JavaScript 概述

1 认识 JavaScript

1.1 JavaScript 是什么

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言；特点：脚本语言，基于对象，简单性，安全性，动态性，跨平台；

1.2 JavaScript 与 Java 的区别

Java 程式语言，JS 可嵌入到 web 页面中，基于对象和事件驱动的解释性语言，二者区别：

JS	基于对象； 解释； 弱变量； 动态绑定 ； 嵌入方式：在html中用<Script>...</Script>来标识 ； 代码为文本字符格式，可直接嵌入html 也可动态装载；
JAVA	面向对象； 编译； 强变量； 静态绑定； 嵌入方式：在html中用<applet>...</applet>来标识； 代码格式，与html无关，通过引用外媒那样装载；

1.3 第一个 JS 程序

```
<title>第一个 JS</title>
    <script type="text/javascript">
    <!--
    //一下是代码；
        alert("我的第一程序 js 点击");
        document.write("<center><h3>document.write()的说明</h3>:<br>document 是指当前的 html 文档,使用 JavaScript 脚本中的 document 对象的 write 方法,<br>是文档对象输出函数!</center>");
        document.close();  //-->
    </script>
    </head>
```

将 JavaScript 标识放置<Head>... </Head>在头部之间，使之在主页和其余部分代码之前装载，从而可使代码的功能更强大；可以将 JavaScript 标识放置在<Body>... </Body>主体之间以实现某些部分动态地创建文档。

我们也可以把 JavaScript 代码保存在独立的文件中(扩展名为.js),在 Html 中引入这个文件；

```
<body>
<script src="importJs.js">
</script>
```

1.4 JS 中的关键字

abstract	continue	finally	iustanceof	private	this
boolean	default	float	int	public	throw
break	do	for	interface	return	typeof
byte	double	function	long	short	true
case	else	goto	native	static	var
catch	extends	implements	new	super	void
char	false	import	null	switch	while
class	final	in	package	synchronized	with

1.5 基本数据类型

1. 数值：（整数和实数）
2. 字符串型：（用“”号或‘’括起来的字符或数值）
3. 转义字符：“\”开头：如\' 单引号）
4. 布尔型：使 True 或 False 表示
5. 空值：null，空值不等于字符串“”或 0；
6. 未定义值：当使用了一个未声明，或者已声明但未赋值的变量时，返回一个未定义值（undefined）

1.6 表达式和运算符

同其他语言，表达式是变量、常量、布尔及运算符的集合，因此表达式可以分为算术表述式、字串表达式、赋值表达式以及布尔表达式等

逻辑与运算，并将结果赋给左边	a&=b //相当于 a=a&b
逻辑或运算，并将结果赋给左边	a|=b //相当于 a=a|b
异或运算，并将结果赋给左边的	a^=b //相当于 a=a^b

1.7 js 的基本构成

基本构成是由控制语句、函数、对象、方法、属性等

1.8 控制语句

```
if()  else
switch(){
case r1:
[break;]
case r2:

[defaut:
]
```

```
}
for(  ;  ;  ){
      }
```

1.9 函数

```
function 函数名（参数）{
函数体;.
return 表达式;
}
```

由关键字 function 定义，检测参数个数；

```
<Script Language="JavaScript">
functionTestArgLen(a,b,c,d,e)
{
len= TestArgLen.arguments.length;
if(len>0)
{     alert("a="+a);  }
...
if(len>4)
{     alert("e="+e);  }
}
TestArgLen(5,7,9);
document.write("运行结束!");
document.close();
</Script>
</head>
          调用函数：
<script type="text/javascript">
          functionmyfunction()
          {
          alert("你好!这里函数调用的运行结果!")
          }
          </script>
     </head>
<body>
     <form>
          <input type="button" onclick="myfunction()"    value="调用函数">
     </form>
```

1.10 事件驱动及处理

在 JavaScript 中对象事件的处理通常由函数(Function)担任，其基本格式与函数全部一样，可以将前面所介绍的所有函数作为事件处理程序。

```
function 事件处理名（参数列表）
{
```

