

1902



太原理工大学
TAIYUAN UNIVERSITY OF TECHNOLOGY

第十三章 JavaScript对象编程

Web程序设计基础



学习目标

掌握**文档对象模型**、属性
及其方法



掌握**浏览器对象模型**、
属性及方法

掌握**事件驱动应用**



目录

- 13.1 文档对象模型 (DOM)
- 13.2 窗口对象
- 13.3 文档对象
- 13.4 表单对象
- 13.5 练习



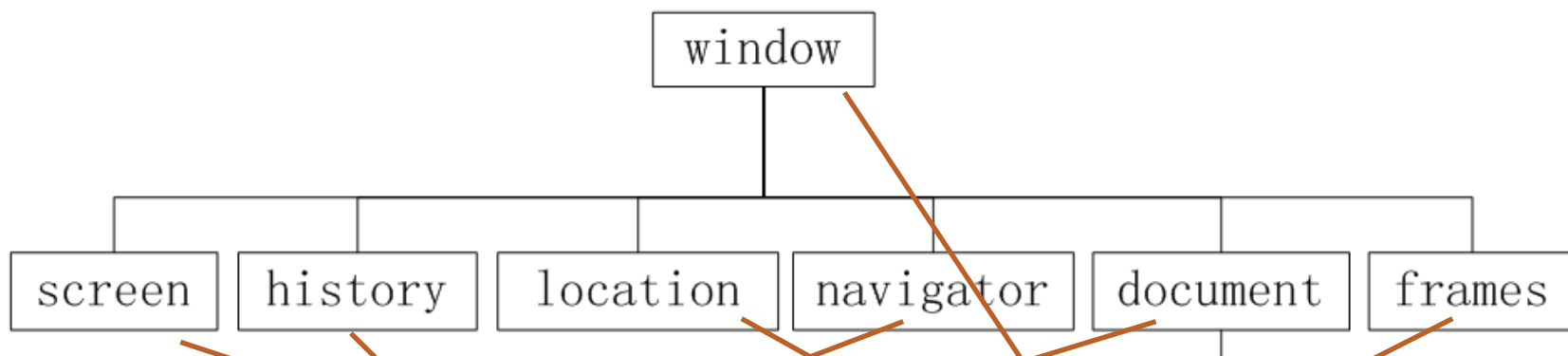


13.1 BOM和DOM概述

1. BOM模型

浏览器对象模型（Browser Object Model, BOM）

是用于描述浏览器中对象与对象之间层次关系的模型，提供了独立于页面内容、并能够与浏览器窗口进行交互的对象结构。



frames对象包含了框架的页面布局信息，以及每一个框架所对应的window对象。



13.1 BOM和BOM概述

2. DOM模型

文档对象模型（Document Object Model，DOM）属于**BOM**的一部分，用于对**BOM**中的核心对象document进行操作，它是一种与平台、语言无关的**接口**，允许程序和脚本动态地访问或更新HTML、XML文档的内容、结构和样式，且提供了一系列的函数和对象来实现访问、添加、修改及删除操作。

