the object's textDecoration property has a value of "blink."

Syntax

object.style.textDecorationBlink [= *bBlink*]

bBlink values

true	The textDecoration property is set to "blink."
false	The textDecoration property is not set to "blink."

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textDecorationLineThrough [ie]

Sets or retrieves a Boolean value indicating whether the text in the object has a line drawn through it.

Syntax

object.style.textDecorationLineThrough [= *bLineThrough*]

bLineThrough values

false	Prevent the line-through
true	Apply the line-through

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textDecorationNone [ie]

Sets or retrieves the Boolean value indicating whether the textDecoration property for the object has been set to none.

Syntax

object.style.textDecorationNone [= *bDecoration*]

bDecoration values

false	Property is not set to none
true	Property is set to none

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html,

i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textDecorationOverline [ie]

Sets or retrieves a Boolean value indicating whether the text in the object has a line drawn over it.

Syntax

object.style.textDecorationOverline [= *bOverline*]

bOverline values

false	A line is not drawn over the text
true	A line is drawn over the text

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textDecorationUnderline [ie]

Sets or retrieves whether the text in the object is underlined.

Syntax

object.style.textDecorationUnderline [= *bUnderline*]

bUnderline values

false	Prevent the underline
true	Apply the underline

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textIndent

Sets or retrieves the indentation of the first line of text in the object.

Syntax

{ text-indent : *sIndent* }

Scripting

object.style.textIndent [= *sIndent*]

sIndent values

<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	Integer, followed by a percent sign (%). This value is a percentage of the width of the parent object

Applies To

blockquote, body, button, center, currentStyle, dd, dfn, dir, div, dl, dt, fieldset, form, hn, hr, isindex, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

textJustify

Sets or retrieves the type of alignment used to justify text in the object.

Syntax

CSS

{ text-justify : *sAlign* }

Scripting

object.style.textJustify [= *sAlign*]

sAlign values

auto	Default. Allows the browser to determine which justification algorithm to apply.
distribute	Handles spacing much like the newspaper value. This form of justification is optimized for documents in Asian languages, such as Thai.

distribute-all-lines	Justifies lines of text in the same way as the distribute value, except that it also justifies the last line of the paragraph. This form of justification is intended for ideographic text.
distribute-center-last	Not implemented.
inter-cluster	Justifies lines of text that contain no inter-word spacing. This form of justification is optimized for documents in Asian languages.
inter-ideograph	Justifies lines of ideographic text, and increases or decreases both inter-ideograph and inter-word spacing.
inter-word	Aligns text by increasing the space between words. This value's spacing behavior is the fastest way to make all lines of text equal in length. Its justification behavior does not affect the last line of the paragraph.
kashida	Justifies lines of text by elongating characters at chosen points. This form of justification is intended for Arabic script languages. Supported in Microsoft Internet Explorer 5.5 and later.
newspaper	Increases or decreases the spacing between letters and between words. It is the most sophisticated form of justification for Latin alphabets

Applies To

address, blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

textKashidaSpace [ie]

Sets or retrieves the ratio of kashida expansion to white space expansion when justifying lines of text in the object. Part of a proposed addition to Cascading Style Sheets (CSS)

Syntax

CSS

{ text-kashida-space : *vKashida* }

Scripting

object.style.textKashidaSpace(v) [= *vKashida*]

vKashida values

%	Integer, followed by a % . The value is the ratio of kashida expansion to white space expansion. 100% specifies kashida expansion only, and 0% specifies white space expansion only
inherit	Text is expanded like the text in the parent object

Applies To

address, blockquote, body, center, currentStyle, custom, dd, div, dl, dt, fieldset, form, hn, hr, isindex, li, listing, menu, ol, p, plaintext, pre, runtimeStyle, style, ul, xmp

TextNode

Represents a string of text as a node in the document hierarchy.

Ex

This example uses the TextNode object to change the text of an li object.

```
<script>
function fnChangeText(){
    var oTextNode = document.createTextNode("New List Item 1");
    var oReplaceNode = oltem1.firstChild.replaceNode(oTextNode); }
</script>

<UL onclick = "fnChangeText()">
<li ID = oltem1>List Item 1
</UL>
```

Properties

Property	Description
data	Sets or retrieves the value of a TextNode object.
length	Retrieves the number of characters in a TextNode object.
nextSibling	Retrieves a reference to the next child of the parent for the object.
nodeName	Retrieves the name of a particular type of node.
nodeType	Retrieves the type of the requested node.
nodeValue	Sets or retrieves the value of a node.

parentNode	Retrieves the parent object in the document hierarchy.
previousSibling	Retrieves a reference to the previous child of the parent for the object.

Methods

appendData, deleteData, insertData, replaceData, splitText, substringData

textOverflow [ie]

Sets or retrieves a value that indicates whether to render ellipses (...) to indicate text overflow. A Microsoft extension to Cascading Style Sheets (CSS)

Syntax

CSS

{ text-overflow : *iTextOverflow* }

Scripting

object.style.textOverflow(v) [= *iTextOverflow*]

iTextOverflow values

ellipsis	Display ellipses(...) for text overflow
clip	Default. Simply clip the content and do not display ellipses for text overflow

Applies To

frame, frameset, layoutrect, table, td, th, tr

TextRange [ie]

Represents text in an HTML element.

Properties

boundingHeight	Retrieves the height of the rectangle that bounds the TextRange object.
boundingLeft	Retrieves the distance between the left edge of the rectangle that bounds the TextRange object and the left side of the object that contains the TextRange.
boundingTop	Retrieves the distance between the top edge of the rectangle that bounds the TextRange object and the top side of the object that contains the TextRange.
boundingWidth	Retrieves the width of the rectangle that bounds the TextRange object.
htmlText	Retrieves the HTML source as a valid HTML fragment.
offsetLeft	Retrieves the calculated left position of the object relative to the layout or coordinate parent, as specified by the offsetParent property.
offsetTop	Retrieves the calculated top position of the object relative to the layout or coordinate parent, as specified by the offsetParent property.
text	Sets or retrieves the text contained within the range.

Methods

collapse, compareEndpoints, duplicate, execCommand, expand, findText, getBookmark, getBoundingClientRect, getClientRects, inRange, isEqual, move, moveEnd, moveStart, moveToBookmark, moveToElementText, moveToPoint, parentElement, pasteHTML, queryCommandEnabled, queryCommandIndeterm, queryCommandState, queryCommandSupported, queryCommandValue, scrollIntoView, select, setEndPoint

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

TextRange [ie]

Retrieves a collection of TextRange objects.

Syntax

[*oColl* =] selection.TextRange

[*oObject* =] selection.TextRange(*vIndex* [, *iSubIndex*])

Possible Values

<i>oColl</i>	Array of tBody objects.
<i>oObject</i>	Reference to an individual item in the array of elements contained by the object.
<i>index</i>	Required. Integer or string that specifies the element or collection to retrieve. If this parameter is an integer, the method returns the element in the collection at the given position, where the first element has value 0, the second has 1, and so on. If this parameter is a string and there is more than one element with the name or id property equal to the string, the method returns a collection of matching elements.
<i>iSubIndex</i>	Optional. Position of an element to retrieve. This parameter is used when vindex is a string. The method uses the string to construct a collection of all elements that have a name or id property equal to the string, and then retrieves from this collection the element at the position specified by iSubIndex.

Applies To

selection

TextRectangle

Specifies a rectangle that contains a line of text in either an element or a TextRange object.

Properties

bottom, left, right, top

TextRectangle [ie]

A collection of TextRectangle objects returned by the getClientRects method.

Ex

This example shows how to use the getClientRects method and the TextRectangle collection to iterate through the lines of text in an object:

```
<SCRIPT>
function newHightite(obj) {
  oRcts = obj.getClientRects();
  iLength = oRcts.length
  for (i = 0; i < iLength; i++)
  {
    alert("Line number " + (i + 1) + " is " +
      (oRcts(i).right - oRcts(i).left) + " pixels wide.") }
  }
}</SCRIPT>
```

textTransform

Sets or retrieves the rendering of the text in the object.

Syntax

CSS

{ text-transform : *sTransform* }

Scripting

object.style.textTransform [= *sTransform*]

sTransform values

none	Default. Text is not transformed.
capitalize	Transforms the first character of each word to uppercase.
uppercase	Transforms all the characters to uppercase.
lowercase	Transforms all the characters to lowercase.