事件处理语句集；

……

}

JavaScript 事件驱动中的事件是通过鼠标或热键的动作引发的。它主要有以下几个事件：

onblur	发生在窗口失去焦点； 应用于：window 对象
onchange	在文本输入区的内容被更改，然后焦点从文本输入区移走之后； 用于实时检测输入的有效性，或者立刻改变文档内容。 应用于：Password 对象；Select 对象；Text 对象；Textarea 对象
onclik	对象被单击；应用于：Button ； Checkbox ； Image ； Link ； Radio；Reset； submit
onerror	发生错误时； 应用于：window 对象
onfocus	发生在窗口得到焦点时； 应用于：window 对象；
onload	发生在文档下载完毕时； 应用于：window 对象；若在html中，则写在body中
onreset	表单重置时；return false值可阻止表单重置；应用于：form 对象
onresize	发生在窗口被调整大小时； 应用于：window 对象；
onsubmit	发生在表达提交按钮被单击时；false可阻止；应用于：form 对象；
onmousedown	发生错误时； 应用于：window 对象
onmouseout	发生在窗口得到焦点时； 应用于：window 对象；
onmouseover	文档下载完毕时； 用于：window 对象；
onmouseup	发生鼠标放在对象上，按下后，放开鼠标时；若按下后，鼠标并不在放开鼠标的对象上，本事件不发生； 应用于：butto n；link对象
onunload	发生在用户退出文档（或关闭窗口）， 应用于：window 对象； 若在html中，则写在body中

js 常用事件

事　件	何 时 触 发
onabort	对象载入被中断时触发
onblur	元素或窗口本身失去焦点时触发
onchange	改变<select>元素中的选项或其他表单元素失去焦点，并且在其获取焦点后内容发生过改变时触发
onclick	单击鼠标左键时触发。当光标的焦点在按钮上，并按下 Enter 键时，也会触发该事件
ondblclick	双击鼠标左键时触发
onerror	出现错误时触发
onfocus	任何元素或窗口本身获得焦点时触发
onkeydown	键盘上的按键（包括 Shift 或 Alt 等键）被按下时触发，如果一直按着某键，则会不断触发。当返回 false 时，取消默认动作
onkeypress	键盘上的按键被按下，并产生一个字符时发生。也就是说，当按下 Shift 或 Alt 等键时不触发。如果一直按下某键时，会不断触发。当返回 false 时，取消默认动作
onkeyup	释放键盘上的按键时触发
onload	页面完全载入后，在 Window 对象上触发；所有框架都载入后，在框架集上触发；<img>标记指定的图像完全载入后，在其上触发；或<object>标记指定的对象完全载入后，在其上触发
onmousedown	单击任何一个鼠标按键时触发
onmousemove	鼠标在某个元素上移动时持续触发
onmouseout	将鼠标从指定的元素上移开时触发
onmouseover	鼠标移到某个元素上时触发
onmouseup	释放任意一个鼠标按键时触发
onreset	单击重置按钮时，在<form>上触发
onresize	窗口或框架的大小发生改变时触发
onscroll	在任何带滚动条的元素或窗口上滚动时触发
onselect	选中文本时触发
onsubmit	单击提交按钮时，在<form>上触发
onunload	页面完全卸载后，在 Window 对象上触发；或者所有框架都卸载后，在框架集上触发

2　JS 对象基础知识

2.1 js 对象基本结构

JavaScript 中的对象是由属性(properties)和方法(methods)两个基本的元素构成的；

2.2 有关对象操作语句

在 JavaScript 中提供了几个用于操作对象的语句和关键字及运算符；

2.2.1　　for...in 语句

格式如下：for（对象属性名 in 已知对象名）

该语句的功能是用于对已知对象的所有属性进行操作的控制循环。它是将一个已知对象的所有属性反复置给一个变量；而不是使用计数器来实现的,该语句的优点是无需知道对象中属性的个数即可进行操作。

functionshowData(object)

for (vari=0; i<30;i++) document.write(object[i]) 该函数是通过数组下标顺序值，来访问每个对象的属性，使用这种方式首先必须知道数组的下标值，否则若超出范围，则就会发生错误；

如使用 for...in 语句，则根本不需要知道对象属性的个数，如下：

functionshowData(object){

for(var prop in object)

```
document.write(object[prop]);
}
```

使用该函数时，在循环体中，for 自动将的属性取出来，直到最后为此。

2.2.2 with 语句

在该语句体内，任何对变量的引用被认为是这个对象的属性，以节省一些代码。

```
with object{
...
}
```

例：with(Math)

```
document.write(cos(35));
document.write(cos(80));
```

若不使用 with 则引用时相对要复杂些：

```
document.write(Math.cos(35))
document.write(Math.sin(80))
```

2.2.3 this 关键字

this 是对当前的引用

2.2.4 New 运算符

创建对象使用如下格式：

New object=NEW Object(Parameters table);

Newobject 创建的新对象：object 是已经存在的对象；parameters table 参数表

2.3 JS 常用对象

2.3.1 String 字符串对象

(1) 属性:

