1. JavaScript组成

**JavaScript和ECMAScript通常都被人们用来表达相同的含义**，但J**avaScript的含义却比ECMA-262中规定的要多得多**。

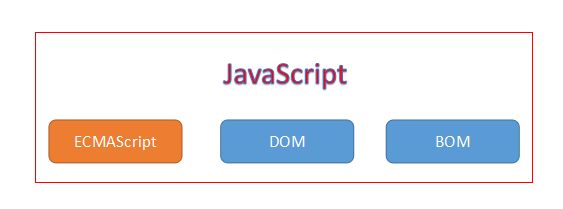
一个运行在的JavaScript实现应该由下列三个不同的部分组成:

1. 核心( ECMAScript)：JavaScript语言的核心，包含基本语法、数据类型、关键字...
2. **文档对象模型( DOM)**：提供**访问和操作网页内容的方法和接口**；可操作HTML中的**标签内容、样式、结构**；

document.write(“你好”)，输出内容至网页，这就是DOM中方法。

1. **浏览器对象模型( BOM)**：**提供与浏览器交互的方法和接口。**可操作浏览器窗口，比如浏览器地址栏，浏览器访问历史等；

window.alert(“你好”);，浏览器弹出一个消息对话框，这就是BOM中的方法。



# DOM和BOM简介

DOM:文档对象模型,主要是操作是HTML中标签；

BOM:浏览器对象模型,主要是操作的浏览器窗口；

DOM(Document Object Model)**即文档对象模型使JavaScript有能力与HTML 文档的所有元素**“对话”。

BOM(Browser Object Model)即浏览器对象模型使 JavaScript 有能力与浏览器“对话”。

BOM提供了访问浏览器各种功能部件的途径，例如浏览器窗口本身、浏览历史等；而DOM提供了访问浏览器中网页文档各元素的途径，包括页面中的超链接、表单等各种HTML元素及其内容。

BOM与DOM是独立于程序语言和平台的标准，W3C定义了一组标准接口，而这些接口在浏览器中以对象的形式实现。BOM与DOM均由一组对象组成，对象定义了属性和方法。

BOM与DOM关系非常密切。BOM的window对象中包含一个document属性，该属性是对DOM模型的document对象的引用，代表了当前窗口中网页文档所对应的document对象，通过该属性可以访问网页文档的各种内容。

* 1. 为什么要学习BOM和DOM

通过BOM对象完成以下内容:

1. 弹出新的浏览器窗口、移动、关闭浏览器窗口及调整窗口大小；

2. 提供WEB浏览器详细信息的导航对象；

3. 提供装载到浏览器中页面的详细信息的定位对象；

4. 提供用户屏幕分辩率详细信息的屏幕对象；

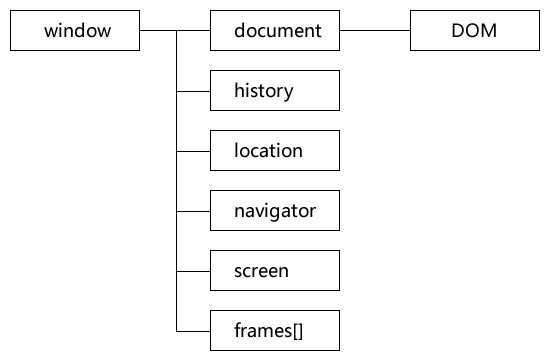
5. 对cookie的支持。 等等

通过DOM对象完成以下内容:

使得程序的脚本可以动态地访问并更新页面的内容，结构以及样式，文档可以被进一步加工处理并且处理结果可以显示在当前页面上. 如当当网的事例,已经所有网页上的动态效果(除flash外).

* 1. BOM模型对象结构图

BOM并不是只有一个对象而是有以下对象组成的.



