**浏览器端JavaScript编程**

# 3 浏览器端JS对象

## 3.1 BOM和DOM

**BOM Browser Object Model** 浏览器对象模型，浏览器的很多行为都可以通过JavaScript控制，例如打开窗口、关闭窗口、把网页设为主页或添加到收藏夹等。这些操作实际上就是对BOM进行操作。BOM就是由一系列独立于内容而与浏览器窗口窗口进行交互的对象构成，也就是说，浏览器上见到的一切都属于BOM。

包括如下几个对象：

window

document

location

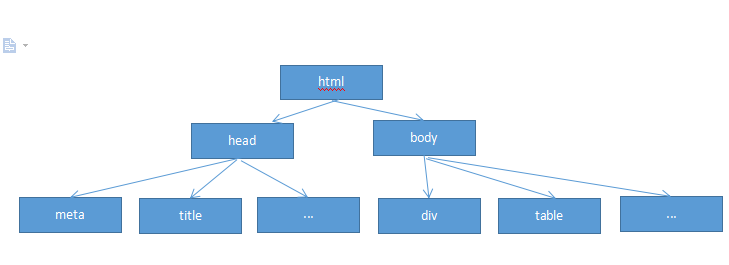
screen

navigator

history

**DOM Document Object Model** 文档对象模型，HTML页面中的内容加载到内存，在内存中形成一个DOM对象。DOM用来操作页面上的内容。

DOM本质上就是BOM中的document。





## 3.2 window

window对象表示整个浏览器窗体（在使用页面框架时，每个frame都是一个window）

**window对象的常用操作：在使用时，window关键字可以省略**

**1 定时操作：**

**setTimeout() clearTimeout()**

var st = window.setTimeout(function(){},2000);

含义：2000毫秒后，执行function一次。

clearTimeout(st); 停止定时操作。

还可以写成： function fun(){}

window.setTimeout(fun,2000) 或 window.setTimeout(“fun()”,2000)

**setInterval() clearInterval()**

var ss = window.setInterval(function(){},2000);

含义：每隔2000毫秒，执行function一次。

window.clearInterval(ss); 停止定时操作。

也可写成：

function fun(){}

window.setInterval(fun,2000) 或 window.setInterval(“fun()”,2000)

**案例： 利用setInterval函数，制作一个秒表。点击开始按钮开始运行，点击暂停按钮暂停。**

|  |
| --- |
| <title>无标题文档</title>  <script>  var si;  function begin(){  si = window.setInterval(function(){  var date = new Date();  document.getElementById("hh").innerHTML=date.toLocaleString();  },1000);  }  function pause(){    window.clearInterval(si);  }  </script>  </head>  <body>  <input type="button" value=" 开始 " onclick="begin();" />  <input type="button" value=" 暂停 " onclick="pause();" />  <h1 id="hh"></h1>  </body> |

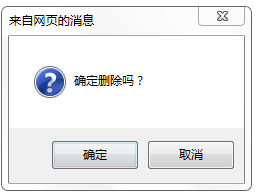
**2 对话框操作**

**alert();** 弹出一个警告框。



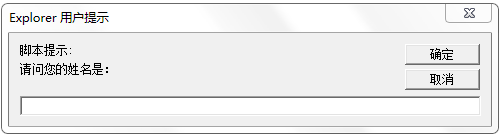
**confirm(“提示内容”)**

|  |
| --- |
| var flag = confirm(“确定删除吗？”);  点击确定时，flag值为true。点击取消时，flag值为false |



**prompt(“文本内容”)**

|  |
| --- |
| var result = prompt(“请问您的姓名是：”)  result为用户填写的值。  var result = prompt(“请问您的姓名是：”，“默认值”)；  两个参数时，第2个参数是默认值。 |



**3 改变窗体大小操作（不是所有浏览器都支持）**

**resizeTo(x,y)**

resizeTo(500,500) 改变窗体大小为500,500

**4 移动窗体位置（不是所有浏览器都支持）**

**moveTo(x,y)**

moveTo(100,200)

