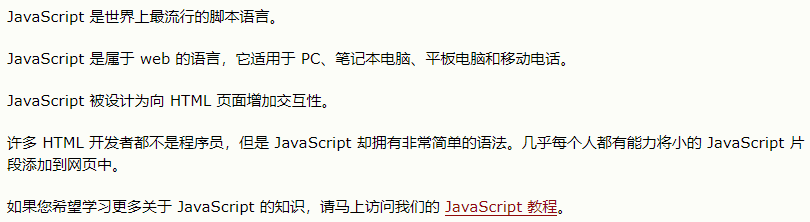
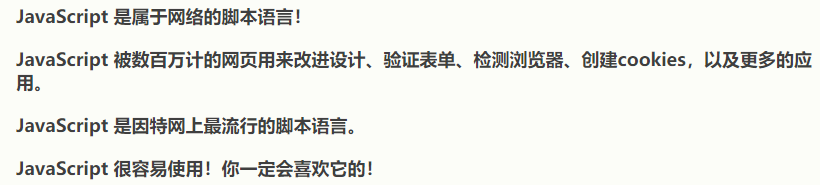
# JavaScript教程

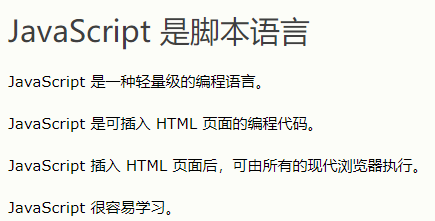
## 一、JavaScript教程

<http://www.w3school.com.cn/js/index.aspjs>



## 二、JavaScript简介





#### 1、JavaScript写入html输出：

<script>

document.write("<h1>This is a heading</h1>");

document.write("<p>This is a paragraph.</p>");

</script>



#### 2、JavaScript对事件做出反应

<button type="button" onclick="alert('恭喜你，点击领取奖品成功！')">点击这里</button>





#### 3、JavaScript改变html内容：

<script>

function myFunction()

{

x=document.getElementById("demo"); // 找到元素

x.innerHTML="JavaScript 能改变 HTML 元素的内容。是呀是呀！"; // 改变内容

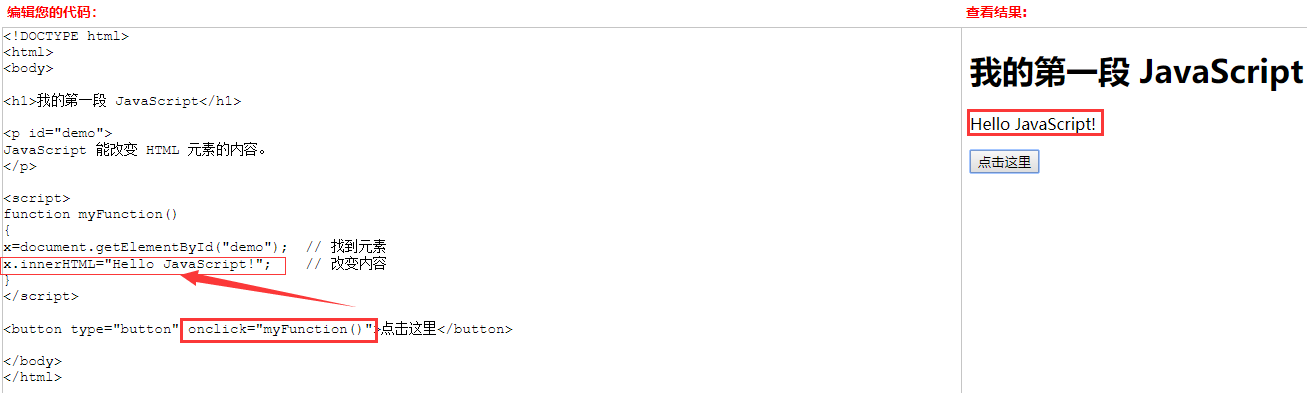
}

</script>



document.getElementByID("some id")。这个方法是 HTML DOM 中定义的。

DOM（文档对象模型）是用以访问 HTML 元素的正式 W3C 标准。



#### 4、JavaScript：改变html图像

<script>

function changeImage()

{

element=document.getElementById('myimage')

if (element.src.match("bulbon"))

{

element.src="/i/eg\_bulboff.gif";

