BOM

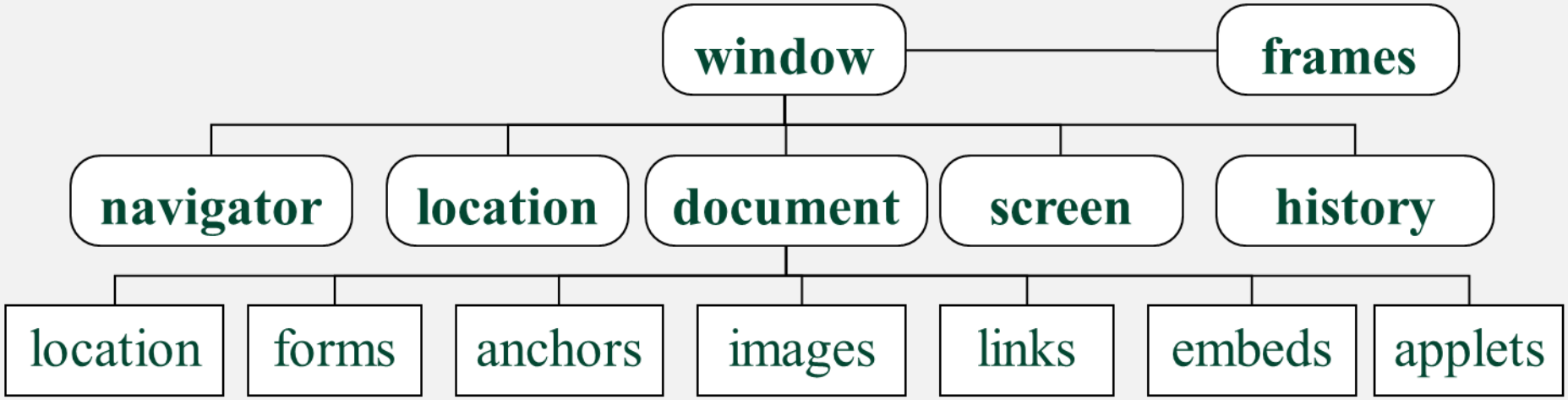
管理科学与工程学科
耿方方

主要内容

- Window对象
- location对象
- navigator对象
- screen对象
- history对象

JavaScript是运行在浏览器中的，因此提供了一系列对象用于与浏览器窗口进行交互。这些功能与任何网页内容无关。

这些对象主要有：window、document、location、navigator和screen等，把它们统称为BOM（Browser Object Model，浏览器对象模型）。



- BOM中定义了6种重要的对象：
- window对象表示浏览器打开的窗口；
- document对象表示浏览器中加载页面的文档对象；
- location对象包含了浏览器当前的URL信息；
- navigator对象包含了浏览器本身的信息；
- screen对象包含了客户端屏幕及渲染能力的信息；
- history对象包含了浏览器访问网页的历史信息。

window对象

- Window对象表示整个浏览器窗口，但不包括其中的页面内容。
- Window对象可以打开窗口与关闭窗口，控制窗口的大小和位置，由窗口弹出对话框，还可以控制窗口上是否显示地址栏、工具栏和状态栏等栏目。对于窗口中的内容，Window对象可以控制是否重载网页、返回上一个文档或前进到下一个文档。

window对象

window对象属性

属性	描述
closed	返回窗口是否已被关闭。
defaultStatus	设置或返回窗口状态栏中的默认文本
innerheight	返回窗口的文档显示区的高度
innerwidth	返回窗口的文档显示区的宽度
length	设置或返回窗口中的框架数
name	设置或返回窗口的名字
pageXOffset	设置或返回当前页面相对于窗口显示区左上角X位置
pageYOffset	设置或返回当前页面相对于窗口显示区左上角Y位置
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。

window对象

window对象方法

方　法	描　述
alert()	弹出一个警告对话框
confirm()	在确认对话框中显示指定的字符串
prompt()	弹出一个提示对话框
open()	打开新浏览器对话框并且显示由URL或名字引用的文档，并设置创建对话框的属性
close()	关闭被引用的对话框
focus()	将被引用的对话框放在所有打开对话框的前面
blur()	将被引用的对话框放在所有打开对话框的后面
scrollTo(x,y)	把对话框滚动到指定的坐标
scrollBy(offsetx,offsety)	按照指定的位移量滚动对话框
setTimeout(timer)	在指定的毫秒数过后，对传递的表达式求值
setInterval(interval)	指定周期性执行代码
moveTo(x,y)	将对话框移动到指定坐标处
moveBy(offsetx,offsety)	将对话框移动到指定的位移量处
resizeTo(x,y)	设置对话框的大小
resizeBy(offsetx,offsety)	按照指定的位移量设置对话框的大小
print()	相当于浏览器工具栏中的“打印”按钮
navigate(URL)	使用对话框显示URL指定的页面
status()	状态条，位于对话框下部的信息条
Defaultstatus()	状态条，位于对话框下部的信息条