属性:	描述
length	<字符串对象>.length；返回该字符串的长度

(2) 常用方法

方法	描述
charAt()	<字符串对象>.charAt(<位置>)；返回该字符串位于第<位置>位的单个字符
charCodeAt()	返回该字符串位于第<位置>位的单个字符的 ASCII 码。
indexOf()	<字符串对象>.indexOf(<另一个字符串对象>[, <起始位置>])；该方法从<字符串对象>中查找<另一个字符串对象>（如果给出<起始位置>就忽略之前的位置），如果找到了，就返回它的位置，varstr="SZNSSchools is great!" varpos=str.indexOf("School")
lastIndexOf()	<字符串对象>.lastIndexOf(<另一个字符串对象>[, <起始位置>])；跟 indexOf() 相似，

	不过是从后边开始找。
substring()	<字符串对象>.substring(<始>[, <终>])；返回原字符串的子字符串，<终> - <始> = 返回字符串的长度（length）。如果没有指定<终>或指定得超过字符串长度，则子字符串从<始>位置一直取到原字符串尾。
substr()	<字符串对象>.substr(<始>[, <长>])；返回原字符串的子字符串，该字符串是原字符串从<始>位置开始，长度为<长>的一段。其他同上。
split()	<字符串对象>.split(<分隔符字符>)；返回一个数组，该数组是从<字符串对象>中分离开来的，<分隔符字符>决定了分离的地方，它本身不会包含在所返回的数组中。例如：'1&2&345&678'.split('&')返回数组：1,2,345,678。
toLowerCase()	<字符串对象>.toLowerCase()；返回把原字符串所有大写字母都变成小写的字符串
toUpperCase()	小写变大写；<script type="text/javascript">varstr=("Hello JavaScripters!") document.write(str.toUpperCase()) //小写变大写；
match()	<字符串对象>.match()；在源字符串中查找指定子串；若存在，返回指定字串，如不存在返回 null；varstr = "SZNSSchools is great!"; document.write(str.match("great"));
anchor(name)	为字符串对象中的内容两边加上 html 的标签
big()	为字符串对象中的内容两边加上 html 的<big></big>标签
bold()	为字符串对象中的内容两边加上 html 的标签
fontcolor	为字符串对象中的内容两边加上 html 的标签，并设置 color 属性，
fontsize(size)	为字符串对象中的内容两边加上 html 的标签，并设置 size 属性，
localeCompare(s)	用特定比较方法比较字符串与 s 字符串，若字符串相等，则返回 0，否则返回非 0 数字值。
replace()	string.replace(regExp,substring)；替换一个与正则表达式匹配的子串，regExp 正则表达式，substring 用于指定替换文本或生成替换文本的函数，$具有特殊含义，

2.3.2 Array 数组对象

(1) 定义方法：var<数组名> = new Array(size);

(2) 属性

length 用法：<数组对象>.length；

返回数组的长度，即数组里有多少个元素。它等于数组里最后一个元素的下标加一.

(3) 常用方法：

方法	描述
sort()	<数组对象>.sort([<方法函数>])；按指定方法排序，若不指定，按字母排序； 按升序排列数字： function sortMethod(a, b) {return a - b;} myArray.sort(sortMethod)；按降序排列数字：把上面的"a - b"该成"b - a"。
join()	<数组对象>.join(<分隔符>)；返回一个字符串，该字符串把数组中的各个元素串起来，用<分隔符>置于元素与元素之间,如果不指定分割符默认用","连接。类似字符串的 split()
reverse()	<数组对象>.reverse()；使数组中元素反序。对数组[1, 2, 3]用此方法，变成：[3, 2, 1]
slice()	<数组对象>.slice(<始>[, <终>])；返回一个从<始>到<终>的数组，该数组是原数组的子集
push()	<数组对象>.push("d","Jon")；使数组增加元素，返回数组长度；
pop()	<数组对象>.pop()，取出数组最后一个元素，

shift()	<数组对象>.shift()，取出开始的元素,数组长度减少

```
<body>
        <script type="text/javascript">
        varfamname=new Array(5)
            famname[0]="I"
            famname[1]="love"
            famname[2]="u"
            famname[3]="happy"
            famname[4]="everyday"
        for (i=0;i<famname.length;i++)
        {
            document.write(famname[i]+"<br>")
        }
document.write("数组长度： ")
document.write(famname.length + "<br>")
document.write("使用.链接成一个字符串:<br>");
document.write(famname.join(".") + "<br>" +"<br>")
document.write("使数组中的元素反序排列:<br>" )
document.write(famname.reverse() +"<br>" +"<br>")
document.write("在使数组中的增加两个元素,<br>返回数组长度:"+"/t"+"\t");
document.write(famname.push("Ola","Jon") + "<br>")
document.write(famname.join(" . ") + "<br>" +"<br>")
        document.write("取出最后一个元素,<br>数组长度减少:"+"\t");
        document.write(famname.pop() + "<br>")
        document.write(famname.join(".") + "<br>"+ "<br>")

        document.write("取出开始的元素,<br>数组长度减少:  ");
        document.write(famname.shift() + "<br>")

        document.write(famname.join(" & ") + "<br>")
        document.write(famname.shift() + "<br>")
        document.write(famname.join(".") + "<br>")
</script>
```