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

textUnderlinePosition [ie]

Sets or retrieves the position of the underline decoration that is set through the textDecoration property of the object. A Microsoft extension to Cascading Style Sheets (CSS)

Syntax

CSS

{ text-underline-position : *sPosition* }

Scripting

object.style.textUnderlinePosition [= *sPosition*]

sPosition values

above	Decoration appears above the text.
below	Decoration appears below the text.
auto	Default. Microsoft Internet Explorer 6 and later. Decoration appears above the text if the LANG attribute is set to ja, which is the language code abbreviation for the Japanese language, and the writing-mode attribute is set to tb-rl, which causes vertical inline text progression. If not, the decoration appears below the text.
auto-pos	Internet Explorer 6 and later. Identical to auto

Applies To

a, abbr, acronym, address, applet, b, base, basefont, bdo, bgsound, big, blockquote, body, button, caption, center, cite, code, col, colgroup, comment, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frameset, head, hn, hr, html, i, img, input, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, link, listing, marquee, menu, nextID, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xml, xmp

<tfoot>

Designates rows as the table's footer. If you use the thead, tfoot and tbody elements, you must use every element. They should appear in this order: <thead>, <tfoot> and <tbody>, so that browsers can render the foot before receiving all the data. You must use these tags within the table element.

Ex

```
<table border = "1">
<thead>
<tr>
<td>This text is in the thead</td>
</tr>
</thead>
<tfoot>
<tr>
<td>This text is in the tfoot</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>This text is in the tbody</td>
</tr>
</tbody>
</table>
```

Optional Attributes

align, char, charoff, valign

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, deleteRow, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, insertRow, mergeAttributes, moveRow, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight,

layoutGrid, layoutGridMode, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordSpacing, zIndex, zoom

tFoot

Retrieves the tFoot object of the table.

Syntax

HTML

N/A

Scripting

[*oTFoot* =] *table.tFoot*

Applies To

table

<th>

Specifies a header column. Header columns are centered within the cell and are bold.

Ex

```
<table border = "1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>Cell A</td>
<td>Cell B</td>
</tr>
</table>
```

Optional Attributes

abbr, align, axis, bgcolor, char, charoff, colspan, headers, height, nowrap, rowspan, scope, valign, width

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, filters

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, letterSpacing, lineBreak, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minHeigh, minWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

<tHead>

Designates rows as the table's header.

```
Ex
<table border = "1">
<thead>
<tr>
<td>This text is in the thead</td>
</tr>
</thead>
<tfoot>
<tr>
<td>This text is in the tfoot</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>This text is in the tbody</td>
</tr>
</tbody>
</table>
```

Optional Attributes

align, char, charoff, valign

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children, rows

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, deleteRow, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, insertRow, mergeAttributes, moveRow, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, layoutGrid, layoutGridMode, letterSpacing, lineHeight, maxHeight, maxWidth, minHeight, minWidth, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textDecoration, text-decorationBlink, text-decorationLineThrough, text-decorationNone, text-decorationOverline, text-decorationUnderline, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordSpacing, zIndex, zoom

tHead

Retrieves the tHead object of the table.

Syntax

HTML

N/A

Scripting

[*oTHead* =] *table*.tHead

Applies To

table

this

The this keyword refers to the context object (a.k.a. current object). In general, in a method, this refers to the calling object. This is read-only.

Syntax

this[*.propertyName*]

There are 4 ways this can be passed:

method call	object.method(arg1 , arg2)	object
-------------	----------------------------	--------

Function's call method	func.call(object, arg1 , arg2)	object
Function's apply method	func.apply(object, [arg1 , arg2])	object
event handling	an event	the event target that the event handler is a property of

If none of the above ways are used, the global object is passed as the context object.

<title>

Contains the title of the document.

Ex

```
<html>
<head>
<title>XHTML Tag Reference</title>
</head>
<body>
The content of the document.....
</body>
</html>
```

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Methods

addBehavior, applyElement, attachEvent, clearAttributes, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, getAdjacentText, getAttribute, getAttributeNode, getElementsByTagName, hasChildNodes, insertAdjacentElement, mergeAttributes, normalize, removeAttribute, removeAttributeNode, removeBehavior, replaceAdjacentText, setAttribute, setAttributeNode, swapNode

Style Properties and attributes

backgroundPositionX, backgroundPositionY, behavior, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textUnderlinePosition

title

(a, abbr, acronym, ...)

Sets or retrieves advisory information (a ToolTip) for the object.

Syntax

HTML

<ELEMENT title = *sTitle*... >

Scripting

object.title [= *sTitle*]

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, map, marquee, menu, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

title

(styleSheet)

Sets or retrieves the title of the style sheet.

Syntax

styleSheet.title [= *sTitle*]

Applies To

styleSheet

toElement [ie]

Sets or retrieves a reference to the object toward which the user is moving the mouse pointer.

Syntax

event.toElement [= *oObject*]

Applies To

event

toLowerCase() [JavaScript]

The toLowerCase() method is used to display a string in lowercase letters.

Syntax

stringObject.toLowerCase()

Ex

```
<script type="text/javascript">
var str="Hello World!"
document.write(str.toLowerCase())
</script>
```

Applies to

String

toGMTString() [JavaScript]

The toGMTString() method converts a Date object, according to Greenwich Mean Time (GMT), to a string and returns the result.

Syntax

dateObject.toGMTString()

Ex

```
<script type="text/javascript">
var d = new Date()
document.write (d.toGMTString())
</script>
```

The output of the code above will be:

Wed, 28 Jun 2006 17:21:00 GMT

Applies to

Date

toLocaleString() [JavaScript]

The toLocaleString() method converts a Date object, according to local time, to a string and returns the result.

Syntax

dateObject.toLocaleString()

Ex

In the example below we will convert today's date (according to local time) to a string:

```
<script type="text/javascript">
var d = new Date()
document.write(d.toLocaleString())
</script>
```

The output of the code above will be:

quarta-feira, 28 de junho de 2006 14:23:33

Applies to

Date

top

(css)

Sets or retrieves the position of the object relative to the top of the next positioned object in the document hierarchy.

Syntax

CSS

{ top : *sTop* }

Scripting

object.style.top [= *sTop*]

sTop values

auto	Default, position according to the regular HTML layout of the page
<i>length</i>	Floating-point number, followed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px)
%	Integer, followed by a percent sign (%). This value is a percentage of the height of the parent object

Applies To

a, address, applet, b, big, blockquote, button, center, cite, code, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, object, ol, p, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, textarea, tt, u, ul, var, xmp

top

(textRectangle)

Sets or retrieves the top coordinate of the rectangle surrounding the object content.

Syntax

TextRectangle.top [= *iCoord*]

Applies To

TextRectangle

top

(window)

Retrieves the topmost ancestor window.

Syntax

[*oTop* =] *window*.top

Applies To

window

topMargin [ie]

Sets or retrieves the margin for the top of the page, in px.

Syntax

HTML

<body TOPMARGIN = *iMargin*... >

Scripting

body.topMargin [= *iMargin*]

Applies To

body

toSource() [JavaScript]

The toSource() method represents the source code of an object.

Does not work in Internet Explorer.

Syntax

object.toSource()

Ex

```
<script type="text/javascript">
function employee(name, jobtitle, born)
{
  this.name=name
  this.jobtitle=jobtitle
  this.born=born
}
var fred=new employee("Fred Flintstone", "Caveman", 1970)
document.write(fred.toSource())
</script>
```

The output of the code above will be:

(({name:"Fred Flintstone", jobtitle:"Caveman", born:1970})

Applies to

Array, Boolean, Date, Math, String

<tr>

Specifies a row in a table.

Ex

```
<table border = "1">
<tr>
<td>Cell A</td>
<td>Cell B</td>
</tr>
</table>
```

Optional Attributes

align, bgcolor, char, charoff, valign

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, time, time2

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, deleteCell, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode,

getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, insertCell, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, clear, clip, color, cursor, direction, display, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, height, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, letterSpacing, lineBreak, lineHeight, maxHeight, maxWidth, minHeight, minWidth, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, textAlign, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textTransform, textUnderlinePosition, unicodeBidi, verticalAlign, visibility, wordBreak, wordSpacing, zIndex, zoom

throw

The throw statement throws a user-defined exception.

Syntax

throw *expression*;

Ex

You can use throw to rethrow an exception after you catch it. The following example catches an exception with a numeric value and rethrows it if the value is over 50. The rethrown exception propagates up to the enclosing function or to the top level so that the user sees it.

```
try {
    throw n; // throws an exception with a numeric value
} catch (e) {
    if (e <= 50) {
        // statements to handle exceptions 1-50
    } else {
        // cannot handle this exception, so rethrow
        throw e;
    }
}
```

try ... catch

The try statement consists of a try block, which contains one or more statements, and at least one catch clause or a finally clause, or both. That is, there are three forms of the try statement:

- 1. try...catch
- 2. try...finally
- 3. try...catch...finally

Syntax

```
try {
    try_statements }
[catch (exception_var_1 if condition_1) {
    catch_statements_1 }]
...
[catch (exception_var_2) {
    catch_statements_2 }]
[finally {
    finally_statements }]
```

Parameters

<i>try_statements</i>	The statements to be executed.
<i>catch_statements_1</i> , <i>catch_statements_2</i>	Statements that are executed if an exception is thrown in the try block.
<i>exception_var_1</i> , <i>exception_var_2</i>	An identifier to hold an exception object for the associated catch clause.
<i>condition_1</i>	A conditional expression.
<i>finally_statements</i>	Statements that are executed before the try statement completes. These statements execute regardless of whether or not an exception was thrown or caught.