DOM将网页中文档的对象关系规划为**节点层级**，构成它们之间的**等级关系**，这种各对象间的层次结构被称为**节点树**。



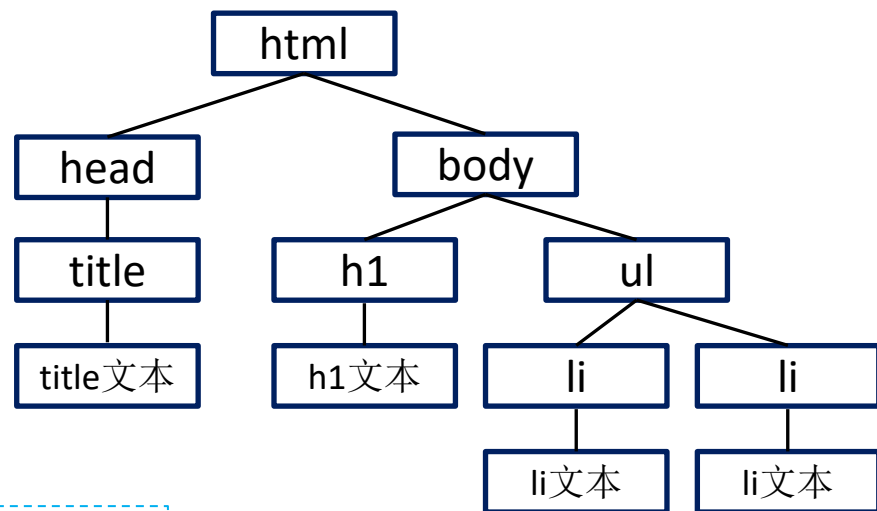
13.1 BOM和BOM概述

2. DOM模型

一个文档的树形结构是由各种不同的节点组成，**Html**代表根节点。除了根节点以外，每个节点都有一个父节点；每个节点也可以有许多子节点；具有相同父节点的节点叫兄弟节点。

在**DOM**中，每个节点都是一个对象，这些对象都有若干属性和方法。

JavaScript通过使用节点的**属性和方法**可以访问**指定元素**和**相关元素**，从而得到文档中的各个**元素对象**。



Html文档的树形结构



13.1 DOM和BOM概述

2. DOM模型

可以通过id、name、class及标签名来访问指定的元素。

类型	方法	说明
访问指定节点	getElementById()	获取拥有指定ID的第一个元素对象的引用
	getElementsByName()	获取带有指定名称的元素对象集合
	getElementsByTagName()	获取带有指定标签名的元素对象集合
	getElementsByClassName()	获取指定class的元素对象集合（不支持IE6~8浏览器）



13.1 DOM和BOM概述

2. DOM模型

也可以访问指定的元素父、子或兄弟元素。

属性	说明
parentNode	元素节点的父节点
childNodes	元素节点的子节点数组
firstChild	第一个子节点
lastChild	最后一个子节点
previousSibling	前一个兄弟节点
nextSibling	后一个兄弟节点



【例13-1】获取并输出HTML文档的DOM节点。

 [\[点击查看案例\]](#)



```
<!doctype html>
<html><head>
<meta charset="utf-8">
<title>HTML DOM</title>
<script>
window.onload = function(){
var myPage = document.documentElement;
alert(myPage.nodeName);
var pageBody = document.body;
alert(pageBody.nodeName);
var curTarget = document.getElementById("pt");
alert("父节点: " + curTarget.parentNode.nodeName);
alert("第一个: " +
curTarget.parentNode.firstChild.nextSibling.innerHTML);
alert("上一个: " + curTarget.previousSibling.previousSibling.innerHTML);
alert("下一个: " + curTarget.nextSibling.nextSibling.innerHTML);
```

显示节点名称

获取对象起始结束标签间的内容



[点击查看案例]



```
alert("最后一个: " +  
curTarget.parentNode.lastChild.previousSibling.innerHTML);  
}  
</script>  
</head>  
<body>  
  <ol>  
    <li>五台山</li>  
    <li>峨眉山</li>  
    <li id="pt">普陀山</li>  
    <li>九华山</li>  
    <li>梵净山</li>  
  </ol>  
</body>  
</html>
```



13.1 DOM和BOM概述

3. 事件机制

JavaScript是基于对象（**Object Based**）的语言，而基于对象的基本特征就是采用事件驱动（**Event Driven**），在图形界面环境下使得输入变的简单。