BOM结构图说明:

* + - 1. 反映了BOM中各对象之间的层次关系；
      2. 在BOM中，window对象是顶层对象，其它对象均是其子对象；
      3. history浏览器的浏览历史
      4. location对象代表当前显示的文档的地址
      5. Navigator对象提供有关浏览器的各种信息
      6. screen对象提供显示器屏幕相关的一些信息
      7. Frames[]为window对象的数组型属性，每一个数组元素对应框架集(frameset)中的一个框架(frame)所对应的窗口。
      8. Document对象是对DOM的引用，代表了当前浏览器窗口中的网页文档

# Window对象

Window对象是BOM与DOM的顶层对象，表示浏览器窗口或者网页中的框架。

Window对象是脚本中的全局对象，可以在任何地方调用，脚本中任何对象的使用最终都要追溯到对window对象的访问，所以在使用window对象的各种属性和方法时，window前缀可以省略。

访问window对象或子对象属性和方法，要按照层次关系，使用“.”运算符将它们连接起来。

window.alert(); 简写方式 alert();

window.document.write(); 简写方式 document.write()

所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。

全局变量是 window 对象的属性。

全局函数是 window 对象的方法。

甚至 HTML DOM 的 document 也是 window 对象的属性之一.

* 1. window对象的属性和方法

学习一个对象的使用无非就是学习里面的属性和方法.

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。 |  |  |  |

**Window 对象属性**

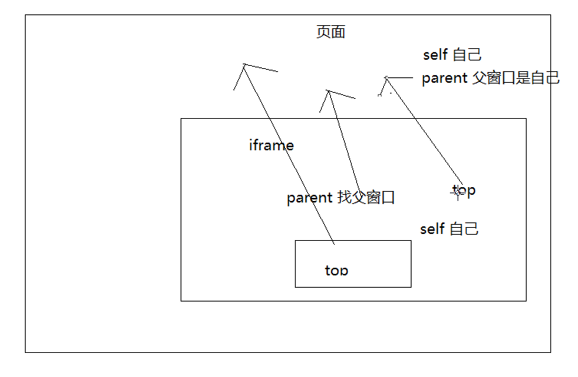
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性** | **描述** | **IE** | **F** | **O** |
| [closed](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_closed.asp.htm) | 返回窗口是否已被关闭。 | 4 | 1 | 9 |
| [document](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_document.asp.htm) | 对 Document 对象的只读引用。请参阅 [Document 对象](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_document.asp.htm)。 | 4 | 1 | 9 |
| [history](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_history.asp.htm) | 对 History 对象的只读引用。请参数 [History 对象](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_history.asp.htm)。 | 4 | 1 | 9 |
| [innerheight](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_innerheight_innerwidth.asp.htm) | 返回窗口的文档显示区的高度。 | No | No | No |
| [innerwidth](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_innerheight_innerwidth.asp.htm) | 返回窗口的文档显示区的宽度。 | No | No | No |
| length | 设置或返回窗口中的框架数量。 | 4 | 1 | 9 |
| [location](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_location.asp.htm) | 用于窗口或框架的 Location 对象。请参阅 [Location 对象](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_location.asp.htm)。 | 4 | 1 | 9 |
| [name](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_name.asp.htm) | 设置或返回窗口的名称。 | 4 | 1 | 9 |
| [Navigator](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_navigator.asp.htm) | 对 Navigator 对象的只读引用。请参数 [Navigator 对象](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_navigator.asp.htm)。 | 4 | 1 | 9 |
| [opener](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_opener.asp.htm) | 返回对创建此窗口的窗口的引用。 | 4 | 1 | 9 |
| [outerheight](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_outerheight.asp.htm) | 返回窗口的外部高度。 | No | No | No |
| [outerwidth](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_outerwidth.asp.htm) | 返回窗口的外部宽度。 | No | No | No |
| 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 | No |
| 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 | No |
| parent | 返回父窗口。 | 4 | 1 | 9 |
| [Screen](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_screen.asp.htm) | 对 Screen 对象的只读引用。请参数 [Screen 对象](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/dom_obj_screen.asp.htm)。 | 4 | 1 | 9 |
| [self](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_self.asp.htm) | 返回对当前窗口的引用。等价于 Window 属性。 | 4 | 1 | 9 |
| [status](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_status.asp.htm) | 设置窗口状态栏的文本。 | 4 | No | 9 |
| [top](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/prop_win_top.asp.htm) | 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 * screenY | 只读整数。声明了窗口的左上角在屏幕上的的 x 坐标和 y 坐标。IE、Safari 和 Opera 支持 screenLeft 和 screenTop，而 Firefox 和 Safari 支持 screenX 和 screenY。 | 4 | 1 | 9 |