Ex

```
try {
    myroutine(); // may throw three exceptions
} catch (e if e instanceof TypeError) {
    // statements to handle TypeError exceptions
} catch (e if e instanceof RangeError) {
    // statements to handle RangeError exceptions
} catch (e if e instanceof EvalError) {
    // statements to handle EvalError exceptions
} catch (e) {
    // statements to handle any unspecified exceptions
    logMyErrors(e); // pass exception object to error handler }
```

toString() [JavaScript]

The toString() method converts an array, boolean or date object to a string and returns the result, separated by commas.

Syntax

```
arrayObject.toString()
booleanObject.toString()
dateObject.toString()
```

Ex

```
<script type="text/javascript">
var arr = new Array(3)
arr[0] = "Jani"
arr[1] = "Hege"
arr[2] = "Stale"
document.write(arr.toString())
</script>
The output of the code above will be:
Jani, Hege, Stale
```

Applies to

Array, Boolean, Date

<tt>

Renders text in a fixed-width font.

Ex

```
<tt>Teletype text</tt>
```

Standard Attributes

id, class, title, style, dir, lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgrounPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

toUTCString() [JavaScript]

The toUTCString() method converts a Date object, according to Universal Coordinated Time (UTC), to a string and returns the result.

Syntax

dateObject.toUTCString()

Ex

```
<script type="text/javascript">
var d = new Date()
document.write (d.toUTCString())
</script>
```

The output of the code above will be:
Wed, 28 Jun 2006 17:22:38 GMT

Applies to
Date

trueSpeed [ie]

Sets or retrieves whether the position of the marquee is calculated using the scrollDelay and scrollAmount properties and the actual time elapsed from the last clock tick.

Syntax

HTML

<marquee TRUESPEED ... >

Scripting

marquee.trueSpeed [= bSpeed]

bSpeed values

false	Default. Marquee computes movement based on 60-millisecond ticks of the clock. This means every scrollDelay value under 60 is ignored, and the marquee advances the amount of scrollAmount each 60 milliseconds. For example, if scrollDelay is 6 and scrollAmount is 10, the marquee advances 10 pixels every 60 milliseconds
true	Marquee advances the pixel value of scrollAmount by the number of milliseconds set for scrollDelay. For example, the marquee would advance 10 pixels for every 6 milliseconds if scrollDelay is 6, scrollAmount is 10, and the value of trueSpeed is true

Applies To
marquee

type

(a, link, object)

Sets or retrieves the MIME type of the object.

Syntax

HTML

<ELEMENT TYPE = sType... >

Scripting

object.type [= sType]

Applies To

a, object, link

type

(button)

Retrieves the classification and default behavior of the button.

Syntax

HTML

<button TYPE = sType... >

Scripting

[sType =] button.type

sType values

button	Default. Creates a Command button
reset	Creates a Reset button. If it's in a form, this button resets the fields in the form to their initial values
submit	Creates a Submit button. If it's in a form, this button submits the form

Applies To
button

type

(event)

Sets or retrieves the event name from the event object.

Syntax

event.type [= sType]

Applies To
event

type

(input)

Retrieves or initially sets the type of input control represented by the object.

Syntax

<ELEMENT TYPE = sType... >

Scripting

object.type [= sType]

sType values

button	Creates a button control.
checkbox	Creates a check box for simple Boolean attributes or for attributes that can take multiple values at the same time. The input consists of a number of check box controls, each of which has the same name. Each selected check box generates a separate name/value pair in the submitted data, even if this results in duplicate names. The default value is on.
file	Creates a file upload object.
hidden	Creates a control hidden from the user, but the value of the value property is sent with the submitted form.
image	Creates an image control that can be clicked, causing the form to be immediately submitted. The coordinates of the selected point are measured in pixels from the upper-left corner of the image and are submitted with the form as two name/value pairs. The x-coordinate is submitted under the name of the control with .x appended, and the y-coordinate is submitted under the name of the control with .y appended. Any value property is ignored. The image is specified by the src property, exactly as for the img object
password	Creates a control similar to the text control, except that text is not displayed as the user enters it.
radio	Creates radio buttons used for mutually exclusive sets of values. Each radio button control in the group should be given the same name. Only the selected radio button in the group generates a name/value pair in the submitted data. Radio buttons require an explicit value property.
reset	Creates a button that resets the form's controls to their specified initial values. You can specify the label to display on this button just as you can for the Submit button.
submit	Creates a button that submits the form. Use the value attribute to specify a label that is displayed on the button. The default label is application-specific. When the button is clicked, the associated control objects submit name/value pairs to the server.
text	Default. Creates a single-line text-entry control. Use the value in conjunction with the size and maxLength properties

Applies To

input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text

type

(li, ol, ul)

Sets or retrieves the style of the list.

Syntax

HTML

<ELEMENT TYPE = sType... >

Scripting

object.type [= sType]

sType values

1	Default. Associate numbers with each item in an ordered list.
a	Associate lowercase letters with each item in an ordered list.
A	Associate uppercase letters with each item in an ordered list.
i	Associate small Roman numerals with each item in an ordered list.
I	Associate large Roman numerals with each item in an ordered list.
disc	Associate a solid disc with each item in an unordered list.
circle	Associate a hollow circle with each item in an unordered list.
square	Associate a solid square with each item in an unordered list

Applies To

li, ol, UL

type

(param)

Sets or retrieves the content type of the resource designated by the value attribute.

Syntax

<param TYPE = *p*... >

Scripting

param.type(*v*) [= *p*]

value

p	String that specifies or receives the content type of the resource designated by the value attribute only in the case where valueType is set to ref. This attribute thus specifies for the user agent the type of values that will be found at the Uniform Resource Identifier (URI) designated by value
---	--

Applies To

param

type

(script)

Sets or retrieves the MIME type for the associated scripting engine.

Syntax

HTML

<script TYPE = *sType*... >

Scripting

script.type [= *sType*]

sType values

| | |
|-----------------|-----------------------------|
| text/ecmascript | ECMAScript |
| text/Jscript | Microsoft JScript |
| text/javascript | JScript |
| text/vbs | VBScript |
| text/vbscript | VBScript (same as text/vbs) |
| text/xml | XML |

Applies To

script

type

(select)

Retrieves the type of select control based on the value of the MULTIPLE attribute.

Syntax

[*sType* =] *select.type*

sType values

| | |
|-----------------|---|
| select-multiple | The MULTIPLE attribute is true. Multiple option elements can be selected |
| select-one | Default. The MULTIPLE attribute is false Only one option element can be selected. |

Applies To

select

type

(selection)

Retrieves the type of selection.

Syntax

[*sType* =] *selection.type*

sType values

| | |
|---------|--|
| none | No selection/insertion point. |
| text | Specifies a text selection. |
| control | Specifies a control selection, which enables dimension controls allowing the selected object to be resized |

Applies To

selection

type

(style)

Retrieves the Cascading Style Sheets (CSS) language in which the style sheet is written.

Syntax

HTML

<STYLE TYPE = *sType*... >

Scripting

STYLE.type [= *sType*]

Applies To

STYLE

type

(styleSheet)

Retrieves the CSS language in which the style sheet is written.

Syntax

HTML

[*sType* =] *styleSheet.type*

Applies To

styleSheet

type

Retrieves the type of control.

Syntax

[*sType* =] *textarea.type*

Applies To

textarea

typeDetail [ie]

Retrieves the name of the selection type.

Syntax

[*sType* =] *selection.typeDetail*

Applies To

selection

typeof

The typeof operator returns a string indicating the type of the unevaluated operand. operand is the string, variable, keyword, or object for which the type is to be returned. The parentheses are optional.

Syntax

typeof operand

typeof (operand)

Ex

Suppose you define the following variables:

var myFun = new Function("5+2")

var shape="round"

var size=1

var today=new Date()

The typeof operator returns the following results for these variables:

typeof myFun is object

typeof shape is string

typeof size is number

typeof today is object

typeof dontExist is undefined

For the keywords true and null, the typeof operator returns the following results:

typeof true is boolean

typeof null is object

For methods and functions, the typeof operator returns results as follows:

typeof blur is function

typeof eval is function

For predefined objects, the typeof operator returns results as follows:

typeof Date is function

typeof Function is function

<u>

Renders text that is underlined.

Ex

A new version is <u>now available!</u>

Standard Attributes

id, class, title, style, dir, lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, cloneNode, componentFromPoint, detachEvent, fireEvent, focus, getAdjacentText, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertBefore, mergeAttributes, normalize, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, setActive, setAttributeNode, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, backgroundPositionX, backgroundPositionY, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, direction, display, filter, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, posLeft, posRight, posTop, posWidth, textAutospace, textOverflow, textUnderlinePosition, unicodeBidi, whiteSpace, width, wordWrap, writingMode, zoom

Draws lines of text as a bulleted list.