事件（**Event**）是指可以被JavaScript侦测到的行为（**ACTION**）

事件驱动（**Event Driver**）是指由事件引发的一连串程序的动作。

事件处理程序（**Event Handler**）是指对事件进行处理的程序或函数。



13.1 DOM和BOM概述

3. 事件机制

常见事件的类型



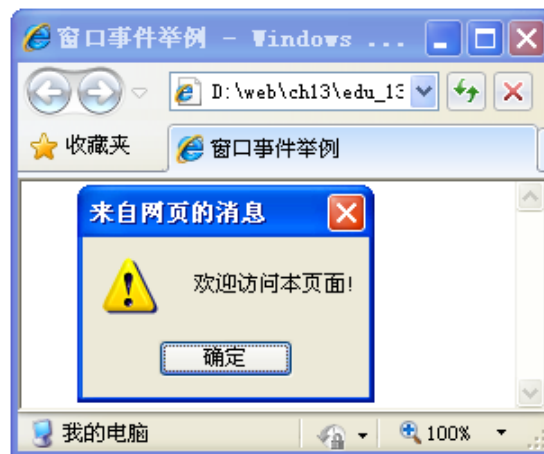
鼠标事件



键盘事件



表单事件



窗口事件



13.1 DOM和BOM概述

3. 事件机制

事件处理的过程

触发事件

启动事件处
理程序

事件处理程
序作出反应



13.1 DOM和BOM概述

3. 事件机制

触发事件

通过**事件句柄**（**Event**）触发事件，事件句柄是指事件发生要进行的操作。如onclick。

类型	事件句柄	说明
鼠标事件	onclick	当鼠标被单击时触发事件
	ondblclick	当鼠标被双击时触发事件
	onmousedown	当鼠标按钮被按下时触发事件
	onmousemove	当鼠标指针移动时触发事件
	onmouseout	当鼠标指针移出某元素时触发事件
	onmouseover	当鼠标指针悬停于某元素之上时触发事件
	onmouseup	当鼠标按钮被松开时触发事件



13.1 DOM和BOM概述

类型	事件句柄	说明
键盘事件	onkeydown	当键盘被按下时触发事件
	onkeypress	当键盘被按下后又松开时触发事件
	onkeyup	当键盘被松开时触发事件
表单事件	onchange	当元素改变时触发事件
	onsubmit	当表单被提交时触发事件
	onreset	当表单被重置时触发事件
	onselect	当元素被选取时触发事件
	onblur	当元素失去焦点时触发事件
	onfocus	当元素获得焦点时触发事件
窗口事件	onload	当文档载入时触发事件
	onunload	当文档卸载时触发事件



13.1 DOM和BOM概述

3. 事件机制

启动事件

通过**绑定**事件处理程序启动事件。

□ **静态绑定**：在HTML标签内，使用**事件句柄**为该元素绑定指定的事件处理函数。

```
<input type="button" value="单击" onclick="alert('大家好')">
```

□ **动态绑定**：通过JavaScript脚本获得文档中的某一对象，然后通过事件句柄为该元素绑定指定的事件处理函数

```
var myButton = document.getElementById("myButton");  
myButton.onmouseover=function(){alert('大家好');}
```




事件绑定案例1

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>在Javascript中调用事件处理程序</title>
```

```
</head>
```

```
<body>
```

```
<input type="button" onclick="doSomething()" id="myButton"/>
```

```
<script type="text/javascript">
```

```
function doSomething(){  
    alert('响应用户的操作');
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

 [\[点击查看案例\]](#)





事件绑定案例2

```
<!DOCTYPE html>
<html>
  <head>
    <title>在Javascript中调用事件处理程序</title>
  </head>
  <body>
    <input type="button" id="myButton"/>
    <script type="text/javascript">
      var myButton =document.getElementById("myButton");
      myButton.onmouseover=function(){
        alert('鼠标移到按钮上面');
      }
    </script>
  </body>
</html>
```

 [\[点击查看案例\]](#)





【例13-2】事件驱动示例。

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript 事件驱动</title>
    <script>
      function countTotal(){
        var elements = document.getElementsByTagName("input");
        window.alert("input类型节点总数: " + elements.length);}
      function submitValue(){
        var sv = document.getElementById("submit");
        window.alert("按钮的value是: " + sv.value);}
    </script>
  </head>
```

 [\[点击查看案例\]](#)



```
<body>
```

```
<form id="form1" name="form1" method="post">
```

```
<table width="300" border="1" cellpadding="0" cellspacing="0">
```

```
<tbody>
```

```
<tr>
```

```
<td width="20%">用户名</td>
```

```
<td width="80%"><input type="text" name="input1" id="input1"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>密码</td>
```

```
<td><input type="password" name="password1" id="password1"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td><input type="submit" name="submit" id="submit" value="提交"></td>
```