window对象

- window对象可以直接调用其方法和属性，例如：
- window.属性名
- window.方法名

window对象

- open() 方法用于打开一个新的浏览器窗口或查找一个已命名的窗口。
- 语法：
- window.open(URL,name,features,replace)
- URL:声明了要在新窗口中显示的文档的 URL。如果省略了这个参数，或者它的值是空字符串，那么新窗口就不会显示任何文档。
- name:该字符声明了新窗口的名称。
- features:声明了新窗口要显示的标准浏览器的特征。如果省略该参数，新窗口将具有所有标准特征。
- Replace:一个可选的布尔值。规定了装载到窗口的 URL 是在窗口的浏览历史中创建一个新条目，还是替换浏览历史中的当前条目。支持下面的值：

  true - URL 替换浏览历史中的当前条目。false - URL 在浏览历史中创建新的条目。

window对象

打开与关闭窗口

窗口特征（Window Features）

channelmode=yes|no|1|0	是否使用剧院模式显示窗口。默认为 no。
directories=yes|no|1|0	是否添加目录按钮。默认为 yes。
fullscreen=yes|no|1|0	是否使用全屏模式显示浏览器。默认是 no。处于全屏模式的窗口必须同时处于剧院模式。
height=pixels	窗口文档显示区的高度。以像素计。
left=pixels	窗口的 x 坐标。以像素计。
location=yes|no|1|0	是否显示地址字段。默认是 yes。
menubar=yes|no|1|0	是否显示菜单栏。默认是 yes。
resizable=yes|no|1|0	窗口是否可调节尺寸。默认是 yes。
scrollbars=yes|no|1|0	是否显示滚动条。默认是 yes。
status=yes|no|1|0	是否添加状态栏。默认是 yes。
titlebar=yes|no|1|0	是否显示标题栏。默认是 yes。
toolbar=yes|no|1|0	是否显示浏览器的工具栏。默认是 yes。
top=pixels	窗口的 y 坐标。
width=pixels	窗口的文档显示区的宽度。以像素计。

window对象

打开与关闭窗口

- 实例2-01:
- <html>
- <head>
- <title>打开新窗口</title>
- <meta charset="UTF-8">
- </head>
- <body>
- <img src="Images/img01.jpg" width="600" height="400">
- <script>

```
window.open("Demo5-01.html","new","height=140,width=300,top=100,left=200");
```

- </script>
- </body>
- </html>

window对象

打开与关闭窗口

- 实例2-02:
- <body>
- <script>
- function openWindow(){
- window.open("Demo5-01.html","new","toolbar,menubar,scrollbars,status,location,height=200,width=200");
- }
- </script>
- <form name="form1" method="post" action="">
- <input type="button" name="Button" value="创建新窗口" onclick="openWindow()">
- </form>
- </body>

window对象

- 关闭窗口
- 利用window对象的close ()方法可以实现关闭当前窗口的功能。
- 语法:
- window.close()
- 例如 2-03: 关闭子窗口
- <form name="form1" method="post" action="">
- <input type="button" name="Button" value="关闭子窗口" onclick="newClose()">
- </form>
- <script>
- var win;
- win=window.open("Demo5-01.html","new","width=300,height=200");
- function newClose(){
- win.close();
- }
- </script>

window对象

对话框

- 浏览器通过alert()、confirm()和prompt()方法可以调用系统对话框向用户显示消息。
- 警告框：警告框经常用于确保用户可以得到某些信息。当警告框出现后，用户需要点击确定按钮才能继续进行操作。
- window.alert() 方法可以不带上window对象，直接使用alert()方法。

window对象

实例3:

```
<html>
    <head>
        <title>警告框</title>
        <meta charset="UTF-8">
<script>
function myFunction(){
    alert("你好，我是一个警告框！");
}
</script>
    </head>
    <body>
        <div>警告框</div>
        <input type="button" onclick="myFunction()" value="显示警告框" />
    </body>
</html>
```

