

HTML DOM 教程

目 录

HTML DOM 简介.....	- 1 -
DOM 节点.....	- 1 -
DOM 节点树.....	- 2 -
DOM 访问节点.....	- 3 -
DOM 节点信息.....	- 4 -
HTML DOM 的实例.....	- 5 -
HTML DOM 参考手册.....	- 6 -
DOM Window 对象.....	- 7 -
DOM Navigator 对象.....	- 10 -
DOM Screen 对象.....	- 11 -
DOM History 对象.....	- 11 -
DOM Location 对象.....	- 12 -
DOM Document 对象.....	- 13 -
DOM Anchor 对象.....	- 14 -
DOM Area 对象.....	- 15 -
DOM Base 对象.....	- 16 -
DOM Button 对象.....	- 17 -
DOM Canvas 对象.....	- 17 -
DOM Event 对象.....	- 18 -
DOM Form 对象.....	- 20 -
DOM Frame 对象.....	- 21 -
DOM Frameset 对象.....	- 22 -
DOM IFrame 对象.....	- 23 -
DOM Image 对象.....	- 23 -
DOM Input 对象.....	- 24 -
DOM Checkbox 对象.....	- 25 -
DOM FileUpload 对象.....	- 26 -
DOM Hidden 对象.....	- 27 -
DOM Radio 对象.....	- 29 -
DOM Reset 对象.....	- 30 -
DOM Submit 对象.....	- 31 -
DOM Link 对象.....	- 33 -
DOM Meta 对象.....	- 34 -
DOM Object 对象.....	- 34 -
DOM Option 对象.....	- 35 -
DOM Select 对象.....	- 36 -
DOM Table 对象.....	- 37 -
DOM TableCell 对象.....	- 39 -
DOM TableRow 对象.....	- 39 -
DOM Textarea 对象.....	- 40 -

HTML DOM 简介

您应当具备的基础知识

HTML / XHTML

JavaScript

什么是 DOM

通过 JavaScript, 您可以重构整个 HTML 文档。您可以添加、移除、改变或重排页面上的项目。

要改变页面的某个东西, JavaScript 就需要对 HTML 文档中所有元素进行访问的入口。这个入口, 连同对 HTML 元素进行添加、移动、改变或移除的方法和属性, 都是通过文档对象模型来获得的 (DOM)。

在 1998 年, W3C 发布了第一级的 DOM 规范。这个规范允许访问和操作 HTML 页面中的每一个单独的元素。所有的浏览器都执行了这个标准, 因此, DOM 的兼容性问题也几乎难觅踪影了。

DOM 可被 JavaScript 用来读取、改变 HTML、XHTML 以及 XML 文档。

DOM 被分为不同的部分 (核心、XML 及 HTML) 和级别 (DOM Level 1/2/3):

Core DOM

定义了一套标准的针对任何结构化文档的对象

XML DOM

定义了一套标准的针对 XML 文档的对象

HTML DOM

定义了一套标准的针对 HTML 文档的对象。

DOM 节点

HTML 文档中的每个成分都是一个节点。

节点

根据 DOM, HTML 文档中的每个成分都是一个节点。

DOM 是这样规定的:

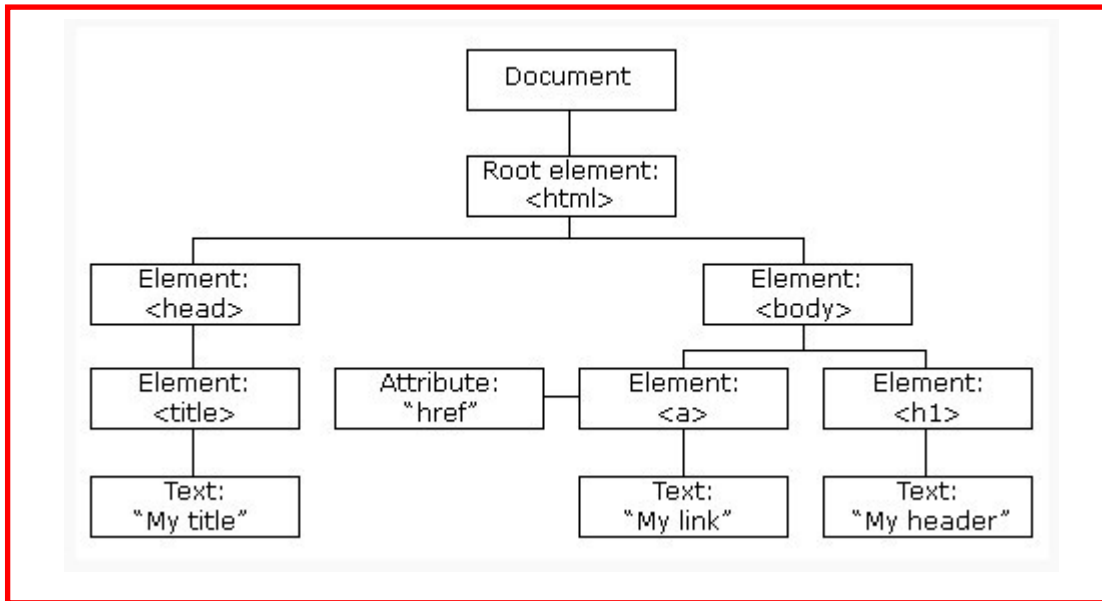
- 整个文档是一个文档节点
- 每个 HTML 标签是一个元素节点
- 包含在 HTML 元素中的文本是文本节点
- 每一个 HTML 属性是一个属性节点
- 注释属于注释节点

Node 层次

节点彼此都有等级关系。

HTML 文档中的所有节点组成了一个文档树 (或节点树)。HTML 文档中的每个元素、属性、文本等都代表着树中的一个节点。树起始于文档节点, 并由此继续伸出枝条, 直到处于这棵树最低级别的所有文本节点为止。

下面这个图片表示一个文档树（节点树）：



DOM 节点树

一棵节点树中的所有节点彼此都是有关系的。

文档树（节点数）

请看下面这个 HTML 文档：

```

<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>

```

上面所有的节点彼此间都存在关系。

除文档节点之外的每个节点都有父节点。举例，<head> 和 <body> 的父节点是 <html> 节点，文本节点 "Hello world!" 的父节点是 <p> 节点。

大部分元素节点都有子节点。比方说，<head> 节点有一个子节点：<title> 节点。<title> 节点也有一个子节点：文本节点 "DOM Tutorial"。

当节点分享同一个父节点时，它们就是同辈（同级节点）。比方说，<h1> 和 <p> 是同辈，因为它们的父节点均是 <body> 节点。

节点也可以拥有后代，后代指某个节点的所有子节点，或者这些子节点的子节点，以此类推。比方说，所有的文本节点都是 <html> 节点的后代，而第一个文本节点是 <head> 节点的后代。

节点也可以拥有先辈。先辈是某个节点的父节点，或者父节点的父节点，以此类推。比方说，所有的文本节点都可把 <html> 节点作为先辈节点。

DOM 访问节点

通过 DOM，您可访问 HTML 文档中的每个节点。

查找并访问节点

你可通过若干种方法来查找您希望操作的元素：

- 通过使用 `getElementById()` 和 `getElementsByName()` 方法
- 通过使用一个元素节点的 `parentNode`、`firstChild` 以及 `lastChild` 属性

`getElementById()` 和 `getElementsByName()`

`getElementById()` 和 `getElementsByName()` 这两种方法，可查找整个 HTML 文档中的任何 HTML 元素。

这两种方法会忽略文档的结构。假如您希望查找文档中所有的 `<p>` 元素，`getElementsByName()` 会把它全部找到，不管 `<p>` 元素处于文档中的哪个层次。同时，`getElementById()` 方法也会返回正确的元素，不论它被隐藏在文档结构中的什么位置。

这两种方法会像您提供任何你所需要的 HTML 元素，不论它们在文档中所处的位置！

`getElementById()` 可通过指定的 ID 来返回元素：

`getElementById()` 语法

```
document.getElementById("ID");
```

注释：`getElementById()` 无法工作在 XML 中。在 XML 文档中，您必须通过拥有类型 `id` 的属性来进行搜索，而此类型必须在 XML DTD 中进行声明。

`getElementsByName()` 方法会使用指定的标签名返回所有的元素（作为一个节点列表），这些元素是您在使用此方法时所处的元素的后代。

`getElementsByName()` 语法

```
document.getElementsByName("标签名称");
```

或者：

```
document.getElementById('ID').getElementsByName("标签名称");
```

实例 1

下面这个例子会返回文档中所有 `<p>` 元素的一个节点列表：

```
document.getElementsByTagName("p");
```

实例 2

下面这个例子会返回所有 `<p>` 元素的一个节点列表，且这些 `<p>` 元素必须是 `id` 为 `"maindiv"` 的元素的后代：

```
document.getElementById('maindiv').getElementsByTagName("p");
```

节点列表（`nodeList`）

当我们使用节点列表时，通常要把此列表保存在一个变量中，就像这样：

```
var x=document.getElementsByTagName("p");
```

现在，变量 `x` 包含着页面中所有 `<p>` 元素的一个列表，并且我们可以通过它们的索引号来访问这些 `<p>` 元素。

注释：索引号从 0 开始。

您可以通过使用 `length` 属性来循环遍历节点列表：

```
var x=document.getElementsByTagName("p");
for (var i=0;i<x.length;i++)
{
    // do something with each paragraph
}
```

您也可以通过索引号来访问某个具体的元素。

要访问第三个 `<p>` 元素，您可以这么写：

```
var y=x[2];
```

parentNode、firstChild 以及 lastChild

这三个属性 `parentNode`、`firstChild` 以及 `lastChild` 可遵循文档的结构，在文档中进行“短距离的旅行”。

请看下面这个 HTML 片段：

```
<table>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>Alaska</td>
  </tr>
</table>
```