}

else

{

element.src="/i/eg\_bulbon.gif";

}

}

</script>

<img id="myimage" onclick="changeImage()" src="/i/eg\_bulboff.gif">

本例会动态地改变 HTML <image> 的来源 (src)：

The Light bulb

点击灯泡就可以打开或关闭这盏灯

#### 5、JavaScript改变html样式：

<script>

function myFunction()

{

x=document.getElementById("demo") // 找到元素

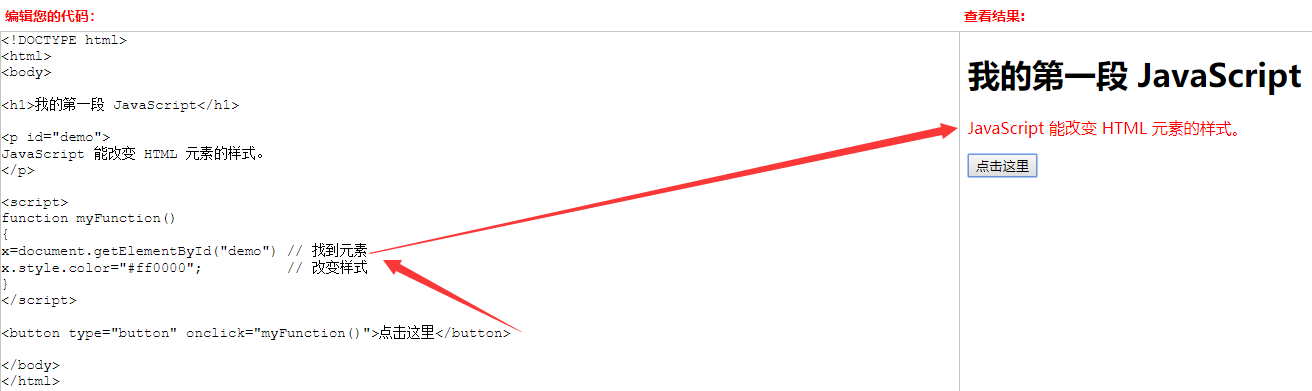
x.style.color=" #FF8C00"; // 改变样式

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

改变 HTML 元素的样式，属于改变 HTML 属性的变种。



#### 6、JavaScript验证用户输入：

<script>

function myFunction()

{

var x=document.getElementById("demo").value;

if(x==""||isNaN(x))

{

alert("Not Numeric");