window对象

对话框

- 确认框
- 确认框通常用于验证是否接受用户操作。
- 当确认卡弹出时，用户可以点击“确认”或者“取消”来确定用户操作。
- 当你点击“确认”，确认框返回 true， 如果点击“取消”，确认框返回 false。
- window.confirm() 方法可以不带上window对象，直接使用confirm()方法。

window对象

对话框

- 实例4:
- <html>
- <head>
- <title>确认框</title>
- <meta charset="UTF-8">
- </head>
- <body>
- 点击按钮，显示确认框。
- <button onclick="myFunction()"> 点 我 </button>
-
- <script>
- function myFunction(){
- var x;
-

```
var r=confirm("按下按钮!");
            if (r==true){
              x="你按下了\"确定\"按钮!";
            }
            else{
              x="你按下了\"取消\"按钮!";
            }
            document.getElementById("
demo").innerHTML=x;
}
</script>
  </body>
</html>
```

window对象

对话框

- 提示框

  提示框经常用于提示用户在进入页面前输入某个值。

- 当提示框出现后，用户需要输入某个值，然后点击确认或取消按钮才能继续操纵。
- 如果用户点击确认，那么返回值为输入的值。如果用户点击取消，那么返回值为 null。
- window.prompt(“str1”,“str2”);
- Str1:指定在对话框内要被显示的信息。
- Str2: 指定对话框内输入文本框的值，如果忽略此参数，将被设置为undefined。

window对象

对话框

- 实例5:
- <html>
- <head>
- <title>TODO supply a title</title>
- <meta charset="UTF-8">
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- </head>
- <body>
- 点击按钮查看输入的对话框。
- <button onclick="myFunction()">点我</button>
-

```
<script>
function myFunction(){
        var x;
        var person=prompt("请输入你的名字","Harry Potter");
        if (person!=null && person!=""){
            x="你好 " + person + " ！今天感觉如何？";

document.getElementById("demo").innerHTML=x;
        }
}
</script>
  </body>
</html>
```

window对象

窗口大小和位置

- 窗口大小

有三种方法能够确定浏览器窗口的尺寸（浏览器的窗口，不包括工具栏和滚动条）。

对于Internet Explorer、Chrome、Firefox、Opera 以及 Safari:

•window.innerHeight - 浏览器窗口的内部高度

•window.innerWidth - 浏览器窗口的内部宽度

对于 Internet Explorer 8、7、6、5:

•document.documentElement.clientHeight

•document.documentElement.clientWidth

或者

•document.body.clientHeight

•document.body.clientWidth

window对象

窗口大小和位置

- 实例1：

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
</head>
  <body>
    <p id="demo"></p>
<script>
var w=window.innerWidth||
document.documentElement.clie
ntWidth||
document.body.clientWidth;
var h=window.innerHeight
|| document.documentElement.clientHeight
|| document.body.clientHeight;
x=document.getElementById("demo");
x.innerHTML="浏览器的内部窗口宽度：" + w
+ "，高度：" + h + "。"
</script>
  </body>
</html>
```

window对象

窗口大小和位置

- 使用resizeTo()和resizeBy()方法可以调整浏览器窗口的大小。这两个方法都接受两个参数，其中resizeTo()接收浏览器窗口的新宽度和新高度，而resizeBy()接收新窗口与原窗口的宽度和高度之差。
- 例如：window.resizeTo(100,100)
- window.resizeBy(100,50)
- 注意：这两个方法可能被浏览器禁用，在Opera和IE7以上的版本中就是默认禁用的。

window对象

窗口滚动

- 使用window对象的scroll()方法可以指定窗口的当前位置，从而实现窗口滚动效果。
- 语法：
- scroll(x,y);
- 共有3种方法可以用来滚动窗口中的文档，这3种方法使用如下：
- Window.scroll(x,y)：该方法可以将窗口滚动到指定的绝对位置。
- Window.scrollTo(x,y)：与Window.scroll(x,y)相同。
- Window.scrollBy(x,y)：该方法可以将文档滚动到指定的相对位置。

window对象

窗口滚动

- 实例7-02:
- <script language="JavaScript">
- var position = 0;
- function scroller(){
- if (true){
- position++;
- scroll(0,position);
- var timer = setTimeout("scroller()",10);
- }
- }
- scroller();
- </script>

window对象

- 窗口位置
- 使用window.moveBy(dx, dy)和window.moveTo(x, y)可以移动窗口的位置。
- window.moveBy(dx, dy)
- 该方法将浏览器窗口相对于当前的位置移动指定的距离（相对定位），当dx和dy为负数时则向反方向移动。
- window.moveTo(x, y)
- 该方法将浏览器窗口移动到屏幕指定的位置（x、y处）（绝对定位）。同样可使用负数，只不过这样会把窗口移出屏幕。

window对象

窗口的大小与位置

- 实例2：