在上面的 HTML 代码中，第一个 `<td>` 是 `<tr>` 元素的首个子元素（`firstChild`），而最后一个 `<td>` 是 `<tr>` 元素的最后一个子元素（`lastChild`）。

此外，`<tr>` 是每个 `<td>` 元素的父节点（`parentNode`）。

对 `firstChild` 最普遍的用法是访问某个元素的文本：

```
var x=[a paragraph];
var text=x.firstChild.nodeValue;
```

`parentNode` 属性常被用来改变文档的结构。假设您希望从文档中删除带有 `id` 为 `"maindiv"` 的节点：

```
var x=document.getElementById("maindiv");
x.parentNode.removeChild(x);
```

首先，您需要找到带有指定 `id` 的节点，然后移至其父节点并执行 `removeChild()` 方法。

根节点

有两种特殊的文档属性可用来访问根节点：

```
document.documentElement
```

```
document.body
```

第一个属性可返回存在于 XML 以及 HTML 文档中的文档根节点。

第二个属性是对 HTML 页面的特殊扩展，提供了对 `<body>` 标签的直接访问。

DOM 节点信息

`nodeName`、`nodeValue` 以及 `nodeType` 包含有关于节点的信息。

每个节点都拥有包含着关于节点某些信息的属性。这些属性是：

- nodeName (节点名称)
- nodeValue (节点值)
- nodeType (节点类型)

nodeName

nodeName 属性含有某个节点的名称。

- 元素节点的 nodeName 是标签名称
- 属性节点的 nodeName 是属性名称
- 文本节点的 nodeName 永远是 #text
- 文档节点的 nodeName 永远是 #document

注释：nodeName 所包含的 XML 元素的标签名称永远是大写的

nodeValue

对于文本节点，nodeValue 属性包含文本。

对于属性节点，nodeValue 属性包含属性值。

nodeValue 属性对于文档节点和元素节点是不可用的。

nodeType

nodeType 属性可返回节点的类型。最重要的节点类型是：

元素类型	节点类型
元素	1
属性	2
文本	3
注释	8
文档	9

HTML DOM 的实例

```
<html>
<head>
<script type="text/javascript">
function ChangeColor()
{
document.body.bgColor="yellow"
}
</script>
</head>
<body onclick="ChangeColor()">
```

Click on this document!

</body>

</html>

HTML DOM 参考手册

Browser 对象参考手册

对象	描述
Window	JavaScript 层级中的顶层对象，表示浏览器窗口。
Navigator	包含客户端浏览器的信息。
Screen	包含客户端显示屏的信息。
History	包含了浏览器窗口访问过的 URL。
Location	包含了当前 URL 的信息。

HTML DOM 对象参考手册

对象	描述
Document	代表整个 HTML 文档，可被用来访问页面中的所有元素
Anchor	代表 <a> 元素
Area	代表图像映射中的 <area> 元素
Base	代表 <base> 元素
Body	代表 <body> 元素
Button	代表 <button> 元素
Event	代表某个事件的状态
Form	代表 <form> 元素
Frame	代表 <frame> 元素
Frameset	代表 <frameset> 元素
Iframe	代表 <iframe> 元素
Image	代表 元素
Input button	代表 HTML 表单中的一个按钮
Input checkbox	代表 HTML 表单中的复选框
Input file	代表 HTML 表单中的文件上传
Input hidden	代表 HTML 表单中的隐藏域

Input password	代表 HTML 表单中的密码域
Input radio	代表 HTML 表单中的单选按钮
Input reset	代表 HTML 表单中的重置按钮
Input submit	代表 HTML 表单中的确认按钮
Input text	代表 HTML 表单中的文本输入域（文本框）
Link	代表 <link> 元素
Meta	代表 <meta> 元素
Object	代表 <Object> 元素
Option	代表 <option> 元素
Select	代表 HTML 表单中的选择列表
Style	代表单独的样式声明
Table	代表 <table> 元素
TableData	代表 <td> 元素
TableRow	代表 <tr> 元素
Textarea	代表 <textarea> 元素

DOM Window 对象

Window 对象表示浏览器中打开的窗口。

如果文档包含框架（frame 或 iframe 标签），浏览器会为 HTML 文档创建一个 window 对象，并为每个框架创建一个额外的 window 对象。

注释：没有应用于 window 对象的公开标准，不过所有浏览器都支持该对象。

Window 对象集合

集合	描述	IE	F	O
frames[]	返回窗口中所有命名的框架。 该集合是 Window 对象的数组，每个 Window 对象在窗口中含有一个框架或 <iframe>。属性 frames.length 存放数组 frames[] 中含有的元素个数。注意，frames[] 数组中引用的框架可能还包括框架，它们自己也具有 frames[] 数组。	4	1	9

Window 对象属性

属性	描述	IE	F	O
closed	返回窗口是否已被关闭。	4	1	9
defaultStatus	设置或返回窗口状态栏中的默认文本。	4	No	9
document	对 Document 对象的只读引用。请参阅 Document 对象。	4	1	9

history	对 History 对象的只读引用。请参数 History 对象。	4	1	9
innerheight	返回窗口的文档显示区的高度。	No	No	N 0
innerwidth	返回窗口的文档显示区的高度。	No	No	N 0
length	设置或返回窗口中的框架数量。	4	1	9
location	用于窗口或框架的 Location 对象。请参阅 Location 对象。	4	1	9
name	设置或返回窗口的名称。	4	1	9
Navigator	对 Navigator 对象的只读引用。请参数 Navigator 对象。	4	1	9
opener	返回对创建此窗口的窗口的引用。	4	1	9
outerheight	返回窗口的外部高度。	No	No	N 0
outerwidth	返回窗口的外部宽度。	No	No	N 0
pageXOffset	设置或返回当前页面相对于窗口显示区左上角的 X 位置。	No	No	N 0
pageYOffset	设置或返回当前页面相对于窗口显示区左上角的 Y 位置。	No	No	N 0
parent	返回父窗口。	4	1	9
Screen	对 Screen 对象的只读引用。请参数 Screen 对象。	4	1	9
self	返回对当前窗口的引用。等价于 Window 属性。	4	1	9
status	设置窗口状态栏的文本。	4	No	9
top	返回最顶层的先辈窗口。	4	1	9
window	window 属性等价于 self 属性，它包含了对窗口自身的引用。	4	1	9
screenLeft				
screenTop				
screenX	只读整数。声明了窗口的左上角在屏幕上的 x 坐标和 y 坐标。IE、Safari 和 Opera	4	1	9
screenY	支持 screenLeft 和 screenTop，而 Firefox 和 Safari 支持 screenX 和 screenY。			

Window 对象方法

方法	描述	IE	F	
alert()	显示带有一段消息和一个确认按钮的警告框。	4	1	↓
blur()	把键盘焦点从顶层窗口移开。	4	1	↓
clearInterval()	取消由 setInterval() 设置的 timeout。	4	1	↓
clearTimeout()	取消由 setTimeout() 方法设置的 timeout。	4	1	↓
close()	关闭浏览器窗口。	4	1	↓
confirm()	显示带有一段消息以及确认按钮和取消按钮的对话框。	4	1	↓
createPopup()	创建一个 pop-up 窗口。	4	No	N ↓

focus()	把键盘焦点给予一个窗口。	4	1	
moveBy()	可相对窗口的当前坐标把它移动指定的像素。	4	1	
moveTo()	把窗口的左上角移动到一个指定的坐标。	4	1	
open()	打开一个新的浏览器窗口或查找一个已命名的窗口。	4	1	
print()	打印当前窗口的内容。	5	1	
prompt()	显示可提示用户输入的对话框。	4	1	
resizeBy()	按照指定的像素调整窗口的大小。	4	1	
resizeTo()	把窗口的大小调整到指定的宽度和高度。	4	1.5	
scrollBy()	按照指定的像素值来滚动内容。	4	1	
scrollTo()	把内容滚动到指定的坐标。	4	1	
setInterval()	按照指定的周期（以毫秒计）来调用函数或计算表达式。	4	1	
setTimeout()	在指定的毫秒数后调用函数或计算表达式。	4	1	

Window 对象描述

Window 对象表示一个浏览器窗口或一个框架。在客户端 JavaScript 中，Window 对象是全局对象，所有的表达式都在当前的环境中计算。也就是说，要引用当前窗口根本不需要特殊的语法，可以把那个窗口的属性作为全局变量来使用。例如，可以只写 `document`，而不必写 `window.document`。

同样，可以把当前窗口对象的方法当作函数来使用，如只写 `alert()`，而不必写 `Window.alert()`。

除了上面列出的属性和方法，Window 对象还实现了核心 JavaScript 所定义的所有全局属性和方法。

Window 对象的 `window` 属性和 `self` 属性引用的都是它自己。当你想明确地引用当前窗口，而不仅仅是隐式地引用它时，可以使用这两个属性。除了这两个属性之外，`parent` 属性、`top` 属性以及 `frame[]` 数组都引用了与当前 Window 对象相关的其他 Window 对象。

要引用窗口中的一个框架，可以使用如下语法：

```
frame[i]      //当前窗口的框架
self.frame[i] //当前窗口的框架
w.frame[i]    //窗口 w 的框架
```

要引用一个框架的父窗口（或父框架），可以使用下面的语法：

```
parent      //当前窗口的父窗口
self.parent //当前窗口的父窗口
w.parent    //窗口 w 的父窗口
```

要从顶层窗口含有的任何一个框架中引用它，可以使用如下语法：

```
top          //当前框架的顶层窗口
self.top     //当前框架的顶层窗口
f.top        //框架 f 的顶层窗口
```