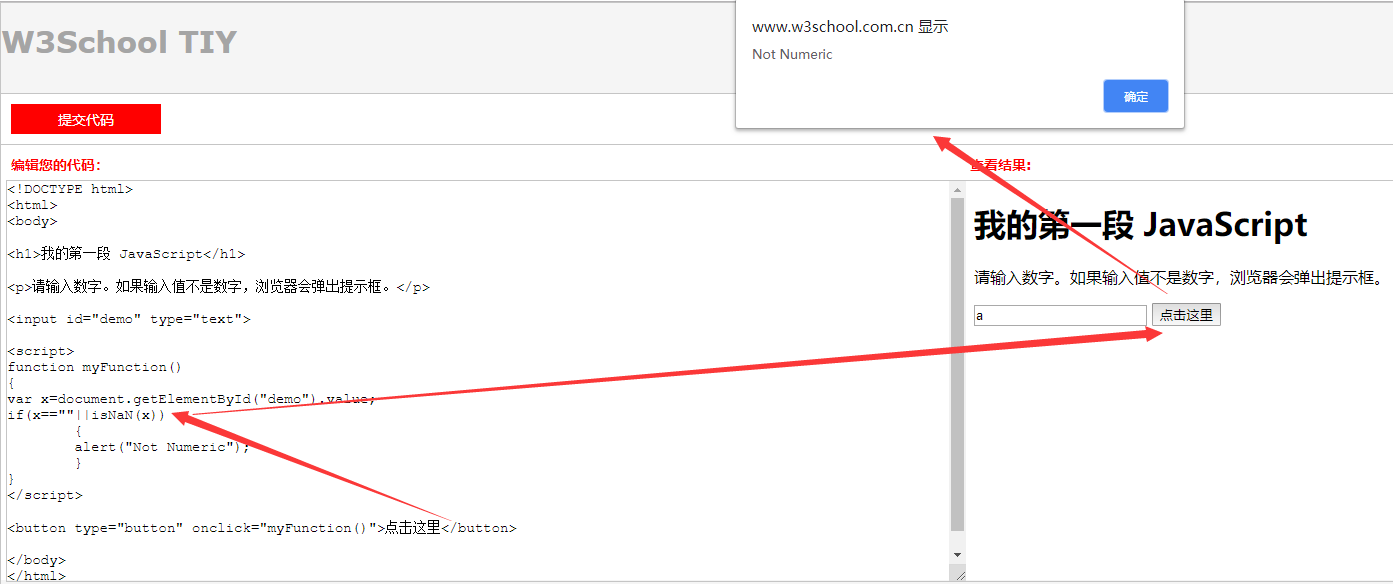
}

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

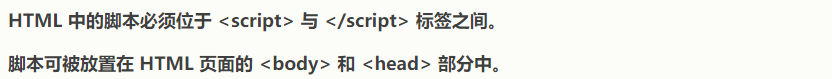
请输入数字。如果输入值不是数字，浏览器会弹出提示框。



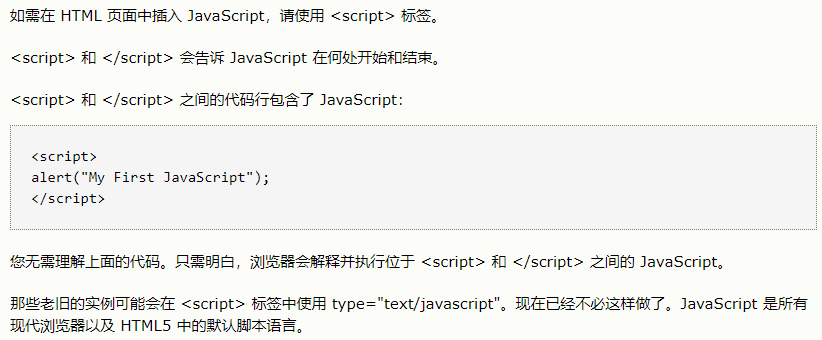
#### 7、JavaScript与Java区别

JavaScript 与 Java 是两种完全不同的语言，无论在概念还是设计上。Java（由 Sun 发明）是更复杂的编程语言。ECMA-262 是 JavaScript 标准的官方名称。JavaScript 由 Brendan Eich 发明。它于 1995 年出现在 Netscape 中（该浏览器已停止更新），并于 1997 年被 ECMA（一个标准协会）采纳。