Self：当前页面中window. 与window相等。

Top：当前窗口所在顶级窗口。主要是针对iframe中访问父窗口。找最最外面的那个window对象。如果自己就是最外面的那个window,那么top就是自己。

Parent：当前窗口中父级窗口。 主要是针对iframe中访问父窗口。如果没有父窗口，那么parent就是自己。

**注意：与超链接中的target="\_parent"描述类似。**（演示广告打开网址）

****

**Window 对象方法**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **描述** | **IE** | **F** | **O** |
| [alert()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_alert.asp.htm) | 显示带有一段消息和一个确认按钮的警告框。 | 4 | 1 | 9 |
| [blur()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_blur.asp.htm) | 把键盘焦点从顶层窗口移开。 | 4 | 1 | 9 |
| [close()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_close.asp.htm) | 关闭浏览器窗口。 | 4 | 1 | 9 |
| [confirm()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_confirm.asp.htm) | 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 | 9 |
| [createPopup()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_createpopup.asp.htm) | 创建一个 pop-up 窗口。 | 4 | No | No |
| [focus()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_focus.asp.htm) | 把键盘焦点给予一个窗口。 | 4 | 1 | 9 |
| [moveBy()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_moveby.asp.htm) | 可相对窗口的当前坐标把它移动指定的像素。 | 4 | 1 | 9 |
| [moveTo()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_moveto.asp.htm) | 把窗口的左上角移动到一个指定的坐标。 | 4 | 1 | 9 |
| [open()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_open.asp.htm) | 打开一个新的浏览器窗口或查找一个已命名的窗口。 | 4 | 1 | 9 |
| [print()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_print.asp.htm) | 打印当前窗口的内容。 | 5 | 1 | 9 |
| [prompt()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_prompt.asp.htm) | 显示可提示用户输入的对话框。 | 4 | 1 | 9 |
| ~~[resizeBy()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_resizeby.asp.htm)~~ | ~~按照指定的像素调整窗口的大小。~~ | 4 | 1 | 9 |
| ~~[resizeTo()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_resizeto.asp.htm)~~ | ~~把窗口的大小调整到指定的宽度和高度。~~ | 4 | 1.5 | 9 |
| ~~[scrollBy()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_scrollby.asp.htm)~~ | ~~按照指定的像素值来滚动内容。~~ | 4 | 1 | 9 |
| ~~[scrollTo()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_scrollto.asp.htm)~~ | ~~把内容滚动到指定的坐标。~~ | 4 | 1 | 9 |
| [setInterval()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_setinterval.asp.htm) | 按照指定的周期（以毫秒计）来调用函数或计算表达式。 | 4 | 1 | 9 |
| [setTimeout()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_settimeout.asp.htm) | 在指定的毫秒数后调用函数或计算表达式。 |  |  |  |
| [clearInterval()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_clearinterval.asp.htm) | 取消由 setInterval() 设置的 timeID。 | 4 | 1 | 9 |
| [clearTimeout()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/met_win_cleartimeout.asp.htm) | 取消由 setTimeout() 方法设置的 timeID | 4 | 1 | 9 |

* 1. 实例制作：定时器函数

setTimeout(fn,ms) 在指定的毫秒数后调用函数或计算表达式，函数返回一个定时器的timeId。

setInterval(fn,ms) 按照指定的周期（以毫秒计）来调用函数或计算表达式，函数返回一个定时器的timeId。

clearTimeout(timeId) 取消由 setTimeout() 方法设置的 timeId。

clearInterval(timeId) 取消由 setInterval() 设置的 timeId。

定时器一般用于制作动画效果，比如每隔一段时间就移动某个元素的位置；

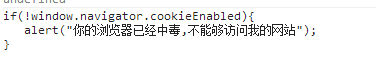
1. BOM其他对象
   1. screen对象

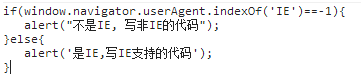
|  |  |
| --- | --- |
| availWidth | 显示器屏幕可用宽度，除任务栏外 |
| availHeight | 显示器屏幕可用高度，除任务栏外 |
| Width | 实际宽 |
| Height | 实际高 |

* 1. navigator对象

该对象提供有关浏览器的各种信息，所有浏览器都支持该对象

|  |  |
| --- | --- |
| userAgent | 返回由客户机发送服务器的 user-agent 头部的值。 |
| cookieEnabled | 返回指明浏览器中是否启用 cookie 的布尔值。 |