新的顶层浏览器窗口由方法 `Window.open()` 创建。当调用该方法时，应把 `open()` 调用的返回值存储在一个变量中，然后使用那个变量来引用新窗口。新窗口的 `opener` 属性反过来引用了打开它的那个窗口。

一般来说，`Window` 对象的方法都是对浏览器窗口或框架进行某种操作。而 `alert()` 方法、`confirm()` 方法和 `prompt` 方法则不同，它们通过简单的对话框与用户进行交互。

DOM Navigator 对象

Navigator 对象包含有关浏览器的信息。

注释：没有应用于 navigator 对象的公开标准，不过所有浏览器都支持该对象。

Navigator 对象集合

集合	描述	IE	F	O
<code>plugins[]</code>	<p>返回对文档中所有嵌入式对象的引用。</p> <p>该集合是一个 <code>Plugin</code> 对象的数组，其中的元素代表浏览器已经安装的插件。<code>Plug-in</code> 对象提供的是有关插件的信息，其中包括它所支持的 <code>MIME</code> 类型的列表。</p> <p>虽然 <code>plugins[]</code> 数组是由 IE 4 定义的，但是在 IE 4 中它却总是空的，因为 IE 4 不支持插件和 <code>Plugin</code> 对象。</p>	4	1	9

Navigator 对象属性

属性	描述	IE	F	O
<code>appCodeName</code>	返回浏览器的代码名。	4	1	9
<code>appMinorVersion</code>	返回浏览器的次级版本。	4	No	No
<code>appName</code>	返回浏览器的名称。	4	1	9
<code>appVersion</code>	返回浏览器的平台和版本信息。	4	1	9
<code>browserLanguage</code>	返回当前浏览器的语言。	4	No	9
<code>cookieEnabled</code>	返回指明浏览器中是否启用 <code>cookie</code> 的布尔值。	4	1	9
<code>cpuClass</code>	返回浏览器系统的 CPU 等级。	4	No	No
<code>onLine</code>	返回指明系统是否处于脱机模式的布尔值。	4	No	No
<code>platform</code>	返回运行浏览器的操作系统平台。	4	1	9
<code>systemLanguage</code>	返回 OS 使用的默认语言。	4	No	No
<code>userAgent</code>	返回由客户机发送服务器的 <code>user-agent</code> 头部的值。	4	1	9
<code>userLanguage</code>	返回 OS 的自然语言设置。	4	No	9

Navigator 对象方法

方法	描述	IE	F	O
----	----	----	---	---

javaEnabled()	规定浏览器是否启用 Java。	4	1	9
taintEnabled()	规定浏览器是否启用数据污点 (data tainting)。	4	1	9

Navigator 对象描述

Navigator 对象包含的属性描述了正在使用的浏览器。可以使用这些属性进行平台专用的配置。虽然这个对象的名称显而易见的是 Netscape 的 Navigator 浏览器，但其他实现了 JavaScript 的浏览器也支持这个对象。

Navigator 对象的实例是唯一的，可以用 Window 对象的 navigator 属性来引用它。

DOM Screen 对象

Screen 对象

Screen 对象实际上是 JavaScript 对象，而不是 HTML DOM 对象。

Screen 对象是由 JavaScript runtime engine 自动创建的，包含有关客户机显示屏幕的信息。

每个 Window 对象的 screen 属性都引用一个 Screen 对象。Screen 对象中存放着有关显示浏览器屏幕的信息。JavaScript 程序将利用这些信息来优化它们的输出，以达到用户的显示要求。例如，一个程序可以根据显示器的尺寸选择使用大图像还是使用小图像，它还可以根据显示器的颜色深度选择使用 16 位色还是使用 8 位色的图形。另外，JavaScript 程序还能根据有关屏幕尺寸的信息将新的浏览器窗口定位在屏幕中间。

Screen 对象的属性

属性	描述	IE	F	O
availHeight	返回显示屏幕的高度 (除 Windows 任务栏之外)。	4	1	9
availWidth	返回显示屏幕的宽度 (除 Windows 任务栏之外)。	4	1	9
bufferDepth	设置或返回在 off-screen bitmap buffer 中调色板的比特深度。	4	No	No
colorDepth	返回目标设备或缓冲器上的调色板的比特深度。	4	1	9
deviceXDPI	返回显示屏幕的每英寸水平点数。	6	No	No
deviceYDPI	返回显示屏幕的每英寸垂直点数。	6	No	No
fontSmoothingEnabled	返回用户是否在显示控制面板中启用了字体平滑。	4	No	No
height	返回显示屏幕的高度。	4	1	9
logicalXDPI	返回显示屏幕每英寸的水平方向的常规点数。	6	No	No
logicalYDPI	返回显示屏幕每英寸的垂直方向的常规点数。	6	No	No
pixelDepth	返回显示屏幕的颜色分辨率 (比特每像素)。	No	1	9
updateInterval	设置或返回屏幕的刷新率。	4	No	No
width	返回显示器屏幕的宽度。	4	1	9

DOM History 对象

History 对象实际上是 JavaScript 对象，而不是 HTML DOM 对象。

History 对象是由 JavaScript runtime engine 自动创建的，由一系列的 URL 组成。这些 URL 是用户在一个浏览器窗口内已访问的 URL。

History 对象最初设计来表示窗口的浏览历史。但出于隐私方面的原因，History 对象不再允许脚本访问已经访问

过的实际 URL。唯一保持使用的功能只有 `back()`、`forward()` 和 `go()` 方法。
 History 对象是 window 对象的一部分，可通过 `window.history` 属性对其进行访问。
 IE: Internet Explorer, F: Firefox, O: Opera.

History 对象的属性

属性	描述	IE	F	O
<code>length</code>	返回浏览器历史列表中的 URL 数量	4	1	9

History 对象的方法

方法	描述	IE	F	O
<code>back()</code>	加载 history 列表中的前一个 URL	4	1	9
<code>forward()</code>	加载 history 列表中的下一个 URL	4	1	9
<code>go()</code>	加载 history 列表中的某个具体页面	4	1	9

DOM Location 对象

Location 对象实际上是 JavaScript 对象，而不是 HTML DOM 对象。
 Location 对象是由 JavaScript runtime engine 自动创建的，包含有关当前 URL 的信息。

例子：把用户带到一个新的地址

Location 对象是 Window 对象的一个部分，可通过 `window.location` 属性来访问。
 有关 Location 对象的详细描述。

Location 对象的属性

属性	描述	IE	F	O
<code>hash</code>	设置或返回从井号 (#) 开始的 URL (锚)。	4	1	9
<code>host</code>	设置或返回主机名和当前 URL 的端口号。	4	1	9
<code>hostname</code>	设置或返回当前 URL 的主机名。	4	1	9
<code>href</code>	设置或返回完整的 URL。	4	1	9
<code>pathname</code>	设置或返回当前 URL 的路径部分。	4	1	9
<code>port</code>	设置或返回当前 URL 的端口号。	4	1	9
<code>protocol</code>	设置或返回当前 URL 的协议。	4	1	9
<code>search</code>	设置或返回从问号 (?) 开始的 URL (查询部分)。	4	1	9

Location 对象的方法

方法	描述	IE	F	O
<code>assign()</code>	加载新的文档。	4	1	9
<code>reload()</code>	重新加载当前文档。	4	1	9
<code>replace()</code>	用新的文档替换当前文档。	4	1	9

Location 对象的描述

Location 对象存储在 Window 对象的 Location 属性中，表示那个窗口中当前显示的文档的 Web 地址。它

的 href 属性存放的是文档的完整 URL，其他属性则分别描述了 URL 的各个部分。这些属性与 Anchor 对象（或 Area 对象）的 URL 属性非常相似。当一个 Location 对象被转换成字符串，href 属性的值被返回。这意味着你可以使用表达式 location 来替代 location.href。

不过 Anchor 对象表示的是文档中的超链接，Location 对象表示的却是浏览器当前显示的文档的 URL（或位置）。但是 Location 对象所能做的远远不止这些，它还能控制浏览器显示的文档的位置。如果把一个含有 URL 的字符串赋予 Location 对象或它的 href 属性，浏览器就会把新的 URL 所指的文档装载进来，并显示出来。除了设置 location 或 location.href 用完整的 URL 替换当前的 URL 之外，还可以修改部分 URL，只需要给 Location 对象的其他属性赋值即可。这样做就会创建新的 URL，其中的一部分与原来的 URL 不同，浏览器会将它装载并显示出来。例如，假设设置了 Location 对象的 hash 属性，那么浏览器就会转移到当前文档中的一个指定的位置。同样，如果设置了 search 属性，那么浏览器就会重新装载附加了新的查询字符串的 URL。

除了 URL 属性外，Location 对象的 reload() 方法可以重新装载当前文档，replace() 可以装载一个新文档而无须为它创建一个新的历史记录，也就是说，在浏览器的历史列表中，新文档将替换当前文档。

DOM Document 对象

每个载入浏览器的 HTML 文档都会成为 Document 对象。

Document 对象使我们可以从脚本中对 HTML 页面中的所有元素进行访问。

提示：Document 对象是 Window 对象的一部分，可通过 window.document 属性对其进行访问。

Document 对象集合

集合	描述	IE	F	O	W3C
all[]	提供对文档中所有 HTML 元素的访问。	4	1	9	No
anchors[]	返回对文档中所有 Anchor 对象的引用。	4	1	9	Yes
applets	返回对文档中所有 Applet 对象的引用。	-	-	-	-
forms[]	返回对文档中所有 Form 对象引用。	4	1	9	Yes
images[]	返回对文档中所有 Image 对象引用。	4	1	9	Yes
links[]	返回对文档中所有 Area 和 Link 对象引用。	4	1	9	Yes