**5 打开新窗体**

open(“”，“”，“”，true/false)

|  |
| --- |
| var newWin=  window.open("http://www.baidu.com","newWin","width=200,height=200,menubar=yes,location=yes,resizable=yes");  **参数说明：**  第1个参数： 打开新窗口的链接  第2个参数：新窗口名字，可以在<a>的target中使用  第3个参数：新窗口特征（宽度，高度，是否有菜单栏，是否有地址栏，是否可以重置大小......）  第3个参数中可以设置的值  height  width  left  top  menubar  location  status  resizable  scrollbars  toolbar  fullscreen  titlebar  第4个参数 如果在同一窗口打开新窗口，该值用于指定是否在history中替换原窗口的url链接，可取值：true/false |

在新打开的窗口中如何拿到父窗口（打开它的窗口）？**使用 window.opener**

在父窗口中如何拿到新打开的窗口？ **利用open方法的返回值**

**6 关闭窗口**

**window.close(); (很多浏览器不支持)**

**7 window.onload=function(){} 页面加载完成后执行function。**

**window.onunload = function(){ } 页面卸载之前执行这个函数，但现在的浏览器基本都屏蔽了这个方法。**

## 3.3 history

可以获得浏览器访问历史

history.back() 后退一步

history.forward() 前进一步

history.go(number) number为整数时，表示前进。为负数时，表示后退。

## 3.4 navigator

**只读对象，**包含有关浏览器的信息

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | [appCodeName](http://www.w3school.com.cn/jsref/prop_nav_appcodename.asp) | 返回浏览器的代码名。 | | [appMinorVersion](http://www.w3school.com.cn/jsref/prop_nav_appminorversion.asp) | 返回浏览器的次级版本。 | | [appName](http://www.w3school.com.cn/jsref/prop_nav_appname.asp) | 返回浏览器的名称。 | | [appVersion](http://www.w3school.com.cn/jsref/prop_nav_appversion.asp) | 返回浏览器的平台和版本信息。 | | [browserLanguage](http://www.w3school.com.cn/jsref/prop_nav_browserlanguage.asp) | 返回当前浏览器的语言。 | | [cookieEnabled](http://www.w3school.com.cn/jsref/prop_nav_cookieenabled.asp) | 返回指明浏览器中是否启用 cookie 的布尔值。 | | [cpuClass](http://www.w3school.com.cn/jsref/prop_nav_cpuclass.asp) | 返回浏览器系统的 CPU 等级。 | | [onLine](http://www.w3school.com.cn/jsref/prop_nav_online.asp) | 返回指明系统是否处于脱机模式的布尔值。 | | [platform](http://www.w3school.com.cn/jsref/prop_nav_platform.asp) | 返回运行浏览器的操作系统平台。 | | [systemLanguage](http://www.w3school.com.cn/jsref/prop_nav_systemlanguage.asp) | 返回 OS 使用的默认语言。 | | [userAgent](http://www.w3school.com.cn/jsref/prop_nav_useragent.asp) | 返回由客户机发送服务器的 user-agent 头部的值。 | | [userLanguage](http://www.w3school.com.cn/jsref/prop_nav_userlanguage.asp) | 返回 OS 的自然语言设置。 | |

|  |
| --- |
| var obj = window.navigator;  for(key in obj){  alert(key+">>>"+obj[key]);  } |