Ex

Coffee
Tea

Optional Attributes

compact, type (disc, circle, square)

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, dragDrop, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop,

borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridChar, layoutGridLine, layoutGridMode, layoutGridType, left, letterSpacing, lineBreak, lineHeight, listStyle, listStyleImage, listStylePosition, listStyleType, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pageBreakAfter, pageBreakBefore, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAlign, textAlignLast, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textIndent, textJustify, textKashidaSpace, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordBreak, wordSpacing, wordWrap, writingMode, zIndex, zoom

undefined [JavaScript]

The undefined property indicates that a variable has not been assigned a value.

Syntax

undefined

Ex

<script type="text/javascript">
var t1=""
var t2
if (t1==undefined) {document.write("t1 is undefined")}
if (t2==undefined) {document.write("t2 is undefined")}
</script>

The output of the code above will be:

t2 is undefined

Applies To

JavaScript Functions

unescape() [JavaScript]

The unescape() function decodes a string encoded with escape().

Syntax

unescape(string)

Ex

<script type="text/javascript">
var test1="Visit W3Schools!"
test1=escape(test1)
document.write (test1 + "
")
test1=unescape(test1)
document.write(test1 + "
")
</script>

The output of the code above will be:

Visit%20W3Schools%21

Visit W3Schools!

Applies to

all of the built-in JavaScript objects

unicodeBidi

Sets or retrieves the level of embedding with respect to the bidirectional algorithm.

Syntax

CSS

{ unicode-bidi : *sEmbedLevel* }

Scripting

object.style.unicodeBidi [= *sEmbedLevel*]

sEmbedLevel values

normal	Default. Element does not open an additional level of embedding. For inline elements, implicit reordering works across element boundaries.
embed	Element opens an additional level of embedding. The value of the direction property specifies the embedding level. Reordering is implicit inside the element.
bidi-override	Same as the embed value, except that, inside the element, reordering is strictly in sequence according to the direction property. Overrides the implicit bidirectional algorithm.

Applies To

a, abbr, acronym, address, b, bdo, big, blockquote, body, button, caption, center,



cite, code, col, colgroup, currentStyle, custom, dd, defaults, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, hn, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

uniqueID [ie]

Retrieves an autogenerated, unique identifier for the object.

Syntax

HTML

N/A

Scripting

[*sID* =] *object*.uniqueID

Applies To

a, abbr, acronym, address, applet, area, b, base, basefont, bgsound, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, comment, dd, del, dfn, dir, div, dl, document, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, kbd, label, legend, li, link, listing, map, marquee, menu, nobr, object, ol, option, p, plaintext, pre, q, s, samp, script, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var, xmp

units [ie]

Sets or retrieves the height and width units of the embed object.

Syntax

HTML

<EMBED UNITS = *sUnits*... >

Scripting

EMBED.units [= *sUnits*]

sUnits values

px	Height and width are in pixels
em	Height and width are relative to the height and width of the element's font

Applies To

EMBED

UNSELECTABLE [ie]

Specifies that an element cannot be selected.

Syntax

HTML

<ELEMENT UNSELECTABLE=*bUnselectable* ... >

Scripting

N/A

bUnselectable values

off	Default. Permits selection of the element
on	Prohibits selection of the element

Applies To

a, abbr, acronym, address, applet, area, b, bdo, big, blockquote, body, button, caption, center, cite, code, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, hn, hr, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, li, listing, marquee, menu, nextID, nobr, object, ol, p, plaintext, pre, q, rt, ruby, s, samp, select, small, span, strike, strong, sub, sup, table, tbody, td, textarea, tfoot, thead, tt, u, ul, var, xmp

unshift() [JavaScript]

The unshift() method adds one or more elements to the beginning of an array and returns the new length. Does not work properly in Internet Explorer.

Syntax

arrayObject.unshift(newelem1 , newelem2,, newelemX)

Parameters

<i>newelem1</i>	Required. The first element to add to the array
<i>newelem2</i>	Optional. The second element to add to the array
<i>newelemX</i>	Optional. Several elements may be added

Ex

In this example we will create an array, add an element to the beginning of the array and then return the new length:

```
<script type="text/javascript">
var arr = new Array()
arr[0] = "Jani"
arr[1] = "Hege"
arr[2] = "Stale"
document.write(arr + "<br />")
document.write(arr.unshift("Kai Jim") + "<br />")
document.write(arr)
</script>
```

The output of the code above will be:

Jani, Hege, Stale
4
Kai Jim, Jani, Hege, Stale

Applies to

Array

updateInterval [ie]

Sets or retrieves the update interval for the screen, in milliseconds

Syntax

HTML

screen.updateInterval(*v*) [= *interval*]

Applies To

screen

URL

Sets or retrieves the URL for the current document.

Syntax

document.URL [= *sURL*]

Applies To

document

URLUnencoded [ie]

Retrieves the URL for the document, stripped of any character encoding.

Syntax

[*sUrl* =] *document*.URLUnencoded

Applies To

document

urn [ie]

Sets or retrieves a URN for a target document.

Syntax

HTML

<*ELEMENT* URN = *sURN*... >

Scripting

object.urn [= *sURN*]

Applies To

a, namespace

urns() [ie]

Retrieves a collection of all objects to which a specified behavior is attached.

Syntax

collObjects = object.urns(sURN)

Parameter

<i>sUrn</i>	Required. String that specifies the behavior's Uniform Resource Name (URN).
-------------	---

Applies to

all, anchors, applets, areas, boundElements, cells, childNodes, children, elements, embeds, form, forms, images, links, options, rows, scripts, select, styleSheets, tBodies

useMap

Sets or retrieves the URL, often with a bookmark extension (#name), to use as a client-side image map.

Syntax

HTML

<ELEMENT USEMAP = *sURL*... >

Scripting

object.useMap [= *sURL*]

Applies To

img, object, input

userAgent [ie]

Retrieves a string equivalent to the HTTP user-agent request header.

Syntax

[*sUserAgent* =] *object.userAgent*

Applies To

clientInformation, navigator

userLanguage [ie]

Retrieves the operating system's natural language setting.

Syntax

[*sLanguage* =] *object.userLanguage*

Applies To

clientInformation, navigator

userProfile [ie]

Provides methods that allow a script to request read access to and perform read actions on a user's profile information.

Ex

This script example runs on the client and uses the userProfile object to read various values from the profile information:

```
// Queue a request for read access to multiple profile attributes.
navigator.userProfile.addReadRequest("vcard.displayName");
navigator.userProfile.addReadRequest("vcard.gender");
// Request access to the information.
navigator.userProfile.doReadRequest(usage-code, "Acme Corporation");
// Now perform read operations to access the information.
name = navigator.userProfile.getAttribute("vcard.displayName");
gender = navigator.userProfile.getAttribute("vcard.gender");
// The script can now use the 'name' and 'gender' variables
// to personalize content or to send information back to the server.
// Clear the request queue to prepare for later information requests.
navigator.userProfile.clearRequest();
```

Methods

addReadRequest, clearRequest, doReadRequest, getAttribute, setAttribute

UTC() [JavaScript]

The UTC() method takes a date and returns the number of milliseconds since midnight of January 1, 1970 according to universal time.

Syntax

Date.UTC(*year*, *month*, *day*, *hour*, *min*, *sec*, *ms*)

Parameter

<i>year</i>	Required. A four digit number representing the year
<i>month</i>	Required. An integer between 0 and 11 representing the month
<i>day</i>	Required. An integer between 1 and 31 representing the date
<i>hour</i>	Optional. An integer between 0 and 23 representing the hour
<i>min</i>	Optional. An integer between 0 and 59 representing the minutes
<i>sec</i>	Optional. An integer between 0 and 59 representing the seconds
<i>ms</i>	Optional. An integer between 0 and 999 representing the milliseconds

Ex

In this example we will get how many milliseconds there are from 1970/01/01 to 2005/07/08 according to universal time:

```
<script type="text/javascript">
var d = Date.UTC(2005, 7, 8)
document.write(d)
</script>
```

The output of the code above will be:

1123459200000

Applies to

Date

CAPÍTULO 1 A MERCADORIA

1. Os dois fatores da mercadoria: Valor de uso e valor (substância do valor, grandeza do valor)

A riqueza das sociedades em que domina o modo de produção capitalista aparece como uma “imensa coleção de mercadorias”

e a mercadoria individual como sua forma elementar. Nossa investigação começa, portanto, com a análise da mercadoria. A mercadoria é, antes de tudo, um objeto externo, uma coisa, a qual pelas suas propriedades satisfaz necessidades humanas de qualquer espécie. A natureza dessas necessidades, se elas se originam do estômago ou da fantasia, não altera nada na coisa. Aqui também não se trata de como a coisa satisfaz a necessidade humana, se imediatamente, como meio de subsistência, isto é, objeto de consumo, ou se indiretamente, como meio de produção.