Document 对象属性

属性	描述	IE	F	O	W3C
body	提供对 <body> 元素的直接访问。 对于定义了框架集的文档，该属性引用最外层的 <frameset>。				
cookie	设置或返回与当前文档有关的所有 cookie。	4	1	9	Yes
domain	返回当前文档的域名。	4	1	9	Yes

lastModified	返回文档被最后修改的日期和时间。	4	1	No	No
referrer	返回载入当前文档的文档的 URL。	4	1	9	Yes
title	返回当前文档的标题。	4	1	9	Yes
URL	返回当前文档的 URL。	4	1	9	Yes

Document 对象方法

方法	描述	IE	F	O	W3C
close()	关闭用 document.open() 方法打开的输出流，并显示选定的数据。	4	1	9	Yes
getElementById()	返回对拥有指定 id 的第一个对象的引用。	5	1	9	Yes
getElementsByName()	返回带有指定名称的对象集合。	5	1	9	Yes
getElementsByName()	返回带有指定标签名的对象集合。	5	1	9	Yes
open()	打开一个流，以收集来自任何 document.write() 或 document.writeln() 方法的输出。	4	1	9	Yes
write()	向文档写 HTML 表达式 或 JavaScript 代码。	4	1	9	Yes
writeln()	等同于 write() 方法，不同的是在每个表达式之后写一个换行符。	4	1	9	Yes

Document 对象描述

HTMLDocument 接口对 DOM Document 接口进行了扩展，定义 HTML 专用的属性和方法。很多属性和方法都是 HTMLCollection 对象（实际上是可以数组或名称索引的只读数组），其中保存了对锚、表单、链接以及其他可脚本元素的引用。

这些集合属性都源自于 0 级 DOM。它们已经被 Document.getElementsByTagName() 所取代，但是仍然常常使用，因为他们很方便。

write() 方法值得注意，在文档载入和解析的时候，它允许一个脚本向文档中插入动态生成的内容。注意，在 1 级 DOM 中，HTMLDocument 定义了一个名为 getElementById() 的非常有用的方法。在 2 级 DOM 中，该方法已经被转移到了 Document 接口，它现在由 HTMLDocument 继承而不是由它定义了。

DOM Anchor 对象

Anchor 对象表示 HTML 超链接。

在 HTML 文档中 <a> 标签每出现一次，就会创建 Anchor 对象。锚可用于创建指向另一个文档的链接（通过 href 属性），或者创建文档内的书签（通过 name 属性）。您可以通过搜索 Document 对象中的 anchors[] 数组来访问锚，或者使用 document.getElementById()。

Anchor 对象的属性

属性	描述	I	F	O	W3C
----	----	---	---	---	-----

		E			
accessKey	设置或返回访问一个链接的快捷键。	5	1	No	Yes
charset	设置或返回被链接资源的字符集。	6	1	9	Yes
coords	设置或返回逗号分隔列表，包含了图像映射中链接的坐标。	6	1	9	Yes
href	设置或返回被链接资源的 URL。	5	1	9	Yes
hreflang	设置或返回被链接资源的语言代码。	6	1	9	Yes
id	设置或返回一个链接的 id。	4	1	9	Yes
innerHTML	设置或返回一个链接的内容。	4	1	9	No
name	设置或返回一个链接的名称。	4	1	9	Yes
rel	设置或返回当前文档与目标 URL 之间的关系。	5	1	No	Yes
rev	设置或返回目标 URL 与之间当前文档的关系。	5	1	No	Yes
shape	设置或返回图像映射中某个链接的形状。	6	1	9	Yes
tabIndex	设置或返回某个链接的 Tab 键控制次序。	6	1	9	Yes
target	设置或返回在何处打开链接。	5	1	9	Yes
type	设置或返回被链接资源的 MIME 类型。	6	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Anchor 对象的方法

方法	描述	IE	F	O	W3C
blur()	把焦点从链接上移开。	5	1	9	Yes
focus()	给链接应用焦点。	5	1	9	Yes

DOM Area 对象

Area 对象代表图像映射的一个区域（图像映射指的是带有可点击区域的图像）在 HTML 文档中 <area> 标签每出现一次，就会创建一个 Area 对象。

Area 对象的属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问某个区域的快捷键。	5	1	No	Yes
alt	设置或返回当浏览器无法显示某个区域时的替换文字。	5	1	9	Yes
coords	设置或返回图像映射中可点击区域的坐标。	5	1	9	Yes
hash	设置或返回某个区域中 URL 的锚部分。	4	1	No	No
host	设置或返回某个区域中 URL 的主机名和端口。	4	1	No	No
href	设置或返回图像映射中链接的 URL。	4	1	9	Yes

id	设置或返回某个区域的 id。	4	1	9	Yes
noHref	设置或返回某个区域是否应是活动的还是非活动的。	5	1	9	Yes
pathname	设置或返回某个区域中的 URL 的路径名。	4	1	9	No
protocol	设置或返回某个区域中的 URL 的协议。	4	1	9	No
search	设置或返回某个区域中 URL 的查询字符串部分。	4	1	9	No
shape	设置或返回图像映射中某个区域的形状。	5	1	9	Yes
tabIndex	设置或返回某个区域的 tab 键控制次序。	5	1	9	Yes
target	设置或返回在何处打开区域中的 link-URL。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

DOM Base 对象

Base 对象代表 HTML 的 base 元素

在 HTML 文档中 <base> 每出现一次，Base 对象就会被创建。

Base 对象的属性

属性	描述	IE	F	O	W3C
href	设置或返回针对页面中所有链接的基准 URL	5	1	9	Yes
id	设置或返回 <base> 元素的 id	4	1	9	Yes
target	设置或返回针对页面中所有链接的默认目标框架	5	1	9	Yes

DOM Body 对象

Body 对象代表文档的主体 (HTML body) 。

Body 对象的属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性	5	1	9	Yes
dir	设置或返回文本的方向	5	1	9	Yes
id	设置或返回 body 的 id	5	1	9	Yes
lang	设置或返回元素的语言代码	5	1	9	Yes
title	设置或返回元素的咨询性的标题	5	1	9	Yes

DOM Button 对象

Button 对象代表一个按钮。

在 HTML 文档中 <button> 标签每出现一次，Button 对象就会被创建。

Button 对象的属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问某个按钮的快捷键。	6	1	9	Yes
disabled	设置或返回是否禁用按钮。	6	1	9	Yes
form	返回对包含按钮的表单的引用。	6	1	9	Yes
id	设置或返回按钮的 id。	6	1	9	Yes
name	设置或返回按钮的名称。	6	1	9	Yes
tabIndex	设置或返回按钮的 Tab 键控制次序。	6	1	9	Yes
type	返回按钮的表单类型。	6	1	9	Yes
value	设置或返回显示在按钮上的文本。	6	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM Canvas 对象

Canvas 对象表示一个 HTML 画布元素 - <canvas>。它没有自己的行为，但是定义了一个 API 支持脚本化客户端绘图操作。

你可以直接在该对象上指定宽度和高度，但是，其大多数功能都可以通过 CanvasRenderingContext2D 对象获得。这是通过 Canvas 对象的 getContext() 方法并且把直接量字符串 "2d" 作为唯一的参数传递给它而获得的。

<canvas> 标记在 Safari 1.3 中引入，在制作此参考页时，它在 Firefox 1.5 和 Opera 9 中也得到了支持。在 IE 中，<canvas> 标记及其 API 可以使用位于 excanvas.sourceforge.net 的 ExplorerCanvas 开源项目来模拟。

提示：如果希望学习如何使用 <canvas> 来绘制图形，可以访问 Mozilla 提供的 Canvas 教程（英文）以及相应的中文 Canvas 教程。

Canvas 对象的属性

height 属性

画布的高度。和一幅图像一样，这个属性可以指定为一个整数像素值或者是窗口高度的百分比。当这个值改变的时候，在该画布上已经完成的任何绘图都会擦除掉。默认值是 300。

width 属性

画布的宽度。和一幅图像一样，这个属性可以指定为一个整数像素值或者是窗口宽度的百分比。当这个值改变的时候，在该画布上已经完成的任何绘图都会擦除掉。默认值是 300。

Canvas 对象的方法

方法	描述
getContext()	返回一个用于在画布上绘图的环境。

DOM Event 对象

Event 对象代表事件的状态，比如事件在其中发生的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态。事件通常与函数结合使用，函数不会在事件发生前被执行！

事件句柄 (Event Handlers)

HTML 4.0 的新特性之一是能够使 HTML 事件触发浏览器中的行为，比如当用户点击某个 HTML 元素时启动一段 JavaScript。下面是一个属性列表，可将之插入 HTML 标签以定义事件的行为。

属性	此事件发生在何时...	IE	F	O	W3C
onabort	图像的加载被中断。	4	1	9	Yes
onblur	元素失去焦点。	3	1	9	Yes
onchange	域的内容被改变。	3	1	9	Yes
onclick	当用户点击某个对象时调用的事件句柄。	3	1	9	Yes
ondblclick	当用户双击某个对象时调用的事件句柄。	4	1	9	Yes
onerror	在加载文档或图像时发生错误。	4	1	9	Yes
onfocus	元素获得焦点。	3	1	9	Yes
onkey down	某个键盘按键被按下。	3	1	No	Yes
onkeypress	某个键盘按键被按下并松开。	3	1	9	Yes
onkey up	某个键盘按键被松开。	3	1	9	Yes
onload	一张页面或一幅图像完成加载。	3	1	9	Yes
onmousedown	鼠标按钮被按下。	4	1	9	Yes
onmousemove	鼠标被移动。	3	1	9	Yes
onmouseout	鼠标从某元素移开。	4	1	9	Yes
onmouseover	鼠标移到某元素之上。	3	1	9	Yes
onmouseup	鼠标按键被松开。	4	1	9	Yes

onreset	重置按钮被点击。	4	1	9	Yes
onresize	窗口或框架被重新调整大小。	4	1	9	Yes
onselect	文本被选中。	3	1	9	Yes
onsubmit	确认按钮被点击。	3	1	9	Yes
onunload	用户退出页面。	3	1	9	Yes