## 3.5 screen

Screen 对象包含有关客户端显示屏幕的信息

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。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | [availHeight](http://www.w3school.com.cn/jsref/prop_screen_availheight.asp) | 返回显示屏幕的高度 (除 Windows 任务栏之外)。 | | [availWidth](http://www.w3school.com.cn/jsref/prop_screen_availwidth.asp) | 返回显示屏幕的宽度 (除 Windows 任务栏之外)。 | | [bufferDepth](http://www.w3school.com.cn/jsref/prop_screen_bufferdepth.asp) | 设置或返回调色板的比特深度。 | | [colorDepth](http://www.w3school.com.cn/jsref/prop_screen_colordepth.asp) | 返回目标设备或缓冲器上的调色板的比特深度。 | | [deviceXDPI](http://www.w3school.com.cn/jsref/prop_screen_devicexdpi.asp) | 返回显示屏幕的每英寸水平点数。 | | [deviceYDPI](http://www.w3school.com.cn/jsref/prop_screen_deviceydpi.asp) | 返回显示屏幕的每英寸垂直点数。 | | [fontSmoothingEnabled](http://www.w3school.com.cn/jsref/prop_screen_fontsmoothingenabled.asp) | 返回用户是否在显示控制面板中启用了字体平滑。 | | [height](http://www.w3school.com.cn/jsref/prop_screen_height.asp) | 返回显示屏幕的高度。 | | [logicalXDPI](http://www.w3school.com.cn/jsref/prop_screen_logicalxdpi.asp) | 返回显示屏幕每英寸的水平方向的常规点数。 | | [logicalYDPI](http://www.w3school.com.cn/jsref/prop_screen_logicalydpi.asp) | 返回显示屏幕每英寸的垂直方向的常规点数。 | | [pixelDepth](http://www.w3school.com.cn/jsref/prop_screen_pixeldepth.asp) | 返回显示屏幕的颜色分辨率（比特每像素）。 | | [updateInterval](http://www.w3school.com.cn/jsref/prop_screen_updateinterval.asp) | 设置或返回屏幕的刷新率。 | | [width](http://www.w3school.com.cn/jsref/prop_screen_width.asp) | 返回显示器屏幕的宽度。 | |

## 3.6 location

|  |
| --- |
| host www.zparkedu.com:1234  hostname www. zparkedu.com  **href**  pathname  port  protocol  search |

**window.location.href=””**

## 3.7 document

**最常用对象。**

html页面加载到内存中形成document对象。

### 3.7.1 document.write

**document.write(“”)**

1 如果在页面加载完成前使用，将是插入到当前页面中。

2 如果在页面加载完成后使用，将会覆盖原来页面中信息。

**3 不建议使用。**

### 3.7.2 获得页面上元素的集合(数组结构)

|  |
| --- |
| document.anchors 页面上的所有超链接 (要求超链接必须有id属性(IE),火狐要求必须有name属性)  document.applets  document.embeds  document.forms  document.frames  document.images  document.plugins  document.scripts  document.stylesheets |

### 3.7.3 document.all

获得页面上的所有元素。不是w3c的标准，只有IE浏览器支持，现在大部分浏览器也支持。

### 3.7.3 根据元素属性获得元素

|  |
| --- |
| **document.getElementById(“标签id属性值”) 非常常用！！！**  document.getElementsByName(name) 根据name获得元素，返回数组  document.getElementsByTagName(tagName) 根据标签名获得元素，返回数组  node.getElementsByTagName(tagName) 根据标签名获得元素，返回数组  document.getElementsByClassName()（IE8以前没有支持）  document.head  document.body |

**全选、反选、全不选**

|  |
| --- |
| <input type="checkbox" value="游戏" name="hobby" />游戏  <input type="checkbox" value="音乐" name="hobby" />音乐  <input type="checkbox" value="体育" name="hobby" />体育  <input type="button" value="全选" onclick="quanxuan();" />  <input type="button" value="全不选" onclick="quanbuxuan();" />  <input type="button" value="反选" onclick="fanxuan();" />  function quanxuan(){  var cs = document.getElementsByName("hobby");  for(var i=0;i<cs.length;i++){  //alert(cs[i].value)  //cs[i].checked="checked";  cs[i].checked=true;    }  }    function quanbuxuan(){  var cs = document.getElementsByName("hobby");  for(var i=0;i<cs.length;i++){  //alert(cs[i].value)  //cs[i].checked="";  cs[i].checked=false;  }  }  function fanxuan(){  var cs = document.getElementsByName("hobby");  for(var i=0;i<cs.length;i++){  //alert(cs[i].checked);  cs[i].checked = (!cs[i].checked);  }  } |