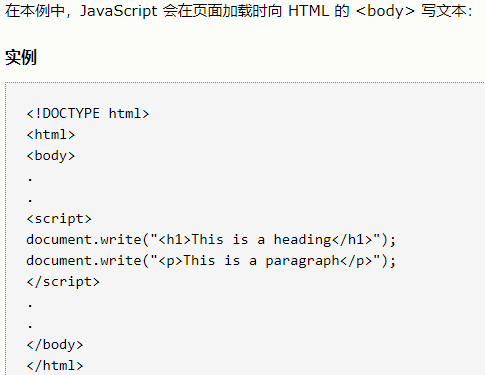
## 三、js实现和使用



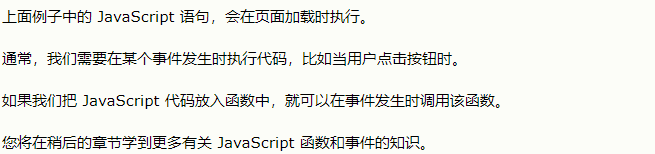
### script标签



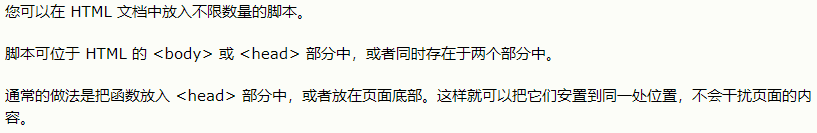
### 2、<body>中的JavaScript



### 3、JavaScript函数和事件



### 4、<head>或<body>中的JavaScript



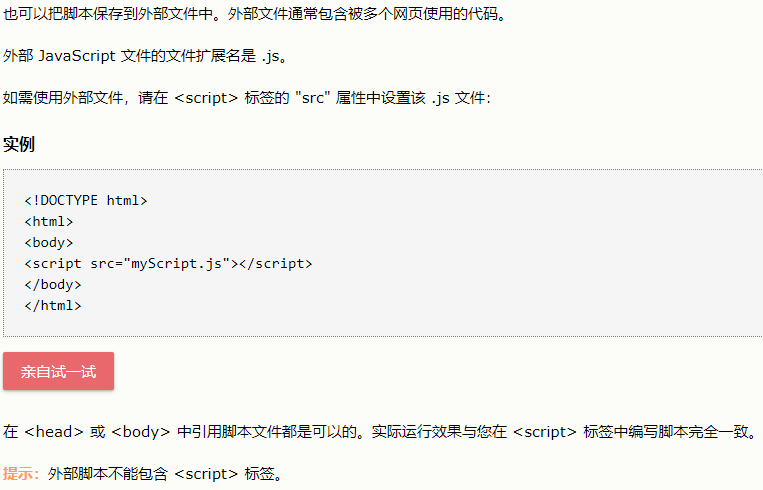
### 5、<head>中的JavaScript函数

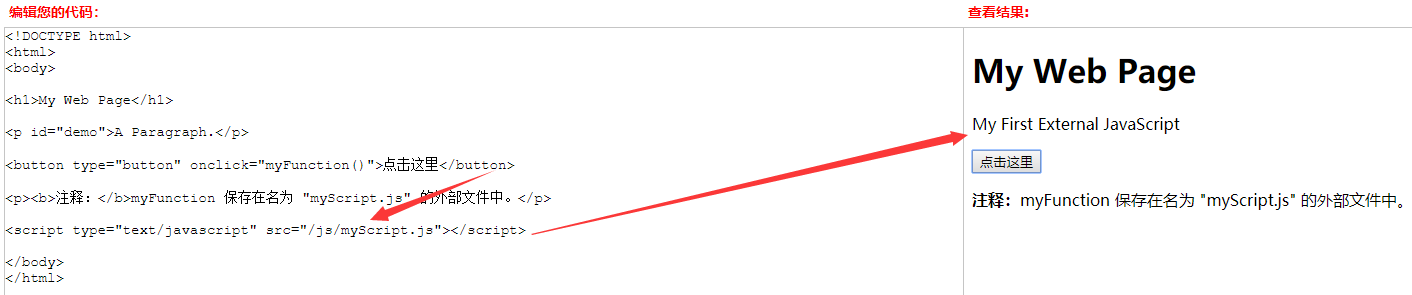


### 6、<body>中的JavaScript函数



### 7、外部的JavaScript





## 四、JavaScript输出



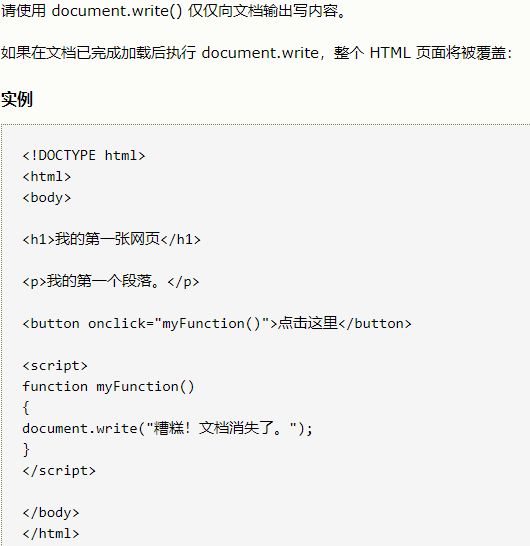