鼠标 / 键盘属性

属性	描述	IE	F	O	W3C
altKey	返回当事件被触发时, "ALT" 是否被按下。	6	1	9	Yes
button	返回当事件被触发时, 哪个鼠标按钮被点击。	6	1	9	Yes
clientX	返回当事件被触发时, 鼠标指针的水平坐标。	6	1	9	Yes
clientY	返回当事件被触发时, 鼠标指针的垂直坐标。	6	1	9	Yes
ctrlKey	返回当事件被触发时, "CTRL" 键是否被按下。	6	1	9	Yes
metaKey	返回当事件被触发时, "meta" 键是否被按下。	No	1	9	Yes
relatedTarget	返回与事件的目标节点相关的节点。	No	1	9	Yes
screenX	返回当某个事件被触发时, 鼠标指针的水平坐标。	6	1	9	Yes
screenY	返回当某个事件被触发时, 鼠标指针的垂直坐标。	6	1	9	Yes
shiftKey	返回当事件被触发时, "SHIFT" 键是否被按下。	6	1	9	Yes

IE 属性

除了上面的鼠标/事件属性, IE 浏览器还支持下面的属性:

属性	描述
cancelBubble	如果事件句柄想阻止事件传播到包容对象, 必须把该属性设为 <code>true</code> 。
fromElement	对于 <code>mouseover</code> 和 <code>mouseout</code> 事件, <code>fromElement</code> 引用移出鼠标的元素。
keyCode	对于 <code>keypress</code> 事件, 该属性声明了被敲击的键生成的 Unicode 字符码。对于 <code>keydown</code> 和 <code>keyup</code> 事件, 它指定了被敲击的键的虚拟键盘码。虚拟键盘码可能和使用的键盘的布局相关。
offsetX,offsetY	发生事件的地点在事件源元素的坐标系统中的 x 坐标和 y 坐标。
returnValue	如果设置了该属性, 它的值比事件句柄的返回值优先级高。把这个属性设置为 <code>false</code> , 可以取消发生事件的源元素的默认动作。
srcElement	对于生成事件的 Window 对象、Document 对象或 Element 对象的引用。
toElement	对于 <code>mouseover</code> 和 <code>mouseout</code> 事件, 该属性引用移入鼠标的元素。

x,y	事件发生的位置的 x 坐标和 y 坐标，它们相对于用 CSS 动态定位的最内层包容元素。
-----	--

标准 Event 属性

下面列出了 2 级 DOM 事件标准定义的属性。

属性	描述	IE	F	O	W3C
bubbles	返回布尔值，指示事件是否是起泡事件类型。	No	1	9	Yes
cancelable	返回布尔值，指示事件是否可拥可取消的默认动作。	No	1	9	Yes
currentTarget	返回其事件监听器触发该事件的元素。	No	1	9	Yes
eventPhase	返回事件传播的当前阶段。				Yes
target	返回触发此事件的元素（事件的目标节点）。	No	1	9	Yes
timeStamp	返回事件生成的日期和时间。	No	1	9	Yes
type	返回当前 Event 对象表示的事件的名称。	6	1	9	Yes

标准 Event 方法

下面列出了 2 级 DOM 事件标准定义的方法。IE 的事件模型不支持这些方法：

方法	描述	IE	F	O	W3C
initEvent()	初始化新创建的 Event 对象的属性。	No	1	9	Yes
preventDefault()	通知浏览器不要执行与事件关联的默认动作。	No	1	9	Yes
stopPropagation()	不再派发事件。	No	1	9	Yes

DOM Form 对象

Form 对象代表一个 HTML 表单。

在 HTML 文档中 <form> 每出现一次，Form 对象就会被创建。

Form 对象集合

集合	描述	IE	F	O	W3C
elements[]	包含表单中所有元素的数组。	5	1	9	Yes

Form 对象属性

属性	描述	IE	F	O	W3C
acceptCharset	服务器可接受的字符集。	No	No	No	Yes
action	设置或返回表单的 action 属性。	5	1	9	Yes

enctype	设置或返回表单用来编码内容的 MIME 类型。	6	1	9	Yes
id	设置或返回表单的 id。	5	1	9	Yes
length	返回表单中的元素数目。	5	1	9	Yes
method	设置或返回将数据发送到服务器的 HTTP 方法。	5	1	9	Yes
name	设置或返回表单的名称。	5	1	9	Yes
target	设置或返回表单提交结果的 Frame 或 Window 名。	5	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Form 对象方法

方法	描述	IE	F	O	W3C
reset()	把表单的所有输入元素重置为它们的默认值。	5	1	9	Yes
submit()	提交表单。	5	1	9	Yes

Form 对象事件句柄

事件句柄	描述	IE	F	O	W3C
onreset	在重置表单元素之前调用。	5	1	9	Yes
onsubmit	在提交表单之前调用。	5	1	9	Yes

DOM Frame 对象

Frame 对象代表一个 HTML 框架。

在 HTML 文档中 <frame> 每出现一次，就会创建一个 Frame 对象。

Frame 对象的属性

属性	描述	IE	F	O	W3C
contentDocument	容纳框架的内容的文档。	No	1	9	Yes

frameBorder	设置或返回是否显示框架周围的边框。	5	1	9	Yes
id	设置或返回框架的 id。	4	1	9	Yes
longDesc	设置或返回指向包含框架内容描述文档的 URL。	6	1	9	Yes
marginHeight	设置或返回框架的顶部和底部页空白。	5	1	9	Yes
marginWidth	设置或返回框架的左边缘和右边缘的空白。	5	1	9	Yes
name	设置或返回框架的名称。	5	1	9	Yes
noResize	设置或返回框架是否可调整大小。	5	1	9	Yes
scrolling	设置或返回框架是否可拥有滚动条。	No	1	No	Yes
src	设置或返回应被加载到框架中的文档的 URL。	5	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM Frameset 对象

Frameset 对象代表 HTML 框架集。

Frameset 对象的属性

属性	描述	IE	F	O	W3C
cols	设置或返回框架集中列的数目。	5	1	9	Yes
id	设置或返回框架集的 id。	4	1	9	Yes
rows	设置或返回框架集中行的数目。	5	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM IFrame 对象

IFrame 对象代表一个 HTML 的内联框架。

在 HTML 文档中 <iframe> 每出现一次，一个 IFrame 对象就会被创建。

IFrame 对象的属性

属性	描述	IE	F	O	W3C
align	根据周围的文字排列 iframe。	6	1	9	Yes
contentDocument	容纳框架的内容的文档。	No	1	9	Yes
frameBorder	设置或返回是否显示 iframe 周围的边框。	No	1	9	Yes
height	设置或返回 iframe 的高度。	5	1	9	Yes
id	设置或返回 iframe 的 id。	4	1	9	Yes
longDesc	设置或返回描述 iframe 内容的文档的 URL。	6	1	9	Yes
marginHeight	设置或返回 iframe 的顶部和底部的页空白。	5	1	9	Yes
marginWidth	设置或返回 iframe 的左侧和右侧的页空白。	5	1	9	Yes
name	设置或返回 iframe 的名称。	5	1	9	Yes
scrolling	设置或返回 iframe 是否可拥有滚动条。	No	1	No	Yes
src	设置或返回应载入 iframe 中的文档的 URL。	5	1	9	Yes
width	设置或返回 iframe 的宽度。	5	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM Image 对象

Image 对象代表一幅嵌入的图像。

 标签每出现一次，一个 Image 对象就会被创建。

Image 对象的属性

属性	描述	IE	F	O	W3C
align	设置或返回与内联内容的对齐方式。	5	1	9	Yes
alt	设置或返回无法显示图像时的替代文本。	5	1	9	Yes
border	设置或返回图像周围的边框。	4	1	9	Yes
complete	返回浏览器是否已完成对图像的加载。	4	1	9	No
height	设置或返回图像的高度。	4	1	9	Yes
hspace	设置或返回图像左侧和右侧的空白。	4	1	9	Yes
id	设置或返回图像的 id。	4	1	9	Yes
isMap	返回图像是否是服务器端的图像映射。	5	1	9	Yes
longDesc	设置或返回指向包含图像描述的文档的 URL。	6	1	9	Yes
lowsrc	设置或返回指向图像的低分辨率版本的 URL。	4	1	9	No
name	设置或返回图像的名称。	4	1	9	Yes
src	设置或返回图像的 URL。	4	1	9	Yes
useMap	设置或返回客户端图像映射的 usemap 属性的值。	5	1	9	Yes
vspace	设置或返回图像的顶部和底部的空白。	4	1	9	Yes
width	设置或返回图像的宽度。	4	1	9	Yes

Standard Properties

Property	Description	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

Image 对象的事件句柄

事件句柄	描述	IE	F	O	W3C
onabort	当用户放弃图像的装载时调用的事件句柄。	5	1	9	Yes
onerror	在装载图像的过程中发生错误时调用的事件句柄。	5	1	9	Yes
onload	当图像装载完毕时调用的事件句柄。	5	1	9	Yes