### 3.7.5 访问属性

使用前面的各种方式获得页面的元素后，需要访问这些元素的属性。

例如：图片有如下属性

|  |
| --- |
| <img src="aaa.jpg" id="img" />  <script>  document.getElementById('img').src=""  </script>  align 'none', 'top', 'left', ...  alt string  border integer (pixels)  height integer (pixels)  hspace integer (pixels)  id string  isMap boolean  src url  useMap url  vspace integer (pixels)  width |

**访问元素属性的两个作用：**

1 获得属性值

var value = node.property;

var name = document.getElementById(‘username’).value;

2 给元素的某个属性赋值

node.property=value;

document.getElementById(‘username’).value=’张三’

**以上的方式称为oldSchool，现在又提供了新方式，称为newSchool**

取值：

document.getElementById(‘username’).getAttribute(‘value’)

赋值：

document.getElementById(‘username’).setAttribute(‘value’，‘张三’)

### 3.7.6 使用js操作元素的css样式

node.className

node.style.stylename

|  |
| --- |
| **css属性 javascript写法**  background-color backgroundColor  font-size fontSize  z-index zIndex  color color  **规则： 一个单词的css属性，在js中写法相同。多个单词的css属性css中使用 - js中使用驼峰形式。** |

**案例：改变<textarea>中内容的样式**

|  |
| --- |
| <title>无标题文档</title>  <style>  .aaa{  width:500px;  height:300px;  }  .bbb{  width:1000px;  height:600px;  }  </style>  <script>  function f1(){  document.getElementById("tt").style.backgroundColor="red";  }  function f2(){  document.getElementById("tt").style.color="blue";  }  function f3(){  document.getElementById("tt").style.fontSize="30px";  }  function f4(){  document.getElementById("tt").style.fontWeight=900;  }  function f5(){  document.getElementById("tt").style.fontFamily="隶书";  }  function f6(){  document.getElementById("tt").className="bbb";  }  function showImg(){  document.getElementById("img").style.display="inline";  }  function hideImg(){  document.getElementById("img").style.display="none";  }  </script>  </head>  <body>  <input type="button" value="改变背景颜色为红色" onclick="f1()" />  <input type="button" value="改变字体颜色" onclick="f2()" />  <input type="button" value="改变字体大小" onclick="f3()" />  <input type="button" value="改变字体粗细" onclick="f4()" />  <input type="button" value="改字体为隶书" onclick="f5()" />  <input type="button" value="改变文本框大小" onclick="f6()" />  <hr/>  <textarea id="tt" class="aaa">默认的值...</textarea>  <hr/>  <input type="button" value="显示图片" onclick="showImg()" />  <input type="button" value="隐藏图片" onclick="hideImg()" />  <img src="img-11.jpg" id="img" style="display:none" /> |

### 3.7.8 innerHTML

node.innerHTML // 取值

node.innerHTML=""; //赋值

node.innerText 获得文本内容（火狐不支持）

不是w3c标准，但是现在被大多数浏览器支持，**很常用**

|  |
| --- |
| <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  <script>  function getInnerHTML(){  var div = document.getElementById("dd1");  var str = div.innerHTML;  var str2 = div.innerText;  alert(str);  alert(str2);  }  function setInnerHTML(){    var str = "<select><option value='beijing'>北京</option><option value='shanghai'>上海</option><option value='guangzhou' selected>广州</option></select>"  //document.getElementById("dd1").innerHTML=str;  document.getElementById("dd1").innerHTML+=str;  }  </script>  </head>  <body>  <div id="dd1">  <textarea cols="30" rows="10">多行文本框中的值...</textarea>  div中的内容！！！  </div>  <input type="button" value="获得标签中的内容" onclick="getInnerHTML()" />  <input type="button" value="向标签中添加内容" onclick="setInnerHTML()" />  </body> |