2.3.3 Math “数学”对象

格式：“Math.<名>”

abs()：返回 x 绝对值；

acos(x)：反余弦值，用弧度表示。

asin(x)：反正弦值。

atan(x) ：反正切值。

sin(x)：返回 x 的正弦。

cos(x)：返回 x 的余弦。

sqrt(x)：返回 x 的平方根。

tan(x)：返回 x 的正切。

random()：返回大于 0 小于 1 的随机数。

max(a, b)：返回 a, b 中较大的数。

min(a, b)：返回 a, b 中较小的数。

exp(x)：返回 e 的 x 次幂 (ex)。

floor(x)：返回小于等于 x 的最大整数。

log(x)：返回 x 的自然对数 (ln x)。

pow(n, m)：返回 n 的 m 次幂 (nm)。
round(x)：返回 x 四舍五入后的值。
ceil(x)：返回大于等于 x 的最小整数。
atan2(x, y)：返回复平面内点(x, y)对应的复数的幅角，用弧度表示，其值在 -π 到 π 之间。

2.3.4 Date 日期对象

定义一个日期对象：

var d = new Date;

方法：以下有很多“g/set[UTC]XXX”这样的方法，它表示既有“getXXX”方法，又有“setXXX”方法。“get”是获得某个数值，而“set”是设定某个数值。如果带有“UTC”字母(如 getUTCXXX)，则表示获得/设定的数值是基于 UTC 时间的，没有则表示基于本地时间或浏览期默认时间

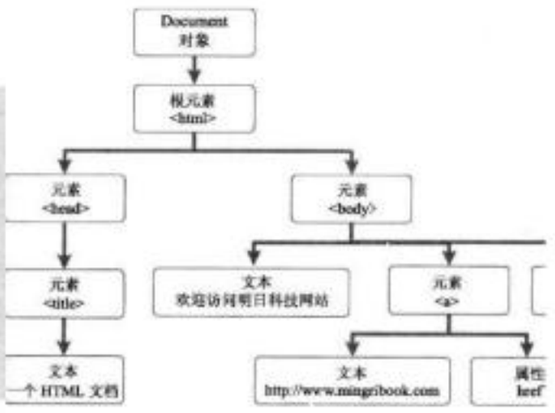
方法	描述	方法	描述
g/set[UTC]FullYear() g/set[UTC]Year()	返回/设置年份，用四位数表示//用两位数表示	**toString()**	返回字符串，描述对象所指日期。格式类似：“Fri Jul 21 15:43:46 UTC+0800 2000”
g/set[UTC]Month()	返回/设置月份返回/设置星期，取值 0-11,0 表示一月	**g/setTime()**	返回/设置时间，该时间就是日期对象的内部处理方法,用毫秒表示
g/set[UTC]Date()	返回/设置日期，取值 1-31,	**g/set[UTC]Minutes()**	返回/设置分钟数,取值 0-59
get[UTC]Day()	返回/设置星期，取值 0-6 ；0 表示星期天	**g/set[UTC]Seconds()**	返回/设置秒钟数,取值 0-59
g/set[UTC]Hours()	返回/设置小时数，取值 0-23,0 为午夜 24 点		

2.3.5 DOM 技术

DOM，Document Object Model 文档对象模型，表示文档和访问操作构成文档的各种元素(如 html 标记和文本串)的应用程序接口(API)

【例 3.40】 一个简单的 HTML 文档说明 DOM 的分层结构。

```
<html>
    <head>
        <title>一个 HTML 文档</title>
    </head>
    <body>
        欢迎访问明日科技网站!
        <br>
        <a href="http://www.mingribook.com"> http://www.mingribook.com</a>
    </body>
</html>
```



DOM 中，各个节点被视为各种类型的 Node 对象，html 文档视为 node 对象的树，

(1) 通过 ID 获取文档指定元素；getElementsById()方法可以通过元素的 ID 属性获取元素；

(2) 通过 Name 属性获取元素：getElementsByName（），不同于 id，此方法返回一个数组，若要页面中唯一元素，可返回数组下标值为 0 的元素，如：

document.getElementsByName(“userName”)[0];