```
</tr>
```

```
</tbody></table></form>
```

```
<a href="javascript:void(0);" onClick="countTotal();">统计input子节点总数</a>
```

```
<a href="javascript:void(0);" onClick="submitValue();">获取提交按钮内容</a>
```

```
</body></html>
```

 [\[点击查看案例\]](#)





13.2 窗口 (window) 对象

window对象在客户端JavaScript中扮演重要的角色，它是客户端程序的全局（默认）对象，它还是客户端对象层次的根。

1. 窗口 (window) 介绍

window对象是一个全局对象，因此可以直接使用全局对象的属性和方法，而无须写出全局对象的对象名。

window对象在某种意义上代表了浏览器本身，它包含了大量关于浏览器的属性。



13.2 窗口 (window) 对象

属 性	描 述
closed	布尔值，窗口关闭时此属性为 true ，默认为 false
defaultStatus, status	字符串，设置或返回窗口状态栏中的默认文本
document	对 Document 对象的只读引用，该对象表示在窗口中显示的 HTML 文件
history	对 History 对象的只读引用，该对象代表用户浏览器窗口的历史
innerheight, innerwidth	窗口的文档显示区的高度和宽度
outerheight, outerwidth	窗口的外部的高度和宽度
location	对 Location 对象的引用，该对象代表在窗口中显示文档的 URL
name	设置或返回窗口的名称
opener	对创建此窗口的 window 对象的引用，如果当前窗口被用户创建，则值为 null
pageXOffset, pageYOffset	设置或返回当前页面相对于窗口显示区左上角的滚动位置
parent	返回父窗口
self	返回对当前窗口的引用，自引用属性
top	返回最顶层的先辈窗口
window	window 属性等价于 self 属性，它包含了对窗口自身的引用



13.2 窗口 (window) 对象

方法	描述
<code>close()</code>	关闭窗口
<code>open()</code>	打开一个新的浏览器窗口或查找一个已命名的窗口
<code>find()</code> , <code>home()</code> , <code>print()</code> , <code>stop()</code>	执行浏览器查找、主页、打印和停止按钮的功能, 就像用户鼠标单击浏览器中这些按钮一样
<code>focus()</code> , <code>blur()</code>	请求或放弃窗口的键盘焦点, <code>focus()</code> 函数还将把窗口置于最上层, 使窗口可见
<code>moveBy()</code> , <code>moveTo()</code>	移动窗口
<code>resizeBy()</code> , <code>resizeTo()</code>	调整窗口大小
<code>scrollBy()</code> , <code>scrollTo()</code>	滚动窗口中的文档
<code>setInterval()</code> , <code>clearInterval()</code>	设置或取消重复调用的函数, 该函数在两次调用之间有指定的延迟
<code>setTimeout()</code> , <code>clearTimeout()</code>	设置或取消在指定的若干秒后调用一次的函数

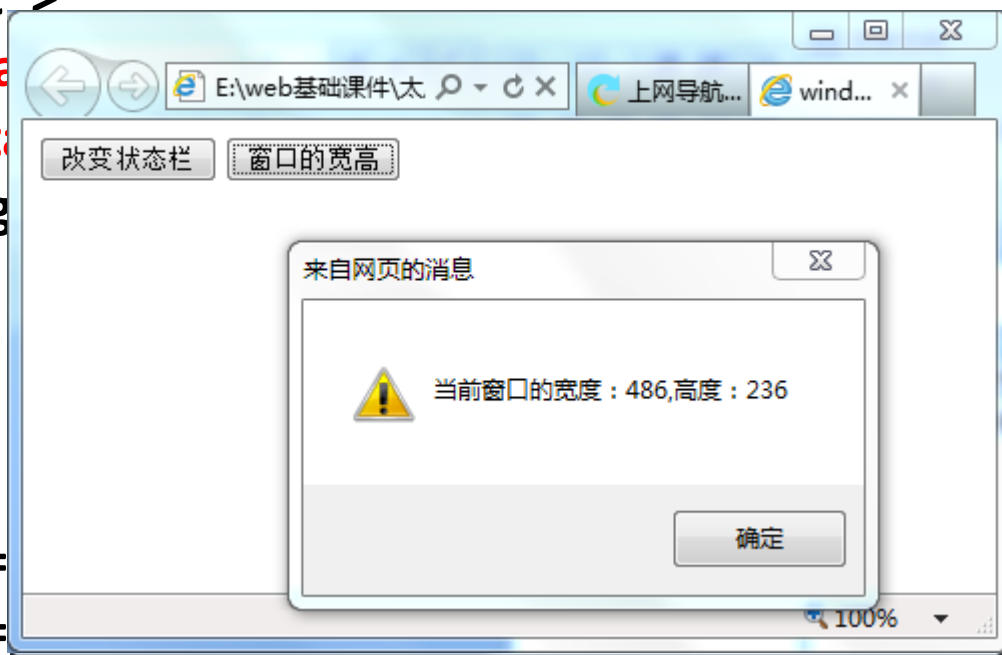


Window属性的应用案例

 [\[点击查看案例\]](#)