# 4 事件

|  |
| --- |
| onclick  ondblclick  onmousedown  onmouseup  onmousemove  onmouseout  onmouseover  onblur  onchange  onfocus  onkeydown 键盘上所有按键都会触发  onkeypress 能输入内容的按键才会触发  onkeyup  onreset <form>的事件 onreset=”return fun()” 当fun返回fasle时，不能重置表单  onsubmit <form>的事件 onsubmit=”return fun()” 当fun返回fasle时，不能提交表单  onreset 和 onsubmit 相当于 重置按钮和提交按钮的 onclick事件 |

**添加事件：**

**W3C标准方式：**

**node.addEventListener(type, f, false);**

**IE方式：**

**node.attachEvent("on" + type, f);**

**传统方式（也是最常用方式）：**

**node["on" + type] = f;**

|  |
| --- |
| <input type=”button” id=’btn’value=”OKOK”/>  <script>  document.getElementById(‘btn’).onclick=function(){  alert(‘111’);  }  </script>  --------------------------------------------------------------------------  <script>  function fun(){  alert(‘111’);  }  </script>  <input type=”button” id=’btn’value=”OKOK”**onclick=”fun()”**/> |

**超链接调用js函数**

|  |
| --- |
| <a onclick="fun3()" style="color:blue;text-decoration:underline;cursor:pointer">超链接</a>    <a onclick="fun3()" href="#">超链接</a>    <a href="javascript:fun3()">超链接</a> |

**调用时传参数**

|  |
| --- |
| <script>  function fun(s){  alert(s);  }  </script>  <input type=”button” id=’btn’value=”OKOK”**onclick=”fun(‘xxx’)”**/> |

|  |
| --- |
| <script>  function fun(s){  Alert(s.value);  }  </script>  <input type=”text” id=’username’**onblur=”fun(this)”**/>  **this表示把当前标签对象传人function中** |

**事件源**

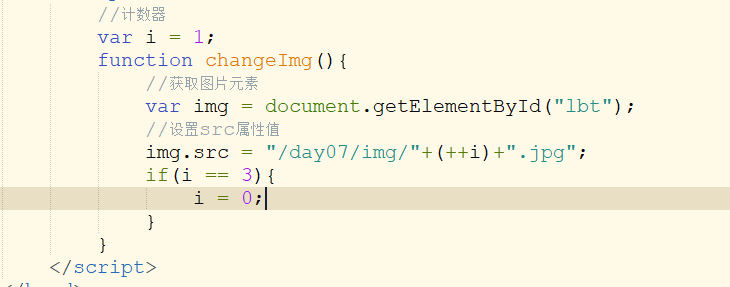
|  |
| --- |
| function (e) {  e = e || event;  var target =  e.target || e.srcElement;  ...  } |

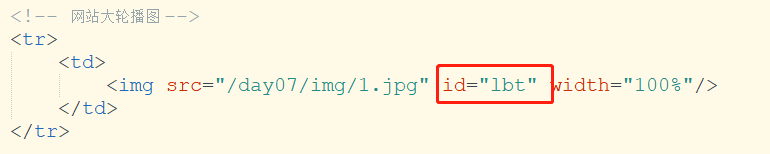
|  |
| --- |
| <script>  function fun(e){  e = e || event;  var target = e.target || e.srcElement;  alert(target.innerHTML);  }  </script>  <div onclick=”fun(event)”>  <div>AAA</div>  <div>BBB</div>  <div>CCC</div>  <div>DDD</div>  </div> |

# 5案例：

## 5.1 表单验证

## 5.2 图片轮播





## 5.3 定时弹广告

* 网站顶部在页面打开两秒后显示广告图；
* 广告图展示两秒后关闭广告图；