```
<html>
    <head>
        <title>弹出窗口及移动窗口</title>
        <meta charset="UTF-8">
       <script type="text/javascript">
        function moveWin()
        {
        myWindow.moveBy(50,50);
        myWindow.focus();
        }
       </script>
    </head>
 <body>
      <div>TODO write content</div>
      <script type="text/javascript">

myWindow=window.open('','','width=200,height=100');
      myWindow.document.write("This is 'myWindow'");
      </script>
<input type="button" value="Move 'myWindow'" onclick="moveWin()" />
  </body>
</html>
```

window对象

定时操作

- 通过使用 JavaScript，在一个设定的时间间隔之后来执行代码，而不是在函数被调用后立即执行。我们称之为计时事件。

- 在 JavaScritp 中使用计时事件是很容易的，两个关键方法是：
- setInterval() - 间隔指定的毫秒数不停地执行指定的代码。
- setTimeout() - 暂停指定的毫秒数后执行指定的代码。

window对象

定时操作

- setInterval() - 间隔指定的毫秒数不停地执行指定的代码。
- window.setInterval("function",milliseconds);
- 第一个参数是要调用的函数，第二个是间隔的毫秒数

- clearInterval() 方法用于停止 setInterval() 方法执行的函数代码。
- window.clearInterval(intervalVariable)

window对象

定时操作

- 实例6:
- <html>
- <head>
- <title>在页面显示一个时钟</title>
- <meta charset="UTF-8">
- </head>
- <body>
- 在页面显示一个时钟
-
- <button onclick="myStopFunction()"> 停 止 时 钟 </button>
- <script>
- var myVar=setInterval(function(){myTimer()},1000);
- function myTimer(){
- var d=new Date();
- var t=d.toLocaleTimeString();
- document.getElementById("demo").innerHTML=t;}

```
function myStopFunction(){
        clearInterval(myVar);
}
</script>
  </body>
</html>
```

window对象

定时操作

- setTimeout() - 暂停指定的毫秒数后执行指定的代码。
- window.setTimeout("javascript 函数",毫秒数);
- clearTimeout() 方法用于停止执行setTimeout()方法的函数代码。
- window.clearTimeout(timeoutVariable)

window对象

定时操作

- 实例7:
- <html>
- <head>
- <title>计时函数</title>
- <meta charset="UTF-8">
- </head>
- <body>
- 点击第一个按钮等待3秒后出现"Hello"弹框。
- 点击第二个按钮来阻止第一个函数运行。（你必须在3秒之前点击它）。
- <button onclick="myFunction()">点我</button>
- <button onclick="myStopFunction()">停止弹框</button>
- <script>

```
var myVar;
function myFunction(){
        myVar=setTimeout(function(){alert("Hello")},3000);
}
function myStopFunction(){
        clearTimeout(myVar);
}
</script>
  </body>
</html>
```

window对象

- 实例7-01：
- <body>
- <script language="JavaScript">
- window.resizeTo(300,300);
- window.moveTo(0,0);
- inter=setInterval("go()", 10);
- var aa=0;
- var bb=0;
- var a=0;
- var b=0;