DOM Button 对象

Button 对象代表 HTML 文档中的一个按钮。

该元素没有默认的行为，但是必须有一个 onclick 事件句柄以便使用。

在 HTML 文档中 <input type="button"> 标签每出现一次，一个 Button 对象 就会被创建。

您可以通过遍历表单的 elements[] 数组来访问某个按钮，或者通过使用 document.getElementById()。

Button 对象的属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问按钮的快捷键。	5	1	9	Yes
alt	设置或返回当浏览器无法显示按钮时供显示的替代文本。	5	1	9	Yes

disabled	设置或返回是否禁用按钮。	5	1	9	Yes
form	返回对包含该按钮的表单对象的引用。	4	1	9	Yes
id	设置或返回按钮的 id。	4	1	9	Yes
name	设置或返回按钮的名称。	4	1	9	Yes
tabIndex	设置或返回按钮的 tab 键控制次序。	5	1	9	Yes
type	返回按钮的表单元素类型。	4	1	9	Yes
value	设置或返回在按钮上显示的文本。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Button 对象的方法

方法	描述	IE	F	O	W3C
blur()	把焦点从元素上移开。	4	1	9	Yes
click()	在某个按钮上模拟一次鼠标单击。	4	1	9	Yes
focus()	为某个按钮赋予焦点。	4	1	9	Yes

DOM Checkbox 对象

Checkbox 对象代表一个 HTML 表单中的一个选择框。

在 HTML 文档中 `<input type="checkbox">` 每出现一次，Checkbox 对象就会被创建。

您可以通过遍历表单的 `elements[]` 数组来访问某个选择框，或者通过使用 `document.getElementById()` 。

Checkbox 对象的属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问 checkbox 的快捷键。	4	1	9	Yes
alt	设置或返回当浏览器不支持 checkbox 时供显示的替代文本。	5	1	9	Yes
checked	设置或返回 checkbox 是否应被选中。	4	1	9	Yes
defaultChecked	返回 checked 属性的默认值。	4	1	9	Yes
disabled	设置或返回 checkbox 是否应被禁用。	4	1	9	Yes
form	返回对包含 checkbox 的表单的引用。	4	1	9	Yes

id	设置或返回 checkbox 的 id。	4	1	9	Yes
name	设置或返回 checkbox 的名称。	4	1	9	Yes
tabIndex	设置或返回 checkbox 的 tab 键控制次序。	4	1	9	Yes
type	返回 checkbox 的表单元素类型。	4	1	9	Yes
value	设置或返回 checkbox 的 value 属性的值	4	1	9	Yes

Standard Properties

Property	Description	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

Checkbox 对象的方法

方法	描述	IE	F	O	W3C
blur()	从 checkbox 上移开焦点。	4	1	9	Yes
click()	模拟在 checkbox 中的一次鼠标点击。	4	1	9	Yes
focus()	为 checkbox 赋予焦点。	4	1	9	Yes

DOM FileUpload 对象

在 HTML 文档中 `<input type="file">` 标签每出现一次，一个 FileUpload 对象就会被创建。

该元素包含一个文本输入字段，用来输入文件名，还有一个按钮，用来打开文件选择对话框以便图形化选择文件。该元素的 value 属性保存了用户指定的文件的名称，但是当包含一个 file-upload 元素的表单被提交的时候，浏览器会向服务器发送选中的文件的内容而不仅仅是发送文件名。

为安全起见，file-upload 元素不允许 HTML 作者或 JavaScript 程序员指定一个默认的文件名。HTML value 属性被忽略，并且对于此类元素来说，value 属性是只读的，这意味着只有用户可以输入一个文件名。当用户选择或编辑一个文件名，file-upload 元素触发 onchange 事件句柄。

您可以通过遍历表单的 elements[] 数组，或者通过使用 document.getElementById() 来访问 FileUpload 对象。

FileUpload 对象的属性

属性	描述	IE	F	O	W3C
accept	设置或返回指示文件传输的 MIME 类型的 MIME 类型列表（逗号分隔）。				Yes
accessKey	设置或返回访问 FileUpload 对象的快捷键。	4			Yes
alt	设置或返回当浏览器不支持 <code><input type="file"></code> 时供显示的替代文字。				Yes
default Value	设置或返回 FileUpload 对象的初始值。	4	1		Yes
disabled	设置或返回是否禁用 FileUpload 对象。	4			Yes
form	返回对包含 FileUpload 对象的表单的引用。	4	1		Yes
id	设置或返回 FileUpload 对象的 id。	4	1		Yes
name	设置或返回 FileUpload 对象的名称。	4	1		Yes
tabIndex	设置或返回定义 FileUpload 对象的 tab 键控制次序的索引号。	4			Yes

type	返回表单元素的类型。对于 FileUpload，则是 "file"。	4	1	Yes
value	返回由用户输入设置的文本后，FileUpload 对象的文件名。	4	1	Yes

Standard Properties

Property	Description	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

FileUpload 对象的方法

方法	描述	IE	F	O	W3C
blur()	从 FileUpload 对象上移开焦点。	4	1		Yes
focus()	为 FileUpload 对象赋予焦点。	4	1		Yes
select()	选取 FileUpload 对象。	4			Yes

DOM Hidden 对象

Hidden 对象代表一个 HTML 表单中的某个隐藏输入域。

这种类型的输入元素实际上是隐藏的。这个不可见的表单元素的 value 属性保存了一个要提交给 Web 服务器的任意字符串。如果想要提交并非用户直接输入的数据的话，就是用这种类型的元素。

在 HTML 表单中 <input type="hidden"> 标签每出现一次，一个 Hidden object 就会被创建。

您可通过遍历表单的 elements[] 数组来访问某个隐藏输入域，或者通过使用 document.getElementById()。

Hidden 对象的属性

属性	描述	IE	F	O	W3C
alt	设置或返回当浏览器不支持隐藏输入域时供显示的替代文本。	5	1	9	Yes
form	返回一个对包含隐藏域的表单的引用。	4	1	9	Yes
id	设置或返回隐藏域的 id。	4	1	9	Yes
name	设置或返回隐藏域的名称。	4	1	9	Yes
type	返回隐藏输入域的表单类型。	4	1	9	Yes
value	设置或返回隐藏域的 value 属性的值。	4	1	9	Yes

Standard Properties

Property	Description	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

DOM Password 对象

Password 对象代表 HTML 表单中的密码字段。

HTML 的 `<input type="password">` 标签在表单上每出现一次，一个 Password 对象就会被创建。

该文本输入字段供用户输入某些敏感的数据，比如密码等。当用户输入的时候，他的输入是被掩盖的（例如使用星号*），以防止旁边的人从他背后看到输入的内容。不过需要注意的是，当表单提交时，输入是用明文发送的。与类型为 "text" 的元素类似，当用户改变显示值时，它会触发 `onchange` 事件句柄。

您可以通过遍历表单的 `elements[]` array 来访问密码字段，或者通过使用 `document.getElementById()` 。

Password 对象属性

属性	描述	IE	F	O	W3C
<code>accessKey</code>	设置或返回访问密码字段的快捷键。	4	1	9	Yes
<code>alt</code>	设置或返回当不支持密码字段时显示的替代文字。	5	1	9	Yes
<code>defaultValue</code>	设置或返回密码字段的默认值。	4	1	9	Yes
<code>disabled</code>	设置或返回是否应被禁用密码字段。	5	1	9	Yes
<code>form</code>	返回对包含此密码字段的表单的引用。	4	1	9	Yes
<code>id</code>	设置或返回密码字段的 <code>id</code> 。	4	1	9	Yes
<code>maxLength</code>	设置或返回密码字段中字符的最大数目。	4	1	9	Yes
<code>name</code>	设置或返回密码字段的名称。	4	1	9	Yes
<code>readOnly</code>	设置或返回密码字段是否应当是只读的。	4	1	9	Yes
<code>size</code>	设置或返回密码字段的长度。	4	1	9	Yes
<code>tabIndex</code>	设置或返回密码字段的 <code>tab</code> 键控制次序。	4	1	9	Yes
<code>type</code>	返回密码字段的表单元素类型。	4	1	9	Yes
<code>value</code>	设置或返回密码字段的 <code>value</code> 属性的值。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
<code>className</code>	设置或返回元素的 <code>class</code> 属性。	5	1	9	Yes
<code>dir</code>	设置或返回文本的方向。	5	1	9	Yes
<code>lang</code>	设置或返回元素的语言代码。	5	1	9	Yes
<code>title</code>	设置或返回元素的 <code>title</code> 属性。	5	1	9	Yes

Password 对象方法

属性	描述	IE	F	O	W3C
----	----	----	---	---	-----

blur()	从密码字段移开焦点。	4	1	9	Yes
focus()	为密码字段赋予焦点。	4	1	9	Yes
select()	选取密码字段中的文本。	4	1	9	Yes

DOM Radio 对象

Radio 对象代表 HTML 表单中的单选按钮。

在 HTML 表单中 `<input type="radio">` 每出现一次，一个 Radio 对象就会被创建。

单选按钮是表示一组互斥选项按钮中的一个。当一个按钮被选中，之前选中的按钮就变为非选中的。

当单选按钮被选中或不选中时，该按钮就会触发 `onclick` 事件句柄。

您可通过遍历表单的 `elements[]` 数组来访问 Radio 对象，或者通过使用 `document.getElementById()`。

Radio 对象属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问单选按钮的快捷键。	4	1	9	Yes
alt	设置或返回在不支持单选按钮时显示的替代文本。	5	1	9	Yes
checked	设置或返回单选按钮的状态。	4	1	9	Yes
defaultChecked	返回单选按钮的默认状态。	4	1	9	Yes
disabled	设置或返回是否禁用单选按钮。	5	1	9	Yes
form	返回一个对包含此单选按钮的表单的引用。	4	1	9	Yes
id	设置或返回单选按钮的 id。	4	1	9	Yes
name	设置或返回单选按钮的名称。	4	1	9	Yes
tabIndex	设置或返回单选按钮的 tab 键控制次序。	4	1	9	Yes
type	返回单选按钮的表单类型。	4	1	9	Yes
value	设置或返回单选按钮的 value 属性的值。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Radio 对象方法