(3) 操作文档

DOM 中，不仅可查询节点，也可对节点进行增删、修改、替换等操作，

Node 对象常用方法：

方　法	描　述
insertBefore(newChild,refChild)	在现有节点 refChild 之前插入节点 newChild
replaceChild(newChild,oldChild)	将子节点列表中的子节点 oldChild 换成 newChild，并返回 oldChild 节点
removeChild(oldChild)	将子节点列表中的子节点 oldChild 删除，并返回 oldChild 节点
appendChild(newChild)	将节点 newChild 添加到该节点的子节点列表末尾。如果 newChild 已经在树中，则先将其删除
hasChildNodes()	返回一个布尔值，表示节点是否有子节点
cloneNode(deep)	返回这个节点的副本（包括属性）。如果 deep 的值为 true，则复制所有包含的节点；否则只复制这个节点

编写 js 函数 deleteElement（），用于在评论列表中添加一条评论，到列表后面，然后清空评论人和评论内容文本框。

编写 js 函数 deleteFirstE（），用于删除列表的最后一条评论信息；

```
function deleteFirstE(){
      var tComment = document.getElementById("comment");          //获取 table 对象
      if(tComment.rows.length>1){
            tComment.deleteRow(1);                                //删除表格的第二行，即第一条评论
      }
}
```

编写 js 函数 deleteLastE（），用于删除列表的最后一条评论信息；

```
if(tComment.rows.length>1){
      tComment.deleteRow(tComment.rows.length-1);          //删除表格的最后一行，即最后一条评论
}
```

调用：

```
<input name="Button" type="button" class="btn_grey" value="删除第一条评论" onclick="deleteFirstE()">
<input name="Button" type="button" class="btn_grey" value="删除最后一条评论" onclick="deleteLastE()">
```

例子见 E:htmlscript\jsDOMmethod.html

2.4 系统函数、常用内部对象

2.4.1 系统函数（内部方法）

eval（字串表达式）：返回字符串表达式中的值；

isNaN()：如括号内的值是“NaN”则返回 true 否则返回 false；

parseInt()：返回把括号内的内容转换成整数之后的值；

parseFloat()：返回把括号内的字符串转换成浮点数之后的值，字符串开头的数字部分被转换成浮点数，如果以字母开头，则返回“NaN”；

toString()：<对象>.toString()；把对象转换成字符串。如果在括号中指定一个数值，则转换过程中所有数值转换成特定进制

escape()：返回括号中的字符串经过编码后的新字符串。该编码应用于 URL，也就是把空格写成“%20”这种格式。“+”不被编码，如果要“+”也被编码，请用：escape('...', 1)。

unescape()：是 escape() 的反过程。

2.4.2 内部对象树

使用浏览器的内部对象系统，可实现与 HTML 文档进行交互。它的作用是将相关元素组织包装起来，提供给程序设计人员使用，从而减轻编程人的劳动，提高设计 Web 页面的能力.

- navigator　浏览器对象
- screen　屏幕对象
- window　窗口对象
  - history　历史对象
  - location　地址对象
  - frames[]: Frame　框架对象
  - document　文档对象
    - anchors[]; links[]: Link　连接对象
    - applets[]　Java小程序对象
    - embeds[]　插件对象
    - forms[]: Form　表单对象
      - Button　按钮对象
      - Checkbox　复选框对象
      - elements[]: Element　表单元素对象
      - Hidden　隐藏对象
      - Password　密码输入区对象
      - Radio　单选域对象
      - Reset　重置按钮对象
      - Select　选择区（下拉菜单、列表）对象
        - options[]: Option　选择项对象
      - Submit　提交按钮对象
      - Text　文本框对象
      - Textarea　多行文本输入区对象
    - images[]: Image　图片对象

2.4.3　navigator 浏览器对象和 screen 屏幕对象

反映了当前使用浏览器的资料

常用属性有

属性	描述
appCodeName	回浏览器的“码名”,流行的 IE 返回 'Mozilla
appName	返回浏览器名。
appVersion	返回浏览器版本，包括了大版本号、小版本号、语言、操作平台等信息
platform	返回浏览器的操作平台
platform	返回浏览器的操作平台

```
<script type="text/javascript">
document.write("<p>浏览器: ")
document.write(navigator.appName + "</p>")
document.write("<p>浏览器代理头信息: ")
document.write(navigator.userAgent + "</p>")
</script>
</body>
```