Cada coisa útil, como ferro, papel etc., deve ser encarada sob duplo ponto de vista, segundo qualidade e quantidade. Cada uma dessas coisas é um todo de muitas propriedades e pode, portanto, ser útil, sob diversos aspectos. Descobrir esses diversos aspectos e, portanto, os múltiplos modos de usar as coisas é um ato histórico. Assim como também o é a descoberta de medidas sociais para a quantidade das coisas úteis.

A diversidade das medidas de mercadorias origina-se em parte da natureza diversa dos objetos a serem medidos, em parte de convenção.

A utilidade de uma coisa faz dela um valor de uso. Essa utilidade, porém, não paira no ar. Determinada pelas propriedades do corpo da mercadoria, ela não existe sem o mesmo.

(...)

Os valores de uso constituem o conteúdo material da riqueza, qualquer que seja a forma social desta. Na forma de sociedade a ser por nós examinada, eles constituem, ao mesmo tempo, os portadores materiais do — valor de troca.

(...)

Deixando de lado então o valor de uso dos corpos das mercadorias, resta a elas apenas uma propriedade, que é a de serem produtos do trabalho. Entretanto, o produto do trabalho também já se transformou em nossas mãos. Se abstraímos o seu valor de uso, abstraímos também os componentes e formas corpóreas que fazem dele valor de uso. Deixa já de ser mesa ou casa ou fio ou qualquer outra coisa útil. Todas as suas qualidades sensoriais se apagaram. Também já não é o produto do trabalho do marceneiro ou do pedreiro ou do fiandeiro ou de qualquer outro trabalho produtivo determinado. Ao desaparecer o caráter útil dos produtos do trabalho, desaparece o caráter útil dos trabalhos neles representados, e desaparecem também, portanto, as diferentes formas concretas desses trabalhos, que deixam de diferenciar-se um do outro para reduzir-se em sua totalidade a igual trabalho humano, a trabalho humano abstrato.

(...) nenhuma coisa pode ser valor, sem ser objeto de uso.

Sendo inútil, do mesmo modo é inútil o trabalho nela contido, não conta como trabalho e não constitui qualquer valor.

Karl Marx — O Capital

vAlign

(caption)

Sets or retrieves whether the caption appears at the top or bottom of the table.

Syntax

HTML

<caption VALIGN = *sAlign*... >

Scripting

caption.vAlign [= *sAlign*]

sAlign values

top	Default. Places the caption at the top of the table
bottom	Places the caption at the bottom of the table

Applies To

caption

vAlign

(col, colgroup, ...)

Sets or retrieves how text and other content are vertically aligned within the object that contains them.

Syntax

HTML

<ELEMENT VALIGN = *sAlign*... >

Scripting

object.vAlign [= *sAlign*]

sAlign values

middle	Default. Aligns the text in the middle of the object.
baseline	Aligns the base line of the first line of text with the base lines in adjacent objects.
bottom	Aligns the text at the bottom of the object.
top	Aligns the text at the top of the object

Applies To

col, colgroup, tbody, td, tfoot, th, thead, tr

value

(attribute)

Sets or retrieves the value of the object.

Syntax

attribute.value(v) [= *sValue*]

Applies To

attribute

value

(button)

Sets or retrieves the default or selected value of the control.

Syntax

HTML

<ELEMENT VALUE = *sValue*... >

Scripting

object.value [= *sValue*]

Applies To

button, input, input type=checkbox, input type=hidden, input type=button, input type=radio, input type=reset, input type=submit

value

(input type=file)

Retrieves the file name of the input object after the text is set by user input.

Syntax

[*sValue* =] *input type=file.value*

Applies To

input type=file

value

(input type=password, input type=text)

Sets or retrieves the displayed value for the control object. This value is returned to the server when the control object is submitted.

Syntax

HTML

<ELEMENT VALUE = *sValue*... >

Scripting

object.value [= *sValue*]

Applies To

input type=password, input type=text

value

(li)

Sets or retrieves the value of a list item.

Syntax

HTML

<li VALUE = *sValue*... >

Scripting

li.value [= *sValue*]

Applies To

li

value

(option, select)

Sets or retrieves the value which is returned to the server when the form control is submitted.

Syntax

HTML

<ELEMENT VALUE = *sValue*... >

Scripting

object.value [= *sValue*]

Applies To

option, select

value

(param)

Sets or retrieves the value of an input parameter for an element.

Syntax

HTML

<param VALUE = *p*... >

Scripting

param.value(v) [= *p*]

Applies To

param

value

(textArea)

Retrieves or sets the text in the entry field of the textArea element.

Syntax

textarea.value [= *sValue*]

Applies To

textarea

valueOf() [JavaScript]

The valueOf() method returns the primitive value of an Array, Boolean, Date, Math or String object.

The primitive value is inherited by all objects descended from the object. The valueOf() method is usually called automatically by JavaScript behind the scenes and not explicitly in code.

Syntax

Object.valueOf()

Ex

```
<script type="text/javascript">
var boo = new Boolean(false)
document.write(boo.valueOf())
</script>
```

The output of the code above will be:
false

Applies to

Array, Boolean, Date, Math, String

valueType [ie]

Sets or retrieves the data type of the value attribute.

Syntax

HTML

<param VALUETYPE = *p*... >

Scripting

*param.valueType(v) [= *p*]*

p values

data	Default. The value specified by the value attribute will be evaluated and passed to the applet, embed, or object element as a string.
ref	The value specified by the value attribute is a Uniform Resource Identifier (URI) that designates a resource where run-time values are stored. This allows support tools to identify URLs given as parameters.
object	The value specified by the value attribute is an identifier (ID) that refers to an object declaration in the same document. The ID must be the value of the id attribute set for the declared object element.

Applies To

param

<var>

Defines a programming variable. Typically renders in an italic font style.

Ex

<var>Keyboard text</var>

Standard Attributes

id, class, title, style, dir, lang, xml:lang

Behaviors

anchorClick, clientCaps, download, homePage, httpFolder, saveFavorite, saveHistory, saveSnapshot, time, time2, userData

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, appendChild, applyElement, attachEvent, blur, clearAttributes, click, cloneNode, componentFromPoint, contains, detachEvent, fireEvent, focus, getAdjacentText, getAttribute, getAttributeNode, getBoundingClientRect, getClientRects, getElementsByTagName, getExpression, hasChildNodes, insertAdjacentElement, insertAdjacentHTML, insertAdjacentText, insertBefore, mergeAttributes, normalize, releaseCapture, removeAttribute, removeAttributeNode, removeBehavior, removeChild, removeExpression, removeNode, replaceAdjacentText, replaceChild, replaceNode, scrollIntoView, setActive, setAttribute, setAttributeNode, setCapture, setExpression, swapNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

:first-child, :hover, accelerator, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundPositionX, backgroundPositionY, backgroundRepeat, behavior, border, borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth, borderColor, borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth, borderRight, borderRightColor, borderRightStyle, borderRightWidth, borderStyle, borderTop, borderTopColor, borderTopStyle, borderTopWidth, borderWidth, bottom, clear, clip, color, cursor, direction, display, filter, font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight, fontWeight, hasLayout, height, layoutFlow, layoutGrid, layoutGridMode, left, letterSpacing, lineHeight, margin, marginBottom, marginLeft, marginRight, marginTop, maxHeight, maxWidth, minHeight, minWidth, overflow, overflowX, overflowY, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, pixelBottom, pixelHeight, pixelLeft, pixelRight, pixelTop, pixelWidth, posBottom, posHeight, position, posLeft, posRight, posTop, posWidth, right, styleFloat, textAutospace, textDecoration, textDecorationBlink, textDecorationLineThrough, textDecorationNone, textDecorationOverline, textDecorationUnderline, textOverflow, textTransform, textUnderlinePosition, top, unicodeBidi, visibility, whiteSpace, width, wordSpacing, wordWrap, writingMode, zIndex, zoom

var

Declares a variable, optionally initializing it to a value. The scope of a variable is the current function or, for variables declared outside a function, the current application.

Syntax

*var varname1 [= *value1*], varname2 [= *value2*], ..., varnameN [= *valueN*];*

Parameters

<i>varnameN</i>	Variable name. It can be any legal identifier.
<i>valueN</i>	Initial value of the variable. It can be any legal expression.

Ex

The following example declares two variables, num_hits and cust_no, and initializes both to the value 0.

var num_hits = 0, cust_no = 0;

vcard_name [ie]

Sets or retrieves the vCard value of the object to use for the AutoComplete box.