* 1. location对象

该对象存储了当前显示的文档的地址，包括完整的URL地址以及组成URL地址的各个部分。其中各个属性被重新赋值后，浏览器将自动刷新，载入新的URL地址。

属性:

|  |  |
| --- | --- |
| **属性** | **描述** |
| hash | 表示URL地址中的锚点部分，包括前导符“#”。可读可写 |
| host | 表示URL中的主机名和端口，即IP地址或者域名端口，可读可写 |
| hostname | 表示URL中的主机名部分。可读可写 |
| href | 表示当前页面完整的URL地址，可读可写。 |
| pathname | 表示URL中的页面路径部分，包含页面文件名称，可读可写 |
| port | 表示URL地址中的端口部分，可读可写 |
| protocol | 表示URL中的协议部分，包括尾部的“：”符号，可读可写。如http: |
| search | 表示URL中的参数部分，可读可写。如：?name=mary |

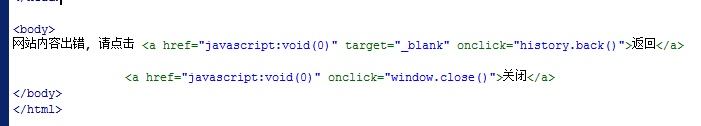
方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| assign() | 该方法为浏览器指定新的URL地址，与设置href属性的值相同。该方法同样会在history对象的浏览列表中增加一条记录。 |
| replace() | 使用指定的页面文档替换浏览器中当前的页面文档。方法是接受一个参数，该参数指定新的页面URL地址。replace() 方法不会在 History 对象中生成一个新的记录。当使用该方法时，新的 URL 将覆盖 History 对象中的当前记录  如：location.replace(“http://www.sina.com.cn”); |
| reload() | 用于刷新当前文档 。  Location.reload() 刷新页面：相当于按F5键 location.reload(true) 强制刷新页：shift+F5键 |

* 1. history对象

该对象记录了浏览器的浏览历史，并提供一组方法访问曾经访问过的历史页面。

|  |  |
| --- | --- |
| length属性 | 整数值，表示浏览器历史列表中的URL的个数 |
| back() | 返回到浏览器列表中当前页面之前的页面，与浏览器的“返回”功能相同 |
| forward() | 前进到浏览器中当前页面之后的页面，与浏览器的“前进”按钮功能相同 |
| go(1) | 访问浏览器列表中指定的页面，该方法接受一个参数。参数可正可负。 |



1. DOM文档对象
   1. 什么是DOM

DOM(文档对象模型)是一个使程序和脚本能够动态地访问和更新文档的内容、结构以及样式，并独立于平台和语言的接口。

W3C DOM提供了一套用于HTML和XML文档的标准对象，以及访问这些文档的标准接口(方法)。

DOM与具体的编程语言无关，可以在C、JavaScript、ActionScript、Java等语言中实现。

* 1. DOM分类

DOM按其内容可分为5个部分

核心DOM：是用于XML与HTML的共用接口； 主要是处理树结构；

XML DOM：XML专用接口，其实就是DOM进行增强，加入了一些适用于自己的接口； (操作XML文档文件)

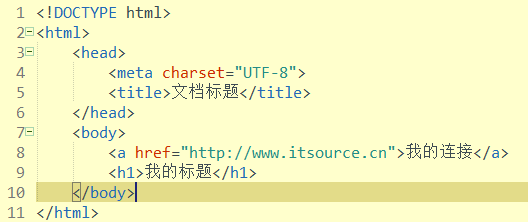
HTML DOM：HTML专用接口，其实就是DOM进行增强，加入了一些适用于自己的接口；(操作HTML文档文件)

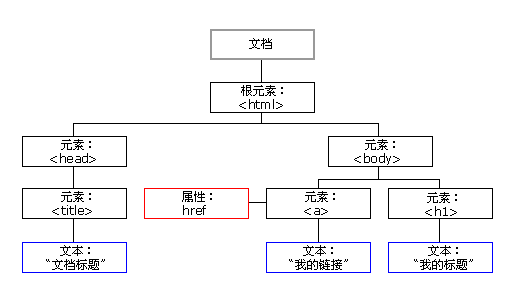
HTML DOM事件模型：定义DOM能够响应的事件，让页面具有交互能力。(比如点击就让一个元素隐藏)；