screen 屏幕对象反映了用户的屏幕设置，

其属性有：

width　返回屏幕的宽度（像素数）。

height　返回屏幕的高度。

availWidth　返回屏幕的可用宽度(除去了一些不自动隐藏的类似任务栏的东西所占用的宽度)。

availHeight　返回屏幕的可用高度。

colorDepth　返回当前颜色设置所用的位数

2.4.4 Window 对象

即浏览器窗口对象，是全局对象，是所有对象的顶级对象，

(1) 常用属性

属　性	描　述
document	对窗口或框架中含有文档的 Document 对象的只读引用
defaultStatus	一个可读写的字符，用于指定状态栏中的默认消息
frames	表示当前窗口中所有 Frame 对象的集合
location	用于代表窗口或框架的 Location 对象。如果将一个 URL 赋予该属性，则浏览器将加载并显示该 URL 指定的文档
length	窗口或框架包含的框架个数
history	对窗口或框架的 History 对象的只读引用
name	用于存放窗口对象的名称
status	一个可读写的字符，用于指定状态栏中的当前信息
top	表示最顶层的浏览器窗口
parent	表示包含当前窗口的父窗口
opener	表示打开当前窗口的父窗口
closed	一个只读的布尔值，表示当前窗口是否关闭。当浏览器窗口关闭时，表示该窗口的 Window 对象并不会消失，不过其 closed 属性被设置为 true
self	表示当前窗口
screen	对窗口或框架的 Screen 对象的只读引用，提供屏幕尺寸、颜色深度等信息
navigator	对窗口或框架的 Navigator 对象的只读引用，通过 Navigator 对象可以获得与浏览器相关的信息

(2) window 对象常用方法：

方法	描述
open()	open(<URL >, <窗口名称字符串（window.name，可以使用'_top'、'_blank'等内建名称，跟“<a href="..." target="...">”里的“target”属性一样），>, <参数字符串被打开的窗口的样貌>)；例打开一个 400 x 100 的干净的窗口： open('','_blank','width=400,height=100,menubar=no,toolbar=no,location=no,directories=no,status=no,scrollbars=yes,resizable=yes') open() 方法有返回值，返回的就是它打开的窗口对象。所以，varnewWindow=open('','_blank')；这样把一个新窗口赋值到“newWindow”变量中，以后通过“newWindow”变量就可以控制窗口了
close()	<窗口对象>.close() 如 self.close()：关闭指定的窗口；如该窗口有状态栏，调用该方法后浏览器会警告：“网页正在试图关闭窗口，是否关闭？”然后等待用户选择是否
blur()	使焦点从窗口移走，窗口变为“非活动窗口”
focus()	使窗口获得焦点，变为“活动窗口”
alert()	alert(<字符串>)；弹出一个只包含“确定”按钮的对话框，显示<字符串>的内容
confirm()	confirm(<字符串>)；弹出一个含“确定”“取消”的对话框，显示<字符串>的内容，要求用户做出选择，按下“确定”，则返回 true 值，如果按下“取消”，则返回 false 值
prompt()	prompt(<字符串>[, <初始值>])；弹出文本框的对话框，显示<字符串>的内容，要求用户在文本框输入数据，按下“确认”，则返回文本框里已有的内容，按下“取消”，则返回 null 值，如指定<初始值>，则文本框里会有默认值

scrollTo(x,y)	把窗口滚动到 x,y 坐标指定的位置
scrollBy(offsetx,offsety)	按照指定的位移量滚动窗口
setTimeout(timer)	在经过指定的时间后执行代码
clearTimeout()	取消对指定代码的延迟执行
moveTo(x,y)	将窗口移动到一个绝对位置
moveBy(offsetx,offsety)	将窗口移动到指定的位移量处
resizeTo(x,y)	设置窗口的大小
resizeBy(offsetx,offsety)	按照指定的位移量设置窗口的大小
print()	相当于浏览器工具栏中的“打印”按钮
setInterval()	周期性执行指定的代码
clearInterval()	停止周期性地执行代码

2.4.5 location 地址对象

格式：“<窗口对象>.location”

属性：

方法	描述	方法	描述
protocol	返回地址的协议，取值为'http:','https:','file:' 等	**search**	返回“？”及以后的内容；如“http://www.a.com/b/c.asp?selection=3&jumpto=4”，location.search == '?selection=3&jumpto=4'；如果地址里没有“?”，则返回空字符串
hostname	返回地址的主机名；如：“http://www.microsoft.com/china/”，location.hostname=='www.microsoft.com'	**href**	返回整个地址；
port	返回地址的端口号，一般 http 的端口号是 '80'	**reload()**	相当于按浏览器上的“刷新”(IE)或“Reload”(Netscape)键
host	返回主机名和端口号；如：location.host=='www.a.com:8080'	**replace()**	打开一个 URL，并取代历史对象中当前位置的地址。即按下浏览器的“后退”键将不能返回到刚才的页面
pathname	返回路径名，如“http://www.a.com/b/c.html”，location.pathname = = 'b/c.html'	**frames[]**	Frame 框架对象；引用方法：window.frames[x]，window.frameName，window.frames['frameName']，x 指该 window 对象指定的第几个框架，'frameName'指该框架名字;