Syntax

HTML

<ELEMENT VCARD_NAME = *sVCard*... >

Scripting

*object.vcard_name [= *sVCard*]*

sVCard values

vCard.Business.City, vCard.Business.Country, vCard.Business.Fax, vCard.Business.Phone, vCard.Business.State, vCard.Business.StreetAddress, vCard.Business.URL, vCard.Business.Zipcode, vCard.Cellular, vCard.Company, vCard.Department, vCard.DisplayName, vCard.Email, vCard.FirstName, vCard.Gender, vCard.Home.City, vCard.Home.Country, vCard.Home.Fax, vCard.Home.Phone, vCard.Home.State, vCard.Home.StreetAddress, vCard.Home.Zipcode, vCard.Homepage, vCard.JobTitle, vCard.LastName, vCard.MiddleName, vCard.Notes, vCard.Office, vCard.Pager

Applies To

input type=password, input type=text

version

Sets or retrieves the Document Type Definition (DTD) version that governs the current document.

Syntax

HTML

<HTML VERSION = *sVersion*... >

Scripting

*HTML.version(v) [= *sVersion*]*

Applies To

HTML

verticalAlign

Sets or retrieves the vertical alignment of the object.

Syntax

CSS

{ vertical-align : *sAlign* }

Scripting

*object.style.verticalAlign [= *sAlign*]*

sAlign values

auto	Aligns the contents of an object according to the value of the layout-flow attribute.
baseline	Default. Aligns the contents of an object supporting VALIGN to the base line.
sub	Vertically aligns the text to subscript.
super	Vertically aligns the text to superscript.
top	Vertically aligns the contents of an object supporting VALIGN to the top of the object.
middle	Vertically aligns the contents of an object supporting VALIGN to the middle of the object.
bottom	Vertically aligns the contents of an object supporting VALIGN to the bottom of the object.
text-top	Vertically aligns the text of an object supporting VALIGN to the top of the object.
text-bottom	Vertically aligns the text of an object supporting VALIGN to the bottom of the object.

Applies To

col, currentStyle, custom, defaults, img, runtimeStyle, span, style, tbody, td, tfoot, th, thead, tr

viewInheritStyle [ie]

Sets or retrieves a value that indicates whether the document fragment inherits the CSS styles set in the primary document.

Syntax

defaults.viewInheritStyle(v) [= *blnherit*]

blnherit values

false	Document fragment does not inherit CSS styles
true	Default. Document fragment inherits CSS styles

Applies To

defaults

viewLink [ie]

Sets or retrieves the document object that supplies content to the master element.

Syntax

[*oDocument* =] *defaults.viewLink(v)*

Applies To

defaults

viewMasterTab [ie]

Sets or retrieves a value that indicates whether the master element of a viewlink is included in the tab sequence of the primary document.

Syntax

defaults.viewMasterTab(v) [= *blnherit*]

blnherit values

false	Tabbing from an element in the primary document to a viewlink sets focus to the next tab stop, bypassing the master element
true	Default. Tabbing from an element in the primary document to a viewlink sets focus to the master element

Applies To

defaults

visibility

Sets or retrieves whether the content of the object is displayed.

Syntax

CSS

{ *visibility* : *sVisibility* }

Scripting

object.style.visibility [= *sVisibility*]

sVisibility values

inherit	Default. Object inherits the visibility of the next parent object.
visible	Object is visible.
hidden	Object is hidden

Applies To

a, address, applet, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, embed, fieldset, form, hn, hr, html, i, iframe, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, nobr, object, ol, p, pre, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

vLink

Sets or retrieves the color of links in the object that have already been visited. Deprecated.

Syntax

HTML

<body VLINK = *sColor*... >

Scripting

body.vLink [= *sColor*]

Applies To

body

vlinkColor [ie]

Sets or retrieves the color of the links that the user has visited. The default value is #800080

Syntax

document.vlinkColor [= *sColor*]

Applies To

document

void

The void operator specifies an expression to be evaluated without returning a value. expression is a JavaScript expression to evaluate. The parentheses surrounding the expression are optional, but it is good style to use them.

You can use the void operator to specify an expression as a hypertext link. The expression is evaluated but is not loaded in place of the current document.

The following code creates a hypertext link that does nothing when the user clicks it. When the user clicks the link, void(0) evaluates to 0, but that has no effect in JavaScript.

Syntax

void (expression)

void expression

Ex

The following code creates a hypertext link that submits a form when the user clicks it.

Click here to submit

volume [ie]

Sets or retrieves the volume setting for the sound. From -10,000 (min) to 0 (full wave output)

Syntax

HTML

<bgsound VOLUME = *iVolume*... >

Scripting

bgsound.volume [= *iVolume*]

Applies To

bgsound

vspace

Sets or retrieves the vertical margin for the object. In px.

Syntax

HTML

<ELEMENT VSPACE = *iMargin*... >

Scripting

object.vspace [= *iMargin*]

Applies To

frame, frameset, layoutrect, table, td, th, tr

<wbr> [ie]

Inserts a soft line break into a block of nobr text. This object is an extension to html, may not work in all browsers.

Ex

```
<noBr>This line of text will not break, no matter how narrow the window gets.</noBr>
<noBr>This one, however, <wbr> will break after the word "however,"
if the window gets small enough.</noBr>
```

Standard Attributes

id

Behaviors

clientCaps, download, homePage

Collection

all, attributes, behaviorUrns, childNodes, children

Events

onFocus, onBlur, onClick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Methods

addBehavior, componentFromPoint, fireEvent, getAttribute, getAttributeNode, normalize, removeAttribute, removeAttributeNode, removeBehavior, scrollIntoView, setAttribute, setAttributeNode

Object

currentStyle

Style Properties and attributes

behavior

wheelDelta [ie]

Retrieves the distance and direction the wheel button has rolled. Measurement values of wheelDelta are in multiples of 120

Syntax

[*p* =] *event*.wheelDelta

Applies To

event

while

This statement creates a loop that executes a specified statement as long as the test condition evaluates to true. The condition is evaluated before executing the statement.

Syntax

while (*condition*)

statement

Parameters

<i>condition</i>	An expression evaluated before each pass through the loop. If this condition evaluates to true, statement is executed. When condition evaluates to false, execution continues with the statement after the while loop.
<i>statement</i>	A statement that is executed as long as the condition evaluates to true. To execute multiple statements within the loop, use a block statement ({ ... }) to group those statements.

Ex

The following while loop iterates as long as n is less than three.

```
n = 0;
x = 0;
while (n < 3) {
    n ++;
    x += n;
}
```

whiteSpace

Sets or retrieves a value that indicates whether lines are automatically broken inside the object.

Syntax

CSS

{ white-space : *sWrap* }

Scripting

object.style.whiteSpace(*v*) [= *sWrap*]

sWrap values

normal	Default. Lines of text break automatically. Content wraps to the next line if it exceeds the width of the object.
nowrap	Line breaks are suppressed. Content does not wrap to the next line.

pre

Line breaks and other whitespace are preserved. This possible value is supported in Microsoft Internet Explorer 6 and later when the !DOCTYPE declaration specifies standards-compliant mode. When the !DOCTYPE declaration does not specify standards-compliant mode, you can retrieve this value, but it does not affect renderingit functions like the normal value.

Applies To

abbr, acronym, address, basefont, blockquote, bdo, big, body, center, cite, code, currentStyle, custom, dd, div, dir, dl, dt, em, fieldset, form, font, hn, hr, isindex, i, ins, kbd, label, legend, li, listing, menu, ol, p, q, plaintext, pre, s, small, span, strike, strong, style, sub, sup, tt, u, ul, xmp

width

(col, colgroup, embed, ...)

Sets or retrieves the width of the object.

Syntax

HTML

<ELEMENT WIDTH = *sWidth*... >

Scripting

object.width [= *sWidth*]

sWidth values

<i>width</i>	Integer that specifies the width of the object in pixels
%	String that specifies an integer value followed by a %. The value is a percentage of the width of the parent object

Applies To

col, colgroup, embed, frame, frameset, iframe, marquee, object, table, td, th, tr

width

(css)

Sets or retrieves the width of the object.

Syntax

CSS

{ width : *sWidth* }

Scripting

object.style.width [= *sWidth*]

sWidth values

auto	Default. Default width of the object
<i>width</i>	Integer that specifies the width of the object in pixels
%	String that specifies an integer value followed by a %. The value is a percentage of the width of the parent object

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, button, caption, center, cite, code, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, hr, i, ins, kbd, label, legend, li, listing, menu, nobr, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, textarea, tt, u, ul, var, xmp

width

(img)

Sets or retrieves the calculated width of the object.

Syntax

HTML

<ELEMENT WIDTH = *iWidth*... >

Scripting

object.width [= *iWidth*]

iWidth values

<i>width</i>	Integer that specifies the width of the object in pixels
%	String that specifies an integer value followed by a %. The value is a percentage of the width of the parent object

Applies To

img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text

width

Sets or retrieves a value that you can use to implement your own width functionality for the object.

Syntax

HTML

<pre WIDTH = *sWidth*... >

Scripting

pre.width(*v*) [= *sWidth*]

Applies To

pre

width

Retrieves the horizontal resolution of the screen.

Syntax

[*iWidth* =] *screen*.width

Applies To

screen

window

Represents an open window in the browser.