```
<!DOCTYPE html>
<html> <head>
  <title>window对象的属性</title>
  <script type="text/javascript">
    defaultStatus="欢迎来到ja
    function changeStatus(){st
    function getWidthAndHeig
    alert("当前窗口的宽度:
    "+window.innerWidth);}
  </script> </head>
  <body>
    <input type="button" value=
    <input type="button" value=
    onclick="getWidthAndHeight()" />
  </body></html>
```





Window方法的应用案例1

 [\[点击查看案例\]](#)



```
<!DOCTYPE html>
<html><head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>打开和关闭窗口的使用</title>
  <script language="javascript">
    var myWindow;
    function openNewWin(){
      myWindow=window.open("window属性.html","myWindow","width=200,height=150,top=200,left=100");
    }
    function closeNewWin(){
      myWindow.close();
    }
  </script></head>
  <body>
    <p><a href="javascript:openNewWin()">打开新窗口</a></p>
    <p><a href="javascript:closeNewWin()">关闭新窗口</a></p>
  </body>
</html>
```

open方法的格式: window.open(url,窗口名,窗口参数)



13.2 窗口 (window) 对象

window对象中**open**方法的参数还有很多，如下表所示：

参数	描述
top	窗口顶部离开屏幕顶部的像素数
left	窗口左端离开屏幕左边的像素数
width	窗口的宽度
height	窗口的高度
menubar	窗口是否有菜单，取值yes或no
toolbar	窗口是否有工具栏，取值yes或no
location	窗口是否有地址栏，取值yes或no
directories	窗口是否有链接区，取值yes或no
scrollbars	窗口是否有滚动条，取值yes或no
status	窗口是否有状态栏，取值yes或no
resizable	窗口是否可以调整大小，取值yes或no



13.2 窗口 (window) 对象

2. 对话框

对话框是跟用户进行交流的一种形式，对话框有提示、选择和获取信息的功能。**JavaScript**提供了三种标准的对话框，分别是弹出对话框、选择对话框和输入对话框。

方法	描述
alert()	弹出一个包含【确定】按钮的对话框
confirm()	弹出一个包含【确定】和【取消】按钮的对话框，如果单击【确定】，则返回 true ，如果单击【取消】，则返回 false
prompt()	弹出一个包含【确定】按钮、【取消】按钮和一个文本框的对话框，可以接收用户输入的信息。如果单击【确定】，则返回文本框已有内容，如果单击【取消】，则返回 null



【例13-4】 window对象三种对话框示例。



[\[点击查看案例\]](#)



```
<!doctype html>
<html><head><meta charset="utf-8">
<title>三种对话框</title>
<script>
function show_alert(){
    alert("弹出对话框");}
function show_prompt(){
    var name = prompt("请输入姓名","");
    if (name!=null && name!="")
    {
        document.write("你好 " + name + "!");}
    }
function show_confirm(){
    var c = confirm("请按下按钮");
    if (c == true){
        document.write("确定按钮");}
    else {document.write("取消按钮");}
    }
</script></head>
```

```
<body>
    <input type="button"
        onClick="show_alert()" value="弹出对话框"/>
    <input type="button"
        onClick="show_prompt()" value="弹出输入对话框"/>
    <input type="button"
        onClick="show_confirm()" value="弹出选择对话框"/>
</body>
</html>
```



13.3 文档 (document) 对象

1. 认识文档对象

document对象是window对象的子对象，是指在浏览器窗口中显示的内容部分。

document对象的方法

方法	描述
close()	关闭由open()方法打开的文档
open()	产生一个新文档，并覆盖已有的文档内容
write()	把文本附加到当前打开的文档中
writeln()	把文本输入到当前打开的文档，并附加一个换行符



13.3 文档 (document) 对象

document对象的属性

属 性	描 述
alinkColor	被激活的超链接的颜色
linkColor	未访问过的超链接的颜色
vlinkColor	访问过的超链接的颜色
cookie	设置或返回与当前稳当有关的所有 cookie
domain	返回当前文档的域名
lastModified	返回当前文档最后修改的日期和时间
referrer	返回载入当前文档的 URL
title	返回当前文档的标题
URL	返回当前文档的 URL
forms[]	form 对象的一个数组，返回文档中所有 form 对象的引用
images[]	image 对象的一个数组，返回文档中所有 image 对象的引用
links[]	link 对象的一个数组，返回文档中所有<a>（包括图片热点和 Link 超链接）标签的引用



document属性的应用案例

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document对象属性</title>
    <script language="javascript">
      document.write(document.domain+"<br>")
      document.write(document.title+"<br>")
      document.write(document.url+"<br>")
      document.write(document.lastModified+"<br>")
      function ChangeColor(){
        document.body.bgColor="blue"
      }
    </script>
  </head>
  <body onclick="ChangeColor()">
    单击文本改变背景色!
  </body>
</html>
```