CSS DOM：CSS专用接口（操作方法）,操作标签的样式 （操作html中class,style）。

* 1. HTML节点树

HTML文档是一种树状的结构化文档、各标记之间是一种树状的层次关系。





节点：

\* 由结构图中我们可以看到，整个文档就是**一个文档节点**。 **document**

\* 而每一个HTML**标签都是一个元素节点**。 element

\* 标签中的**文字则是文本节点(回车，空格也是文本节点)**。Text

\* 标签的**属性是属性节点**。 Attribute

\*  **一切都是节点…… Node**

DOM 是针对xml(html)的基于树的API。

DOM树:节点（node）的层次。

DOM 把一个文档表示为一棵家谱树（父，子，兄弟）

DOM定义了Node的接口以及许多种节点类型来表示XML节点的多个方面

节点树

节点树的概念从图中一目了然，最上面的就是“树根”了。节点之间有父子关系，祖先与子孙关系，兄妹关系。这些关系从图中也很好看出来，直接连线的就是父子关系了。而有一个父亲的就是兄妹关系……

1. HTML DOM介绍
   1. 什么是HTML DOM

HTML DOM为HTML文档定义了一些专用的接口，称为HTML DOM，这些接口使对HTML文档的访问与操作更加方便。

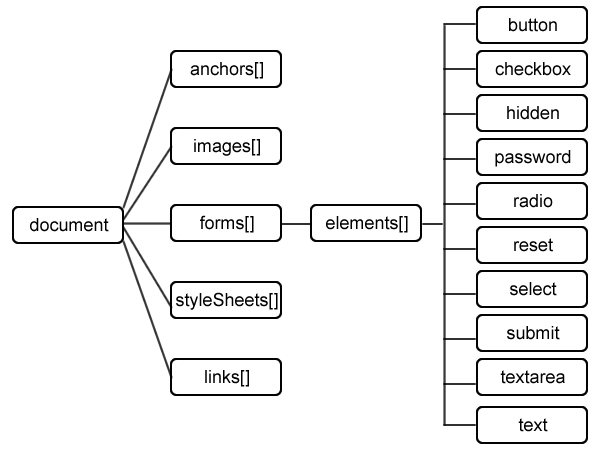
HTML DOM访问HTML元素更方便的特性

一个HTML标记对应一个相应类型的元素对象(Element Object)，如<img>标记对应一个image对象；

**HTML标记的属性对应于元素对象的一个同名属性。例如id\_img.src=“images/01.jpg”**

可以直接通过HTML标记的id属性的值访问标记对应的元素对象。

* 1. HTML DOM中提供的现成的对象来访问到里面的元素



* 1. HTML DOM中访问HTML元素方法
     1. 使用getElementById()方法

功能：在文档中查找指定的id节点，如果找到则返回该节点

例如：var id\_img=document.getElementById(“id\_img”)

说明：所有浏览器都支持。

* + 1. getElementsByTagName()

功能：返回带有指定标签名的对象的数组

例如：var list=ulObj.getElementsByTagName(“li”)

说明：节点列表length属性表示了节点列表中节点的个数

* + 1. getElementsByName()

功能：返回带有指定名字（name）对象的数组

例如：var hobby=document.getElementsByName(“hobby”)

<input type=”radio” vlaue=”写代码” name=”hobby”/>

<input type=”radio” vlaue=”玩游戏” name=”hobby”/>

<input type=”radio” vlaue=”看小说” name=”hobby”/>

* + 1. 快速访问方式（了解）
* **通过 name属性访问**
  + 在表单中，可以通过name属性进行访问。
  + 例如：document.form1.username.value

主要是访问表单中的元素

* **通过元素数组访问**
  + 常用的数组：images、links、forms等
  + 例如：document.images[0].src=“images/01.jpg”;
    1. HTML5 选择器访问（了解）