Ex

This example checks whether the current window contains child windows and, if it does, displays the names of those child windows.

```
if ( window.frames != null ) {  
    for ( i = 0; i< window.frames.length; i++ )  
        window.alert ( "Child window " +i+ " is named "+window.frames(i).name);  
}
```

Properties

| | |
|---------------------|--|
| closed | Retrieves whether the referenced window is closed. |
| defaultStatus | Sets or retrieves the default message displayed in the status bar at the bottom of the window. |
| dialogArguments | Retrieves the variable or array of variables passed into the modal dialog window. |
| dialogHeight | Sets or retrieves the height of the modal dialog window. |
| dialogLeft | Sets or retrieves the left coordinate of the modal dialog window. |
| dialogTop | Sets or retrieves the top coordinate of the modal dialog window. |
| dialogWidth | Sets or retrieves the width of the modal dialog window. |
| frameElement | Retrieves the frame or iframe object that is hosting the window in the parent document. |
| length | Sets or retrieves the number of objects in a collection. |
| name | Sets or retrieves a value that indicates the window name. |
| offscreenBuf-fering | Sets or retrieves whether objects are drawn offscreen before being made visible to the user. |
| opener | Sets or retrieves a reference to the window that cre-ated the current window. |
| parent | Retrieves the parent of the window in the object hierarchy. |
| returnValue | Sets or retrieves the value returned from the modal dialog window. |
| screenLeft | Retrieves the x-coordinate of the upper left-hand corner of the browser's client area, relative to the upper left-hand corner of the screen. |
| screenTop | Retrieves the y-coordinate of the top corner of the browser's client area, relative to the top corner of the screen. |
| self | Retrieves a reference to the current window or frame. |
| status | Sets or retrieves the message in the status bar at the bottom of the window. |
| top | Retrieves the topmost ancestor window. |
| XMLHttpRequest | Instantiates the XMLHttpRequest object for the window. |

Collection

frames

Events

onactivate, onafterprint, onbeforedeactivate, onbeforeprint, onbeforeunload, onblur, oncontrolselect, ondeactivate, onerror, onfocus, onhelp, onload, onmove, onmoveend, onmovestart, onresize, onresizeend, onscroll, onunload

Methods

alert, attachEvent, blur, clearInterval, clearTimeout, close, confirm, createPopup, detachEvent, execScript, focus, moveBy, moveTo, navigate, open, print, prompt, resizeBy, resizeTo, scroll, scrollBy, scrollTo, setActive, setInterval, setTimeout, showHelp, showModalDialog, showModelessDialog

Objects

clientInformation, clipboardData, document, event, external, history, location, navigator, screen

with

The with statement extends the scope chain for a statement.

Syntax

with (*object*)

statement

Parameterers

| | |
|------------------|--|
| <i>object</i> | Adds the given object to the scope chain used when evaluating the statement. The parentheses around object are required. |
| <i>statement</i> | Any statement. To execute multiple statements, use a block statement ({ ... }) to group those statements. |

Ex

The following with statement specifies that the Math object is the default object. The statements following the with statement refer to the PI property and the cos and sin methods, without specifying an object. JavaScript assumes the Math object for these references.

```
var a, x, y;  
var r = 10;  
with (Math) {  
    a = PI * r * r;  
    x = r * cos(PI);  
    y = r * sin(PI / 2); }
```

wordBreak

Sets or retrieves line-breaking behavior within words, particularly where multiple languages appear in the object.

Syntax

CSS

{ word-break : *sBreak* }

Scripting

object.style.wordBreak [= *sBreak*]

sBreak values

| | |
|-----------|--|
| normal | Default. Allows line breaking within words. |
| break-all | Behaves the same as normal for Asian text, yet allows the line to break arbitrarily for non-Asian text. This value is suited to Asian text that contains some excerpts of non-Asian text. |
| keep-all | Does not allow word breaking for Chinese, Japanese, and Korean. Functions the same way as normal for all non-Asian languages. This value is optimized for text that includes small amounts of Chinese, Japanese, or Korean |

Applies To

address, blockquote, body, center, currentStyle, dd, dir, div, dl, dt, fieldset, form, hn, hr, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, style, table, td, th, tr, ul, xmp

wordSpacing

Sets or retrieves the amount of additional space between words in the object.

Syntax

CSS

{ word-spacing : *sSpacing* }

Scripting

object.style.wordSpacing(*v*) [= *sSpacing*]

sSpacing values

| | |
|--------|---|
| normal | Default. Default spacing |
| length | Floating-pointfollowed by an absolute units designator (cm, mm, in, pt, or pc) or a relative units designator (em, ex, or px) |

Applies To

a, address, b, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, dfn, dir, div, dl, dt, em, fieldset, form, hn, html, i, img, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, select, small, span, strike,

Syntax

a, abbr, acronym, address, b, big, blockquote, button, caption, center, cite, code, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, hn, hr, i, input, input type=button, input type=file, input type=password, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, legend, ll, marquee, menu, ol, option, p, plaintext, pre, q, r, ruby, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, td, textarea, th, tt, u, u, u, VAR



X

Sets or retrieves the x-coordinate, in pixels, of the mouse pointer's position relative to a relatively positioned parent element.

Syntax

event.x [= *iX*]

Applies To

event

<xml> [ie]

Defines an XML data island on an HTML page. This object is a Microsoft extension to html, may not work in all browsers

Ex

```
<XML ID="oMetaData">
  <METADATA>
    <AUTHOR>John Smith</AUTHOR>
    <GENERATOR>Visual Notepad</GENERATOR>
    <PAGETYPE>Reference</PAGETYPE>
    <ABSTRACT>Specifies a data island</ABSTRACT>
  </METADATA>
</XML>
```

Optional Attributes

src

Standard Attributes

id

Behaviors

clientCaps, download, homePage

Collection

behaviorUrns,

Events

ondataavailable, ondatasetchanged, ondatasetcomplete, onreadystatechange, onrowenter, onrowexit, onrowsdelete, onrowsinserted

Methods

addBehavior, componentFromPoint, fireEvent, getAttributeNode, namedRecordset, normalize, removeAttributeNode, removeBehavior, setAttributeNode

Objects

currentStyle, runtimeStyle, style

Style Properties and attributes

behavior, textAutospace, textUnderlinePosition

XMLDocument

Retrieves a reference to the Extensible Markup Language (XML) Document Object Model (DOM) exposed by the object.

Syntax

HTML

N/A

Scripting

[*oXMLObject* =] *object*.XMLDocument

Applies To

document, XML

XMLHttpRequest

Represents a Extensible Markup Language (XML) request via HTTP.

With the XMLHttpRequest object, Internet Explorer clients can retrieve and submit XML data directly to a Web server without reloading the page. To convert XML data into renderable HTML content, use the client-side XML DOM or Extensible Stylesheet Language Transformations (XSLT) to compose HTML elements for presentation.

The native scripting object also supports the use of expandos (custom properties), and properly recognizes the 'this' notation of Javascript.

Ex

The following script demonstrates how to create and use the XMLHttpRequest object.

```
if (window.XMLHttpRequest)
{
    var oReq = new XMLHttpRequest;
    oReq.open("GET", "http://localhost/test.xml");
```

```
oReq.send();
alert(oReq.statusText);}
```

Properties

| | |
|--------------------|---|
| onreadystatechange | Sets the event handler for asynchronous requests. |
| readyState | Retrieves the current state of the request operation. |
| responseBody | Retrieves the response body as an array of unsigned bytes. |
| responseText | Retrieves the response body as a string. |
| responseXML | Retrieves the response body as an XML Document Object Model (DOM) object. |
| status | Retrieves the HTTP status code of the request. |
| statusText | Retrieves the friendly HTTP status of the request. |

Methods

abort, getAllResponseHeaders, getResponseHeader, open, send, setRequestHeader

XMLHttpRequest

Instantiates the XMLHttpRequest object for the window.

Syntax

[*oRequest* =] *window*.XMLHttpRequest

Applies To

window

XMLNS

Declares a namespace for custom tags in an HTML document.

Syntax

HTML

<HTML XMLNS:sNamespace ... >

Applies To

HTML

<xmp>

Renders text used for examples in a fixed-width font. HTML elements within an xmp element render as text, not as HTML formatted elements. Deprecated in html, not supported in XHTML 1.0

XSLDocument

Retrieves a reference to the top-level node of the Extensible Stylesheet Language (XSL) document.

Syntax

[*oXSL* =] *document*.XSLDocument

Applies To

document

y

Sets or retrieves the y-coordinate, in pixels, of the mouse pointer's position relative to a relatively positioned parent element.

Syntax

event.y [= *iY*]

Applies To

event

zIndex

Sets or retrieves the stacking order of positioned objects.