2.4.6 document 文档对象

用法：document （当前窗口）或<窗口对象>.document （指定窗口）

属性

方法	描述	方法	描述
cookie		**fgColor**	指<body>标记的 text 属性所表示的文本颜色

title	指<head>标记里用<title>...</title>定义的文字	**bgColor**	指<body>标记的 bgcolor 属性所表示的背景颜色
referrer	如果当前文档是通过点击连接打开的，则 referrer 返回原来的 URL	**linkColor**	指<body>标记的 link 属性所表示的连接颜色
lastModified	当前文档的最后修改日期，是一个 Date 对象	**alinkColor**	指<body>标记的 alink 属性所表示的活动连接颜色
		vlinkColor	vlink 属性所表示的已访问连接颜色

方法：

方法	描述
write(),writeln()	writeln() 在写入数据以后会加一个换行
close()	关闭文档，停止写入数据

2.4.7 anchors[]; links[]; link 连接对象

用法：document.anchors[[x]]; document.links[[x]]; <anchorId>; <linkId>

document.anchors 是一个数组，包含了文档中所有锚标记（包含 name 属性的<a>标记），按照在文档中的次序，从 0 开始给每个锚标记定义了一个下标

document.links 也是一个数组，包含了文档中所有连接标记（包含 href 属性的<a>标记和<map>标记段里的<area>标记），按照在文档中的次序，从 0 开始给每个连接标记定义了一个下标

单个 Anchor 对象没有属性；

单个 Link 对象的属性见下。

属性：

protocol; hostname; port; host; pathname; hash; search; href 与 **location** 对象相同。

target 返回/指定连接的目标窗口（字符串），与<a>标记里的 target 属性是一样的。

事件：

onclick; onmouseover; onmouseout; onmousedown; onmouseup

2.4.8 embeds[] 插件对象

它是一个数组，包含了文档中所有的插件（<embed>标记）。因为每个插件的不同，每个 Embed 对象也有不同的属性和方法。

2.4.9 forms[];Form 表单对象

document.forms[] 数组，包含了文档中所有的表单（<form>）。

引用：document.forms[x] 或“document.<表单名>”，表单名<form>标记的 name="表单名"属性

(1) Form 表单对象属性：

name，action，method，target，encoding；length；length；分别同 form 标签中，<form name="...">等等属性

(2) 方法

reset()，重置表单； submit()，提交表丹；例如：

```
<script type="text/javascript">
        functionformReset()
        {
        var x=document.forms.myForm;
        x.reset();
        alert("表单已经重置");
        }
      •••
</script>
<body>
        <form name="myForm">
        <p>控制表单的提交和重置</p>
        <input type="text" size="20"><br>
        <br>
        <input type="button" onclick="formReset()" value="重置">
        </form>
</body>
```

2.4.10 elements[];element 表单元素元素

<表单对象>.elements 是一个数组，包含了该表单所有的对象。

2.4.11 Hidden 隐藏对象

由“<input type="hidden">”指定

属性

name：返回/设定用<input name="...">指定的元素名称。

value：返回/设定用<input value="...">指定的元素的值。

form：返回包含本元素的表单对象

2.4.12 Password 密码输入区对象

属性	描述	方法	描述
name	返回/设定<input name="">指定的元素名	**blur()**	从对象中移走焦点
value	返回/设置密码输入区当前的值	**focus()**	让对象获得焦点
defautValue	返回/设定<input value="">指定的默认值	**select()**	选中密码输入区里全部文本
form	返回包含本元素的表单对象		
事件	**onchange**		

2.4.13 Radio 单选域对象

属性：name，value，form ；方法：blur()；focus()同上，

属性 checked 返回或谁当该单选域对象是否被选中；

方法 click()模拟鼠标点击对象。

事件：**onclick**

2.4.14 Reset 重置按钮对象

由“<input type="reset">”指定。因为 Reset 也是按钮，所以也有 Button 对象的属性和方法。至于“onclick”事件，一般用 Form 对象的 onreset 代替