方法	描述	IE	F	O	W3C
blur()	从单选按钮移开焦点。	No	1	9	Yes
click()	在单选按钮上模拟一次鼠标点击。	No	2	9	Yes
focus()	为单选按钮赋予焦点。	No	1	9	Yes

DOM Reset 对象

Reset 对象代表 HTML 表单中的一个重置按钮。

在 HTML 表单中 `<input type="reset">` 标签每出现一次，一个 Reset 对象就会被创建。

当重置按钮被点击，包含它的表单中所有输入元素的值都重置为它们的默认值。默认值由 HTML value 属性或 JavaScript 的 defaultValue 属性指定。

重置按钮在重置表单之前触发 onclick 句柄，并且这个句柄可以通过返回 false 来取消。

参阅 Form.reset() 方法和 Form.onreset 事件句柄。

您可以通过遍历表单的 elements[] 数组来访问某个重置按钮，或者通过使用 document.getElementById()。

Reset 对象的属性

属性	描述	IE	F	O	W3C
accesskey	设置或返回访问重置按钮的快捷键。	4	1	9	Yes
alt	设置或返回当浏览器不支持重置按钮时供显示的替代文本。	5	1	9	Yes
disabled	设置或返回重置按钮是否应被禁用。	5	1	9	Yes
form	返回一个对包含此重置按钮的表单对象的引用。	4	1	9	Yes
id	设置或返回重置按钮的 id。	4	1	9	Yes
name	设置或返回重置按钮的名称。	4	1	9	Yes
tabIndex	设置或返回重置按钮的 tab 键控制次序。	4	1	9	Yes
type	返回重置按钮的表单元素类型。	4	1	9	Yes
value	设置或返回重置按钮上显示的文本。	4	1	9	Yes

Standard Properties

Property	Description	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

Reset 对象的方法

方法	描述	IE	F	O	W3C
blur()	从重置按钮上移开焦点。	4	1	9	Yes
click()	在重置按钮上模拟一次鼠标点击。	4	1	9	Yes
focus()	为重置按钮赋予焦点。	4	1	9	Yes

DOM Submit 对象

Submit 对象代表 HTML 表单中的一个提交按钮 (submit button)。

在 HTML 表单中 `<input type="submit">` 标签每出现一次，一个 Submit 对象就会被创建。

在表单提交之前，触发 `onclick` 事件句柄，并且一个句柄可以通过返回 `false` 来取消表单提交。

参阅 `Form.submit()` 方法和 `Form.onsubmit` 事件句柄。

实例：表单验证

您可以通过遍历表单的 `elements[]` 数组来访问某个提交按钮，或者通过使用 `document.getElementById()`。

Submit 对象属性

属性	描述	IE	F	O	W3C
<code>accessKey</code>	设置或返回访问提交按钮的快捷键。	4	1	9	Yes
<code>alt</code>	设置或返回当浏览器不支持提交按钮时供显示的替代文本。	5	1	9	Yes
<code>disabled</code>	设置或返回提交按钮是否应被禁用。	5	1	9	Yes
<code>form</code>	返回一个对包含此提交按钮的表单的引用。	4	1	9	Yes
<code>id</code>	设置或返回提交按钮的 <code>id</code> 。	4	1	9	Yes
<code>name</code>	设置或返回提交按钮的名称。	4	1	9	Yes
<code>tabIndex</code>	设置或返回提交按钮的 <code>tab</code> 键控制次序。	4	1	9	Yes
<code>type</code>	返回提交按钮的表单元素类型。	4	1	9	Yes
<code>value</code>	设置或返回在提交按钮上显示的文本。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
<code>className</code>	设置或返回元素的 <code>class</code> 属性。	5	1	9	Yes
<code>dir</code>	设置或返回文本的方向。	5	1	9	Yes
<code>lang</code>	设置或返回元素的语言代码。	5	1	9	Yes
<code>title</code>	设置或返回元素的 <code>title</code> 属性。	5	1	9	Yes

Submit 对象方法

方法	描述	IE	F	O	W3C
<code>blur()</code>	从提交按钮上移开焦点。	4	1	9	Yes
<code>click()</code>	在提交按钮上模拟一次鼠标点击。	4	1	9	Yes

focus()	为提交按钮赋予焦点。	4	1	9	Yes
---------	------------	---	---	---	-----

DOM Text 对象

Text 对象代表 HTML 表单中的文本输入域。

在 HTML 表单中 `<input type="text">` 每出现一次，Text 对象就会被创建。

该元素可创建一个单行的文本输入字段。当用户编辑显示的文本并随后把输入焦点转移到其他元素的时候，会触发 `onchange` 事件句柄。

您可以使用 HTML `<textarea>` 标记来创建多行文本输入。参阅 Textarea 对象。

对于掩码文本输入，把 `<input type="text">` 中的 `type` 设置为 `"password"`。参阅 Input Password。

您可以通过表单的 `elements[]` 数组来访问文本输入域，或者通过使用 `document.getElementById()`。

Text 对象属性

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问文本域的快捷键。	4	1	9	Yes
alt	设置或返回当浏览器不支持文本域时供显示的替代文本。	5	1	9	Yes
defaultValue	设置或返回文本域的默认值。	4	1	9	Yes
disabled	设置或返回文本域是否应被禁用。	5	1	9	Yes
form	返回一个对包含文本域的表单对象的引用。	4	1	9	Yes
id	设置或返回文本域的 <code>id</code> 。	4	1	9	Yes
maxLength	设置或返回文本域中的最大字符数。	4	1	9	Yes
name	设置或返回文本域的名称。	4	1	9	Yes
readOnly	设置或返回文本域是否应是只读的。	4	1	9	Yes
size	设置或返回文本域的尺寸。	4	1	9	Yes
tabIndex	设置或返回文本域的 <code>tab</code> 键控制次序。	4	1	9	Yes
type	返回文本域的表单元素类型。	4	1	9	Yes
value	设置或返回文本域的 <code>value</code> 属性的值。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 <code>class</code> 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes

title	设置或返回元素的 title 属性。	5	1	9	Yes
-------	--------------------	---	---	---	-----

Text 对象方法

方法	描述	IE	F	O	W3C
blur()	从文本域上移开焦点。	4	1	9	Yes
focus()	在文本域上设置焦点。	4	1	9	Yes
select()	选取文本域中的内容。	4	1	9	Yes

DOM Link 对象

Link 对象代表某个 HTML 的 <link> 元素。<link> 元素可定义两个链接文档之间的关系。<link> 元素被定义于 HTML 文档的 head 部分。

Link 对象属性

属性	描述	IE	F	O	W3C
charset	设置或返回目标 URL 的字符编码。	4	1	9	Yes
disabled	设置或返回目标 URL 是否当被禁用。	4	1	9	Yes
href	设置或返回被链接资源的 URL。	4	1	9	Yes
hreflang	设置或返回目标 URL 的基准语言。	4	1	9	Yes
id	设置或返回某个 <link> 元素的 id。	4	1	9	Yes
media	设置或返回文档显示的设备类型。	6	1	9	Yes
name	设置或返回 <link> 元素的名称。	4	No	No	Yes
rel	设置或返回当前文档与目标 URL 之间的关系。	4	1	9	Yes
rev	设置或返回目标 URL 与当前文档之间的关系。	4	1	9	Yes
type	设置或返回目标 URL 的 MIME 类型。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes

DOM Meta 对象

Meta 对象代表 HTML 的一个 <meta> 元素。

<meta> 元素可提供有关某个 HTML 元素的元信息 (meta-information)，比如描述、针对搜索引擎的关键词以及刷新频率。

Meta 对象属性

属性	描述	IE	F	O	W3C
content	设置或返回 <meta> 元素的 content 属性的值。	5	1	9	Yes
httpEquiv	把 content 属性连接到一个 HTTP 头部。	5	1	9	Yes
name	把 content 属性连接到某个名称。	5	1	9	Yes
scheme	设置或返回用于解释 content 属性的值的格式。	6	1	9	Yes

DOM Object 对象

Object 对象代表 HTML 的 <object> 元素。

<object> 元素用于嵌入的可执行内容。

Link 对象属性

属性	描述	IE	F	O	W3C
align	Sets or returns the alignment of the object according to the surrounding text	5	1	9	Yes
archive	Sets or returns a string that can be used to implement your own archive functionality for the object	6	1	9	Yes
border	Sets or returns the border around the object				Yes
code	Sets or returns the URL of the file that contains the compiled Java class	5	1	9	Yes
codeBase	Sets or returns the URL of the component	5	1	9	Yes
codeType					Yes
data			1	9	Yes
declare					Yes
form	Returns a reference to the object's parent form				Yes
height	Sets or returns the height of the object		1	9	Yes
hspace	Sets or returns the horizontal margin of the object	5	1	9	Yes
id	Sets or returns the id of the object	4	1	9	Yes

name	Sets or returns the name of the object	5	1	9	Yes
standby	Sets or returns a message when loading the object				Yes
tabIndex	Sets or returns the tab order for the object	5			Yes
type	Sets or returns the content type for data downloaded via the data attribute		1	9	Yes
useMap			1	9	Yes
vspace	Sets or returns the vertical margin of the object	5	1	9	Yes
width	Sets or returns the width of the object		1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM Option 对象