### 操作HTML元素

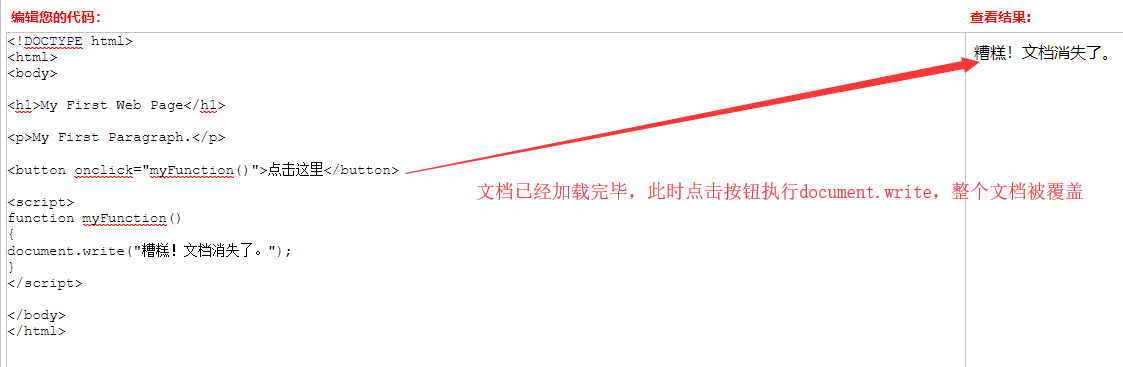


### 写到文档输出

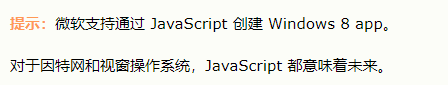


### 警告



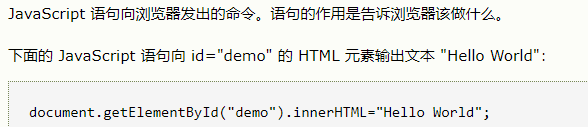


### Windows8中的JavaScript



## 五、JavaScript语句

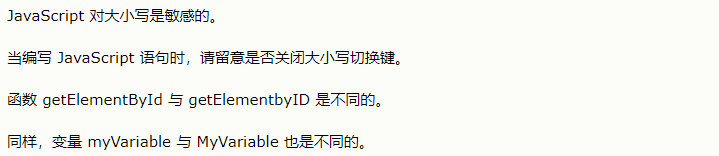
### JavaScript语句



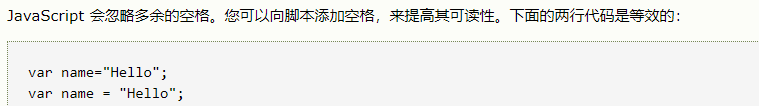
### 分号



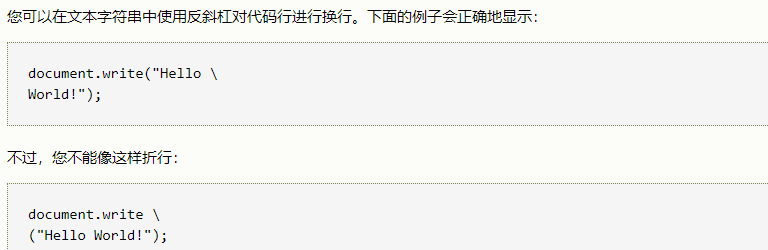
### JavaScript对大小写敏感



### 空格



### 对代码进行折行



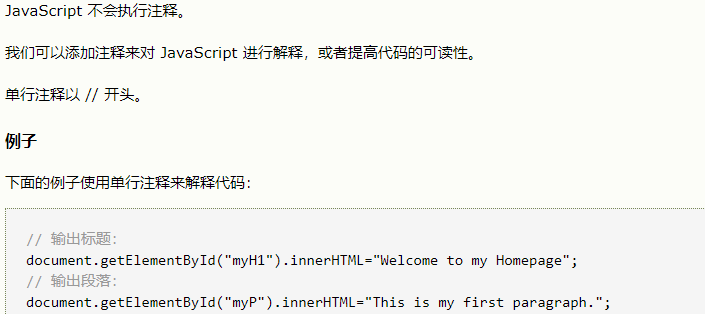
### JavaScript语言特点



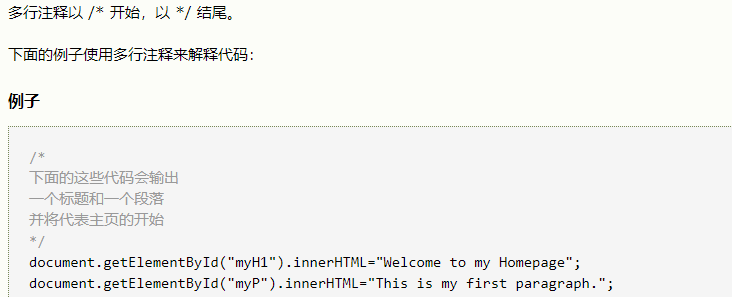