2.4.15 Select 选择区（下拉菜单、列表）对象

由“<select>”指定

属性：name，length，selectIndex(返回选项下标)，form，同上，

方法：blur(),focus(),同上

事件：onchange

2.4.16 options[]; Option 选择项对象

options[]，是一个数组，包含了在同一个 Select 对象下的 Option 对象，Option 对象由“<select>”下的“<options>”指定

options[] 数组的属性：

length; selectedIndex 同 Select 对象；

单个 Option 对象的属性

text：返回/指定 Option 对象所显示的文本

value：返回/指定 Option 对象的值，与<options value="...">一致。

index：返回该 Option 对象的下标。对此并没有什么好说，因为要指定特定的一个 Option 对象，都要先知道该对象的下标。

selected：返回/指定该对象是否被选中。

defaultSelected：返回该对象默认是否被选中

例：

```
function put()
   {
   txt=document.forms[0].myClass.options[document.forms[0].myClass.selectedIndex].te
xt
   document.forms[0].classSec.value=txt
   }
   functionmakeDisable()
   {
         var x=document.getElementById("mySelect")
         x.disabled=true
   }
</script>
</head>
<body>
选择培训课程:<br>
<form>
   选择你喜欢的课程:
      <select name="myClass" onchange="put()">
      <option selected>J2me</option>
```

```
        <option>J2ee</option>
        </select><br>
      你选择的课程是: <input type="text" name="classSec" size="20">
      <input type="button" onClick="makeDisable()" value="屏蔽列表">
</form>
</body>
```

2.4.17 Submit 提交按钮对象

由“<input type="submit">指定。因为 Submit 也是按钮，所以也有 Button 对象的属性和方法。至于“onclick”事件，一般用 Form 对象的 onsubmit 代替。

2.4.18 Text 文本框对象

由“<input type="text">”指定。Password 对象也是 Text 对象的一种，所以 Password 对象所有的属性、方法和事件，Text 对象都有

2.4.19 Textarea 多行文本输入区对象

2.4.20 images[]; Image 图片对象

document.images[] 是一个数组，包含了文档中所有的图片（<img>）。要引用单个图片，可以用 document.images[x]。如果某图片包含“name”属性，也就是用“<img name="...">”

在 IE 中，如果某图片包含 ID 属性，也就是用“<img id="...">”这种格式定义了一幅图片，就可以直接使用“<imageID>”来引用图片

单个 Image 对象的属性

name; src; width; height; vspace; hspace; border ； 跟<img>标记的同名属性一样。

综合例子 E:\HTML script\textObject.html

2.5 使用框架和 cookies

2.5.1 frames[]框架

每一个 HTML 文件占用一个 window 对象，包括定义框架的网页（“框架网页”）。每一个框架都是包含它的页的 window 对象的一个子对象；

frames[]	**Frame** 框架对象；引用方法：**window.frames[x]**，**window.frameName**，**window.frames['frameName']**，**x** 指该 **window** 对象指定的第几个框架，**'frameName'**指该框架名字;

如果使用 window.frameName 指定的 window 对象又是一个框架网页，那么引用它的框架的方法：window.frameName.subFrameName。以此类推

引用“window”对象所返回的，都是“当前”window 对象。要访问其它 window 对象，就要用到 parent 、 top 属性。

parent： “父级”window 对象，也就是包含当前 window 对象的框架网页；

top： 窗口最顶端的 window 对象

2.5.2 cookie 属性

document 的 cookie 属性

每个 Cookie 都是这样的：<cookie 名>=<值>，<cookie 名>的限制与 JavaScript 的命名限制大同小异，不能用 JavaScript 关键字，只能用可以用在 URL 编码中的字符.只要你只用字母和数字命名，就完全没有问题了。

<值>的要求也是“只能用可以用在 URL 编码中的字符

在某文档中添加“document.write(document.cookie)”，结果显示：name=kevin; email=kevin@kevin.com; lastvisited=index.html

escape()方法:Cookie 的值的要求是“只能用可以用在 URL 编码中的字符”。我们知道“escape()”方法是把字符串按 URL 编码方法来编码的，那我们只需要用一个“escape()”方法来处理输出到 Cookie 的值，用“unescape()”来处理从 Cookie 接收过来的值就万无一失了。而且这两个方法的最常用途就是处理 Cookies。其实设定一个 Cookie 只是“document.cookie = 'cookieName=cookieValue'”这么简单，但是为了避免在 cookieValue 中出现 URL 里不准出现的字符，还是用一个 escape() 好。

expires 设定 cookie 的失效时间,toGMTString() 方法：设定 Cookie 的时效日期都是用 GMT 格式的时间的