Option 对象代表 HTML 表单中下拉列表中的一个选项。

在 HTML 表单中 <option> 标签每出现一次，一个 Option 对象就会被创建。

您可通过表单的 elements[] 数组访问一个 Option 对象，或者通过使用 document.getElementById()。

Option 对象的属性

属性	描述	IE	F	O	W3C
defaultSelected	返回 selected 属性的默认值。	4	1	9	Yes
disabled	设置或返回选项是否应被禁用。	4	1	9	Yes
form	返回对包含该元素的 <form> 元素的引用。	4	1	9	Yes
id	设置或返回选项的 id。	4	1	9	Yes
index	返回下拉列表中某个选项的索引位置。	4	1	9	Yes
label	设置或返回选项的标记（仅用于选项组）。	6			Yes
selected	设置或返回 selected 属性的值。	4	1	9	Yes
text	设置或返回某个选项的纯文本值。	4	1	9	Yes
value	设置或返回被送往服务器的值。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM Select 对象

Select 对象代表 HTML 表单中的一个下拉列表。

在 HTML 表单中，<select> 标签每出现一次，一个 Select 对象就会被创建。

您可通过遍历表单的 elements[] 数组来访问某个 Select 对象，或者使用 document.getElementById()。

Select 对象集合

集合	描述	IE	F	O	W3C
options[]	返回包含下拉列表中的所有选项的一个数组。	4	1	9	Yes

Select 对象属性

属性	描述	IE	F	O	W3C
disabled	设置或返回是否应禁用下拉列表。	5	1	9	Yes
form	返回对包含下拉列表的表单的引用。	4	1	9	Yes
id	设置或返回下拉列表的 id。	4	1	9	Yes
length	返回下拉列表中的选项数目。	4	1	9	Yes
multiple	设置或返回是否选择多个项目。	4	1	9	Yes
name	设置或返回下拉列表的名称。	4	1	9	Yes
selectedIndex	设置或返回下拉列表中被选项目的索引号。	4	1	9	Yes
size	设置或返回下拉列表中的可见行数。	4	1	9	Yes
tabIndex	设置或返回下拉列表的 tab 键控制次序。	5	1	9	Yes
type	返回下拉列表的表单类型。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
----	----	----	---	---	-----

className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Select 对象方法

方法	描述	IE	F	O	W3C
add()	向下拉列表添加一个选项。	4	1	9	Yes
blur()	从下拉列表移开焦点。	4	1	9	Yes
focus()	在下拉列表上设置焦点。	4	1	9	Yes
remove()	从下拉列表中删除一个选项。	4	1	9	Yes

Select 对象事件句柄

事件句柄	描述	IE	F	O	W3C
onchange	当改变选择时调用的事件句柄。	4	1	9	Yes

DOM Table 对象

Table 对象代表一个 HTML 表格。

在 HTML 文档中 <table> 标签每出现一次，一个 Table 对象就会被创建。

Table 对象集合

集合	描述	IE	F	O	W3C
cells[]	返回包含表格中所有单元格的一个数组。	5	1	1	No
rows[]	返回包含表格中所有行的一个数组。	4	1	9	Yes
tBodies[]	返回包含表格中所有 tbody 的一个数组。	4			Yes

Table 对象属性

属性	描述	IE	F	O	W3C
align	表在文档中的水平对齐方式。（已废弃）	-	-	-	-
bgColor	表在文档中的水平对齐方式。（已废弃）	-	-	-	-
border	设置或返回表格边框的宽度。	4	1	9	Yes

caption	对表格的 <caption> 元素的引用。	4	1	9	Yes
cellPadding	设置或返回单元格内容和单元格边框之间的空白量。	4	1	9	Yes
cellSpacing	设置或返回在表格中的单元格之间的空白量。	4	1	9	Yes
frame	设置或返回表格的外部边框。	4	1	9	Yes
id	设置或返回表格的 id。	4	1	9	Yes
rules	设置或返回表格的内部边框（行线）。	4	1	9	Yes
summary	设置或返回对表格的描述（概述）。	6	1	9	Yes
tFoot	返回表格的 TFoot 对象。如果不存在该元素，则为 null。	4	1	9	Yes
tHead	返回表格的 THead 对象。如果不存在该元素，则为 null。	4	1	9	Yes
width	设置或返回表格的宽度。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

Table 对象方法

方法	描述	IE	F	O	W3C
createCaption()	为表格创建一个 caption 元素。	4	1	9	Yes
createTFoot()	在表格中创建一个空的 tFoot 元素。	4	1	9	Yes
createTHead()	在表格中创建一个空的 tHead 元素。	4	1	9	Yes
deleteCaption()	从表格删除 caption 元素及其内容。	4	1	9	Yes
deleteRow()	从表格删除一行。	4	1	9	Yes
deleteTFoot()	从表格删除 tFoot 元素及其内容。	4	1	9	Yes
deleteTHead()	从表格删除 tHead 元素及其内容。	4	1	9	Yes
insertRow()	在表格中插入一个新行。	4	1	9	Yes

DOM TableCell 对象

TableCell 对象代表一个 HTML 表格单元格。

在一个 HTML 文档中 <td> 标签每出现一次，一个 TableCell 对象就会被创建。

TableCell 对象属性

属性	描述	IE	F	O	W3C
abbr	设置或返回单元格中内容的缩写版本。	6	1	9	Yes
align	设置或返回单元格内部数据的水平排列方式。	4	1	9	Yes
axis	设置或返回相关单元格的一个逗号分隔的列表。	6	1	9	Yes
cellIndex	返回单元格在某行的单元格集合中的位置。	4	1	9	Yes
ch	设置或返回单元格的对齐字符。				Yes
chOff	设置或返回单元格的对齐字符的偏移量。				Yes
colSpan	单元格横跨的列数。	4	1	9	Yes
headers	设置或返回 header-cell 的 id 值。				Yes
id	设置或返回单元格的 id。	4	1	9	Yes
innerHTML	设置或返回单元格的开始标签和结束标签之间的 HTML。	4	1	9	No
rowSpan	设置或返回单元格可横跨的行数。	4	1	9	Yes
scope	设置或返回此单元格是否可提供标签信息。				Yes
vAlign	设置或返回表格单元格内数据的垂直排列方式。	4	1	9	Yes
width	设置或返回单元格的宽度。	4	1	9	Yes

标准属性

属性	描述	IE	F	O	W3C
className	设置或返回元素的 class 属性。	5	1	9	Yes
dir	设置或返回文本的方向。	5	1	9	Yes
lang	设置或返回元素的语言代码。	5	1	9	Yes
title	设置或返回元素的 title 属性。	5	1	9	Yes

DOM TableRow 对象

TableRow 对象代表一个 HTML 表格行。

在 HTML 文档中 <tr> 标签每出现一次，一个 TableRow 对象就会被创建。

TableRow 对象集合

集合	描述	IE	F	O	W3C
cells[]	返回包含行中所有单元格的一个数组。	4	1	9	Yes

TableRow 对象属性

属性	描述	IE	F	O	W3C
align	设置或返回在行中数据的水平排列。	4	1	9	Yes
ch	设置或返回在行中单元格的对齐字符。				Yes
chOff	设置或返回在行中单元格的对齐字符的偏移量。				Yes
id	设置或返回行的 id。	4	1	9	Yes
innerHTML	设置或返回行的开始标签和结束标签之间的 HTML。	5	1	9	No
rowIndex	返回该行在表中的位置。	4	1	9	Yes
sectionRowIndex	返回在 tBody 、tHead 或 tFoot 中，行的位置。				Yes
vAlign	设置或返回在行中的数据的垂直排列方式。	4	1	9	Yes

TableRow 对象方法

方法	描述	IE	F	O	W3C
deleteCell()	删除行中的指定的单元格。	4	1	9	Yes
insertCell()	在一行中的指定位置插入一个空的 <td> 元素。	4	1	9	Yes

DOM Textarea 对象

Textarea 对象代表 HTML 表单中的一个文本区 (text-area)。在表单中 <textarea> 标签每出现一次，一个 Textarea 对象就会被创建。

您可以通过索引相应表单的元素数组来访问某个 Textarea 对象，或者使用 getElementById()。

Textarea Object Properties

属性	描述	IE	F	O	W3C
accessKey	设置或返回访问 textarea 的键盘快捷键。	4	1	9	Yes
cols	设置或返回 textarea 的宽度。	4	1	9	Yes
defaultValue	设置或返回文本框中的初始内容。	4	1	9	Yes
disabled	设置或返回 textarea 是否应当被禁用。	5	1	9	Yes
form	返回对包含该 textarea 的表单对象的引用。	4	1	9	Yes
id	设置或返回某个 textarea 的 id	4	1	9	Yes
name	设置或返回 textarea 的名称。	4	1	9	Yes

readOnly	设置或返回 <code>textarea</code> 是否应当是只读的。	4	1	9	Yes
rows	设置或返回 <code>textarea</code> 的高度。	4	1	9	Yes
tabIndex	设置或返回 <code>textarea</code> 的 <code>tab</code> 键控制次序。	4	1	9	Yes
type	返回该文本框的表单类型。	4	1	9	Yes
value	设置或返回在 <code>textarea</code> 中的文本。	4	1	9	Yes

Standard Properties

属性	描述	IE	F	O	W3C
className	Sets or returns the class attribute of an element	5	1	9	Yes
dir	Sets or returns the direction of text	5	1	9	Yes
lang	Sets or returns the language code for an element	5	1	9	Yes
title	Sets or returns an element's advisory title	5	1	9	Yes

Textarea 对象的方法

方法	描述	IE	F	O	W3C
blur()	从 <code>textarea</code> 移开焦点。	4	1	9	Yes
focus()	在 <code>textarea</code> 上设置焦点。	4	1	9	Yes
select()	选择 <code>textarea</code> 中的文本。	4	1	9	Yes

Textarea 对象的事件句柄

事件句柄	描述	IE	F	O	W3C
onchange	当输入值改变时调用的事件句柄	4	1	9	Yes