```
function go(){
        try{
                if (aa==0)
                        a=a+2;
                if (a>screen.availWidth-300)
                        aa=1;
                if (aa==1)
                        a=a-2;
                if (a==0)
                        aa=0;
                if (bb==0)
                        b=b+2;
                if (b>screen.availHeight-300)
                        bb=1;
                if (bb==1)
                        b=b-2;
                if (b==0)
                        bb=0;
                window.moveTo(a,b);
        }catch(e){}
}
</script>
会自动移动的窗口
  </body>
```

window对象

- 状态栏控制（status属性）
- 浏览器状态的显示信息可以通过window.status属性直接进行修改。
- 例如：window.status=″看看状态栏中的文字变化了吗？″;

location对象

- window.location 对象用于获得当前页面的地址 (URL)，并把浏览器重定向到新的页面。

属性名	例子	说明
Hash	#contents	返回URL中的hash，如果URL中不包括散列，则返回空字符串
host	www.wrox.com:80	返回服务器名称及端口号
hostname	www.wrox.com	返回服务器名称
href	http://www.wrox.com	返回当前加载页面完整的URL
pathname	/wileyCDA/	返回目录和文件名
port	8080	返回端口号
protocol	http	返回页面使用的协议
search	? q=javascript	返回查询的字符串，以问号开头

location对象

- window.location.assign() 方法加载新的文档。
- 如果不希望用户可以用“后退”按钮返回原来的页面，可以使用replace()方法，该方法也能转到指定的页面，但不能返回到原来的页面了，这常用在注册成功后禁止用户后退到填写注册资料的页面。例如:
- 访问搜狐
- 可以发现转到新页面后，“后退”按钮是灰色的了

location对象

- 案例8:
- <html>
- <head>
- <title>TODO supply a title</title>
- <meta charset="UTF-8">
- </head>
- <body>
- <script>
- function newDoc(){
- window.location.assign("http://www.hactcm.edu.cn");
- }
- </script>
- <input type="button" value="加载新文档" onclick="newDoc()">
- </body>
- </html>

navigator对象

- navigator对象主要是用来检测客户端浏览器信息的，关于Web浏览器的信息，浏览器的类型、版本信息以及操作系统的类型都可以从该对象中获取。
- 属性如下所示：

属性	描述
appName	返回浏览器的名称
appVersion	返回浏览器的平台和版本信息
platform	返回运行浏览器的操作系统平台
userAgent	返回由客户机发送服务器的user-agent头部的值.

navigator对象

- navigator对象最常用的属性是userAgent，通常浏览器及操作系统的判断都是通过该属性来实现的，最基本的方法是首先将它的值赋给一个变量，代码如下：
- var sUserAgent = navigator.userAgent;
- document.write(sUserAgent);

screen对象

- screen对象主要用来获取用户电脑的屏幕信息，包括屏幕的分辨率，屏幕的颜色位数，窗口可显示的最大尺寸。

属性如下：

属性	描述
Width	整个屏幕的水平尺寸，以像素为单位
height	整个屏幕的垂直尺寸，以像素为单位
pixelDepth	显示器的每个像素的位数
colorDepth	返回当前颜色设置所用的位数
availWidth	返回窗口内容区域的水平尺寸，以像素为单位
availHeight	返回窗口内容区域的垂直尺寸，以像素为单位

screen对象

例如:

```
<script>
    document.write("可用宽度: " + screen.availWidth);
    document.write("可用高度: " + screen.availHeight);
</script>
```

history对象

- window.history 对象包含浏览器的历史。它的常用属性：

属性	描述
length	历史列表的长度，用于判断列表中的入口数目
current	当前文档的URL
next	历史列表的下一个URL
previous	历史列表的前一个URL

- 它的常用方法：

方法	描述
back（）	退回前一页
forward（）	重新进入下一页
go（）	进入指定的页面

history对象

- window还有一个非常实用的属性是history。它可以访问历史页面，如果希望浏览器返回前一页可以使用如下代码：
- window.history.go(-1);
- 如果希望前进一页，只需要使用正数1即可，代码如下：
- window.history.go(1);
- 如果希望刷新显示当前页，则使用0即可，代码如下：
- window.history.go(0);

- history.back() - 与在浏览器点击后退按钮相同
- history.forward() - 与在浏览器中点击按钮向前相同

综合案例

案例9：计时器：

```
<script language="javascript">
    var flag=0;
    var timeID;
    function beg(){
            var i=form1.num.value;
            i++;
            form1.num.value=i;

    timeID=setTimeout("beg()",1000);
    }
    function sta(){
            if(flag==0){
                    beg();
                    flag=1;
            }
    }
    function pau(){
                    clearTimeout(timeID);
                    flag=0;
            }
</script>
<form id="form1" name="form1" method="post" action="">
        <input type="text" name="num" size="1" value="0" />
        <input type="button" name="start" value="开始计时" onclick="sta();" />
        <input type="button" name="pause" value="暂停计时" onclick="pau();" />
</form>
```

想一想

- 1、通过js判断浏览器的种类，并弹出提示框。
- 2、使页面自适应浏览器窗口