Syntax

CSS

{ *z-index* : *vOrder* }

Scripting

object.style.zIndex [= *vOrder*]

vOrder values

| | |
|--------------|---|
| <i>auto</i> | Default. String that specifies the stacking order of the positioned objects based on the order in which the objects appear in the HTML source |
| <i>order</i> | Integer that specifies the position of the object in the stacking order |

Applies To

a, address, applet, b, big, blockquote, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, defaults, dfn, dir, div, dl, dt, em, fieldset, form, i, iframe, input type=button, input type=checkbox, input type=file, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, isindex, kbd, label, legend, li, listing, marquee, menu, ol, p, plaintext, pre, runtimeStyle, s, samp, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp

zoom [ie]

Sets or retrieves the magnification scale of the object.

Syntax

CSS

{ *zoom* : *vMagnification* }

Scripting

object.style.zoom [= *vMagnification*]

vMagnification values

| | |
|---------------|--|
| <i>normal</i> | Default. Magnification scale is normal. The object renders normal size |
| <i>number</i> | Floating-point number that specifies the magnification scale, where 1.0 is normal |
| <i>%</i> | Integer, followed by a %. The value is a percentage of the magnification scale, where 100% is normal |

Applies To

a, abbr, acronym, address, applet, b, bdo, big, blockquote, body, button, caption, center, cite, code, col, colgroup, currentStyle, custom, dd, del, dfn, dir, div, dl, dt, em, embed, fieldset, font, form, frame, frameset, head, hn, hr, i, iframe, img, input, input type=button, input type=checkbox, input type=file, input type=hidden, input type=image, input type=password, input type=radio, input type=reset, input type=submit, input type=text, ins, isindex, kbd, label, layoutrect, legend, li, listing, marquee, menu, nobr, noframes, noscript, object, ol, option, p, plaintext, pre, q, rt, ruby, runtimeStyle, s, samp, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, tr, tt, u, ul, var, xmp



Bibliografia

ASHBY, W. Ross—Uma Introdução à Cibernética. São Paulo: Perspectiva, 1970

BAKHTIN, Mikhail—Marxismo e Filosofia da linguagem. São Paulo, editora Hucitec, 1981

BARRETT, Daniel J—Essencial Javascript for web professionals. Upper Saddle River, N.J.

BOAG, Andrew—The representation of type on screen, palestra dada na conferência Monotype de 1990, publicado em 1990; <http://www.boag.co.uk/down/index.html>

BRINGHURST, Robert—Elementos do Estilo Tipográfico. São Paulo: Cosacnaify, 2005

CAMPOS, Haroldo; CAMPOS, Augusto, SCHNEIDERMAN, Boris—MAIAKOVSKY, Poemas. São Paulo: Perspectiva 2003

COTTON, Bbob, ED—The new guide to graphic Design; Oxford: Phaidon, 1990

CRAIG, James—Produção Gráfica: para planejador gráfico, editor, diretor de arte, produtor, estudante; São Paulo: Editora Mosaico, 1980

ECO, Umberto—Tratado Geral de Semiótica. São Paulo: Perspectiva, 1976

FLANAGAN, David—Javascript: The Definitive Guide. Sebastopol, CA: O'Reilly, 2002

FRUTIGER, Adrian—Sinais e símbolos: desenho, projeto e significado, São Paulo: Martins fontes 1999

GARDNER, Martin—Mathematical Circus. New York: Vintage Books, 1981

GOODMAN, Danny—Dynamic HTML: The Definitive Reference. Beijing, Sebastopol, CA: O'Reilly, 1998

GRAHAM, Ian S—HTML, a referência completa para HTML. Rio de Janeiro: Campus 1998

GRÄTZER, George—First Steps in LATEX. Boston: Birkhäuser, 1999

HOFSTADTER, Douglas R. Metamagical themes: Questing for the essence of mind and Patterns, New York: BasicBooks 1985

MARX, Karl—O capital. São Paulo: Nova Cultural 1996

McFARLANE, Nigel—Professional Javascript. Birmingham, England: Wrox Press, 1999

McLEAN, Ruari—The Tames and Hudson Manual of Typography. Thames & Hudson, 1992

MOLES, Abraham A.—Teoria da Informação e Percepção Estética. Rio de Janeiro: Tempo Brasileiro, 1959

MORVAN, Pierre—Dicionário de Informática. Lisboa: Publicações Dom Quixote, 1984

PALAMIN, Vera Maria—Princípios da gestalt na organização da forma: abordagem bidimensional. São Pulo, 1985.

PFAFFENBERGER, Bryan, KARROW, Bill, WHITE, Chuck, SCHAFER, Steven M.—HTML, XHTML, and CSS Bible

Revista de Cultura Vozes—Cibernética e ideologia. Petrópolis: Vozes, 1972

RICHAUDEAU, François—La Lisibilité; Paris: Centre d'étude et de promotion de la lecture, 1969

RUDER, Emil—Typographie; Switzerland; Arthur Niggli Ltd, Teufen AR, 1967

SPENCER, Herbert—The Visible Word. London, Lund Huphries in association with the Royal College of art, 1969.

STOLFI, Ariane—Legibilidade e Evolução das mídias. Pesquisa de iniciação científica, não publicada, 2001

SWANN, Cal—Language and typography; London: Lund Humphries, 1991

SYBEX, Inc—HTML Complete. São Francisco, EUA: Sybex, 1999

TINKER, Miles A.—Legibility of Print; Iowa University , 1969

TUFTE, Edward—The visual display of quantitative information. Cheshire: Graphics Press, 1983

TROTSKY, Leon—O Programa de transição—A agonia do capitalismo e as tarefas da IV Internacional. Santos, negação da negação, 2005

TURNBULL, Athur T. ; BAIRD, Russel N.—Practical exercises in typography, layout, and design[by]; New York, Holt, Rinehart and Winston, 1968

TURNBULL, Athur T. ; BAIRD, Russel N—The graphics of communication: typography, layout, design; New York: Holt, Rinehart and Winston, 1964

TULLIS, Thomas S.; BOYNTON, Jennifer L. ; HERSH, Harry—Readability of Fonts in the Windows Environment; Fidelity Investments Boston, Massachusetts 1995 http://www.acm.org/sigchi/chi95/proceedings/intpost/tst_bdy.htm.

Sites

HTML, Javascript e CSS

www.w3schools.com
tutoriais e referência de várias linguagens de programação

<http://msdn.microsoft.com/archive/default.asp?url=/archive/welcome.asp>
biblioteca técnica de referência em computação

http://developer.mozilla.org/en/docs/Main_Page
mozilla developer center beta
documentação técnica

<http://javascript.internet.com/>
artigos e códigos em javascript

<http://www.javascriptkit.com/>
tutoriais e referência

www.w3.org
Site do W3C consortium, que define as tecnologias, guias e ferramentas para a internet

<http://www.artifice.web.pt/blog.html>
Tutoriais e referência de HTML e CSS em português

<http://www.thenoodleincident.com/tutorials/css/index.html>
Um guia de CSS

<http://www.tableless.com.br/aprenda/>
Artigos e estudos sobre tableless

<http://cssmania.com/>

sites em CSS

<http://www.maujor.com/>
css para webdesign, com dicas para resolver incompatibilidades

<http://www.spoon.com/javascript/>
Scripts e recursos para desenvolvedores

<http://www.scriptbrasil.com.br/>
códigos e apostilas de programação

<http://www.adaptivepath.com/publications/essays/archives/000385.php>
Artigo que cunhou o termo Ajax

<http://prototype.conio.net/>
Biblioteca base em Ajax

<http://www.csszengarden.com/tr/portuguese/>
site mostra as potencialidades do CSS

validadores de código

<http://jigsaw.w3.org/css-validator/>
serviço de validação de arquivos CSS

<http://validator.w3.org/>
validados de XHTML da W3C

<http://www.htmlhelp.com/tools/validator/>
serviço de validação HTML

Referência geral

<http://www.wikipedia.org/>
enciclopédia virtual livre

www.del.icio.us
serviço de bookmarks

<http://del.icio.us/garotasimpatica>
bookmarks da editora

<http://www1.uol.com.br/babylon/>
dicionário inglês/português

www.alexaholic.com
estatísticas comparadas de fluxo de visitas na internet

www.google.com
serviço de buscas

<http://www.google.com/intl/en/press/zeitgeist.html>
padrões de buscas

<http://code.google.com/>
recursos para desenvolvedores do google

http://www.snook.ca/archives/javascript/what_is_ajax/
What is Ajax? De um blog de tecnologia

<http://atelier-labs.incubadora.fapesp.br/portal/Members/garotasimpatica>
web site da editora, com as versões anteriores deste guia

referências para as ilustrações

www.chris.com/ASCII
Biblioteca de ASCII art

<http://www.mess.be/>
emoticons e nicks engraçadinhos para msn

www.concretoarmado.org
site da tendência de arquitetos da negação da negação

www.marxists.org
textos de autores marxistas

www.ebay.com
site de leilões

www.usp.br/~is/
site do matemático Imre Simon

tudo para que hoje sejamos capazes de nada tão ousado