 [\[点击查看案例\]](#)



【例13-6】 document对象的引用。

 [\[点击查看案例\]](#)

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document对象引用</title>
  </head>
  <body>
    <h2>在一个文本框输入内容，观察另一个文本框的变化： </h2>
    <form>
      第一个： <input type="text"
        onChange="document.my.elements[0].value=this.value;">
    </form>
    <form name="my">
      第二个： <input type="text"
        onChange="document.forms[0].elements[0].value=this.value;">
    </form>
  </body>
</html>
```

this: 指代当前对象



13.3 文档 (document) 对象

2. 文档中的图片

图像是网页文档中的常用对象，如果想对文档中的图像标签进行操作，可以通过document对象来操作，document对象提供了多种访问图像标签的方法。

1. 通过对象数组引用

`document.images` //对应页面上的标签

`document.images.length` //对应页面上标签的个数

`document.images[0]` //第一个标签

`document.images[n]` //第n+1个标签



通过数组对象引用图像的应用案例

```
<html> <head> <meta charset="utf-8"> <title>图像引用</title>
</head> <body>
  <img id="img1" name="img1">
  <img id="img2" name="img2">
  <img id="img3" name="img3">
  <img id="img4" name="img4">
  <script type="text/javascript">
    document.write("<br/>第一张图片的名字是:")
    document.write(document.images[0].name+"<br/>")
    document.write("第二张图片的名字是:")
    document.write(document.images[1].name+"<br/>")
    document.write("第三张图片的名字是:")
    document.write(document.images[2].name+"<br/>")
    document.write("第四张图片的名字是:")
    document.write(document.images[3].name+"<br/>")
    document.write("网页中图片数为:")
    document.write(document.images.length+"<br/>")
  </script>
</body></html>
```

 [\[点击查看案例\]](#)





13.3 文档 (document) 对象

2. 文档中的图片

2. 通过name属性值引用

```
<img name="myPic">  
<script>  
document.images.myPic  
</script>
```

3. 通过引用图片的src属性

```
  
<script>  
document.images.myPic.src  
</script>
```



【例13-7】 图片切换效果。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>文档中的图片</title>
<script>
function changeSrc()
{
    document.images.myImage.src="sky.jpg";
}
</script>
</head>
<body>

<br>
<input type="button" onClick="changeSrc()" value="切换图片">
</body>
</html>
```

 [\[点击查看案例\]](#)





13.3 文档 (document) 对象

3. 文档中的超链接

【例13-8】显示页面链接数目。

```
<!doctype html>
<html><head><meta charset="utf-8">
<title>显示当前页面链接数目</title>
</head>
<body id="top">
  <p><a href="www.baidu.com">百度</a>,
  <a href="www.1688.com">阿里巴巴</a>,
  <a href="www.qq.com">腾讯</a></p>
  <p>网易, 雅虎, 搜狐, 新浪</p>
  <p><a href="#top">返回顶部</a></p>
  <p>链接数目:
  <script>
document.write(document.links.length);
</script></p></body></html>
```

 [点击查看案例]





13.4 表单对象

1. Form对象

表单是页面元素的一种，隶属于document对象，表单元素也是以**数组形式**组织，根据表单在页面中的位置先后次序，依次排列形成构成表单数组。

表单的引用方法1：通过i数组来访问

document.forms[0]

表单的引用方法2：通过id值来访问

document.forms["MyForm"]

document.MyForm

表单的引用方法3：使用document对象的getElementById方法来访问

document.getElementById("MyForm")