兼容性不好，故作为了解

document.querySelector(“选择器”);//返回页面中的第一个元素

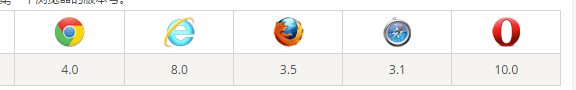
比如:document.querySelector(“div”); //获取页面中的第一个div;

document.querySelector(“#username”);//获取页面中id为username的元素

document.querySelectorAll(“选择器”);//返回页面中所有匹配的元素

比如：document.querySelectorAll(“div”);//获取页面中所有的div，返回值是一个数组；

兼容性（》=版本）：

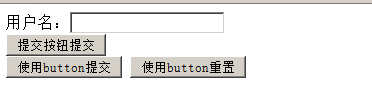


1. 表单对象-Form对象
   1. 什么是表单对象-Form对象
      * 1. 表单及表单元素均对应于相应的DOM对象，这些对象是document对象的子对象。
        2. 一个表单对应脚本中的一个form对象，form对象具有<form>标记的所有属性。
        3. 当表单元素定义name属性时（比如：name=“form1”），便为document对象添加了一个属性，该属性表示了该元素，通过document对象及属性值可以访问该元素。
        4. 有些浏览器并不支持脚本通过name属性访问元素，为了避免兼容问题，可以为元素的id和name指定相同的值。
        5. 当表单中存在多个相同name属性时，便为form元素对象添加了一个数组属性，数组名称为name属性值。
   2. 表单对象中的属性和方法

|  |  |
| --- | --- |
| **属性** | **描述** |
| elements | 表单所有元素组成的数组。可以分别通过整数、id、name作为索引来访问表单中元素。只读属性 |
| length | 表单中元素的个数。该值与elements.length相同，只读属性 |
| enctype | 对应于<form>的enctype属性，可设置或返回表单内容的 MIME 类型  默认值是 "application/x-www-form-urlencoded"  当type是 "file" 时，值是 "multipart/form-data" |
| action | 处理表单的页面的URL地址，对应于<form>的action属性 |
| method | 对应于<form>的method属性 |
| name | 对应于<form>的name属性 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| submit() | 提交表单。可通过JS脚本调用该方法提交表单，与单击submit按钮功能一样。但调用该方法提交表单时，不会执行onsubmit事件句柄。 |
| reset() | 重置表单。与单击reset按钮功能一样。该方法与submit()不同的是，当通过脚本调用时，仍触发执行onreset事件句柄。 |
| **事件** | **描述** |
| onsubmit | 在提交表单之**前**调用，如果onsubmit返回true，则提交表单，返回false，则阻止表单提交。 |
| onreset | 在表单被重置**前**调用，如果onreset返回true，则重置表单，返回false则阻止表单重置。 |

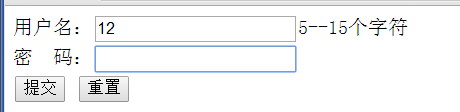
表单实例:



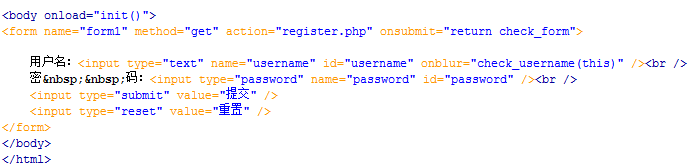


* 1. 单行文本对象

|  |  |
| --- | --- |
| **属性** | **描述** |
| value | 设置或返回文本域的 value 属性的值。 |
| name | 设置或返回文本域的名称。 |
| size | 设置或返回文本域的尺寸。 |
| type | 返回文本域的表单元素类型。 |
| readOnly | 设置或返回文本域是否应是只读的。 |
| maxLength | 设置或返回文本域中的最大字符数。 |
| form | 返回对包含此文本字段的表单的引用。 |
| disabled | 设置或返回文本域是否应被禁用。 |
| **方法** | **描述** |
| blur() | 从文本域上移开焦点。 |
| focus() | 在文本域上设置焦点。 |
| select() | 选取文本域中的内容。 |







* 1. 常用的属性操作

如果我元素上有的属性,都可以通过

**元素对象.属性名** 或者 **元素对象["属性名"];**

|  |
| --- |
| **1.操作属性名和默认属性值相同的属性.**  如:checked（一般指单选或者多选项）  **2. 操作属性名和默认属性值相同的属性.**  如: selected（一般指下拉菜单选项）  **3.操作class属性**  属性名为:className  **4.操作style的属性**  如:background-color  **5.操作readonly属性:readonly--->readOnly**  readOnly 设置或返回文本域是否应是只读的。  **6. 操作disabled属性: disabled --->true**  disabled 设置或返回文本域是否应被禁用。 |