【例13-9】显示表单元素的个数。

```
<!doctype html>
<html><head>
<meta charset="utf-8">
<title>form表单长度</title>
</head>
<body>
<form method="get" id="MyForm">
  名称: <input type="text" size="20"><br>
  密码: <input type="text" size="20">
  <input type="submit" value="登录">
</form>
<script>
document.write("表单中包含子元素的个数为: ");
document.write(document.getElementById("MyForm").length);
</script>
</body>
</html>
```

 [\[点击查看案例\]](#)



13.4 表单对象

2. Form对象属性与方法

form对象常用属性

属 性	描 述
action	设置或返回表单action属性
enctype	设置或返回表单用来编码内容的MIME类型
id	设置或返回表单的id
length	返回表单中元素数目
method	设置或返回将数据发送到服务器的HTTP方法
name	设置或返回表单的名称
target	设置或返回表单提交结果的Frame或Window名

form对象常用方法

方法	描 述
reset()	把表单的所有输入元素重置为它们的默认值
submit()	提交表单



13.4 表单对象

3. 单选按钮与复选框

【例13-10】 单选框与复选框的状态控制。

```
<!doctype html>
<html>
<head><meta charset="utf-8"><title>单选框与复选框</title>
<script>
    function check(){
        document.getElementById("check1").checked = true;
    }
    function uncheck(){
        document.getElementById("check1").checked = false;
    }
</script>
</head>
```



[点击查看案例]





[\[点击查看案例\]](#)



```
<body>
  <form>
<br><br>
    <input type="checkbox" id="check1">
    <input type="button" onClick="check()" value="选中复选框">
    <input type="button" onClick="uncheck()" value="取消复选框">
  </form>
</body>
</html>
```



13.4 表单对象

4. 下拉列表

【例13-11】显示下拉列表选中项目的索引值。

```
<!doctype html>
<html>
  <head><meta charset="utf-8">
    <title>下拉列表</title>
    <script>
      function getIndex(){
        var n = document.getElementById("MySelect");
        alert(n.selectedIndex);
      }
    </script>
  </head>
```

 [\[点击查看案例\]](#)



```
<body>
<form method="post">
  选择自己喜欢的饮料:
  <select id="MySelect">
    <option>咖啡</option>
    <option>红茶</option>
    <option>果汁</option>
    <option>矿泉水</option>
  </select>
</form>
<br><br>
<input type="button" value="显示选中项目索引"
  onClick="getIndex()">
</body>
</html>
```



下拉选项与文本框同步显示的案例



[点击查看案例]



```
<!doctype html>
<html> <head>
  <script language="javascript">
    function change(n){
      document.form1.txt1.value =document.form1.selectmenu.value;}
  </script>
</head>
<body>
  <form name="form1">
    <select name="selectmenu" onchange="change(this)">
      <option value="图1.gif" selected>图1</option>
      <option value="图2.gif">图2</option>
      <option value="图3.gif">图3</option>
      <option value="图4.jpg">图4</option>
    </select>
    <input type="text" name="txt1" value="图1.gif">
  </form> </body></html>
```



链接下列地址的案例

```
<!doctype html>
```

```
<html> <head>
```

```
  <script language="javascript">
```

```
    function change(n){
```

```
      document.form1.txt1.value =document.form1.selectmenu.value;}
```

```
    function submit1(){
```

```
      var url =document.form1.selectmenu.value;
```

```
      location.href =url; }
```

```
  </script> </head>
```

```
<body>
```

```
  <form name="form1">
```

```
    <select name="selectmenu" onchange="change(this)">
```

```
      <option value="图1.gif" selected>图1</option>
```

```
      <option value="https://www.baidu.com">图2</option>
```

```
      <option value="图3.gif">图3</option>
```

```
      <option value="图4.jpg">图4</option>
```

```
    </select>
```

```
    <input type="text" name="txt1" value="图1.gif">
```

```
    <input type="button" value="确定" OnClick="submit1()">
```

```
  </form></body></html>
```

 [\[点击查看案例\]](#)



location对象可按指定URL打开新页面



13.4 表单对象

5. 表单注册与表单验证

【例13-12】 创建表单注册页面，用户填写信息：用户名、性别、密码、密码问题、密码答案、电子邮件。并在注册页面提交前进行验证，除“密码问题、密码答案”以外的项目都要进行验证，如果发现问题，弹出对应的提示信息提醒用户重新修改。

请自学吧！

 [\[点击查看案例\]](#)





本章结束