## 六、JavaScript注释

JavaScript代码注释可用于提高代码可读性

### JavaScript单行注释



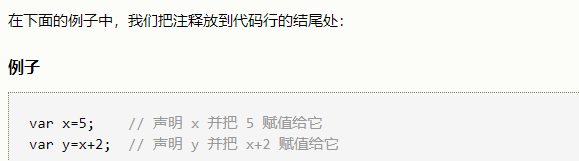
### JavaScript多行注释



### 使用注释来阻止执行



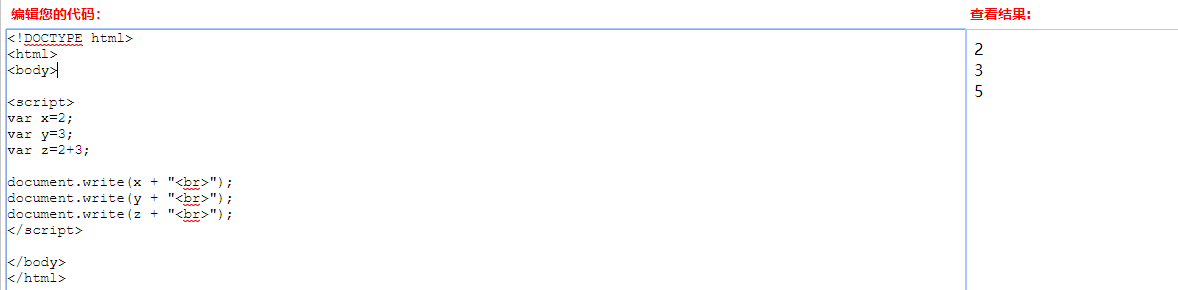
### 在行末执行注释

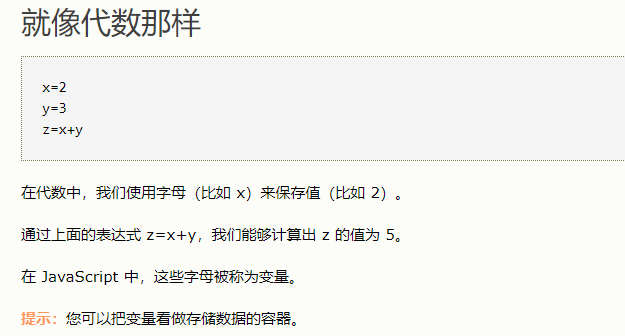


## JavaScript变量



### 实例

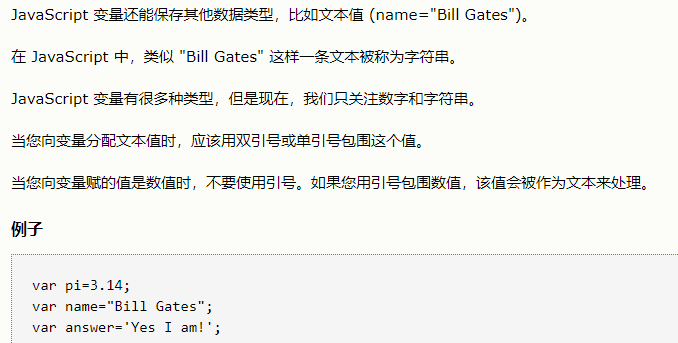




### JavaScript变量



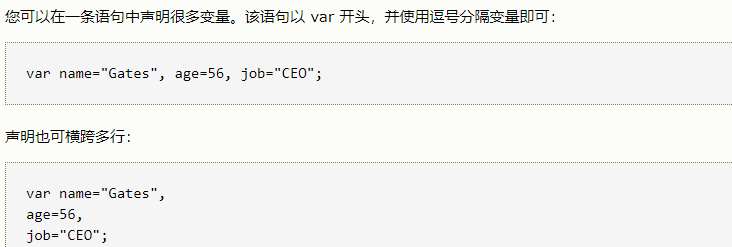
### JavaScript数据类型



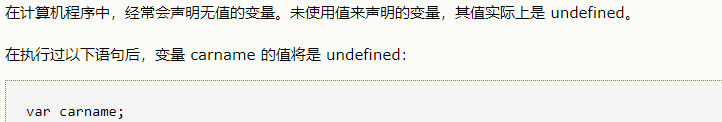
### 声明（创建）JavaScript变量



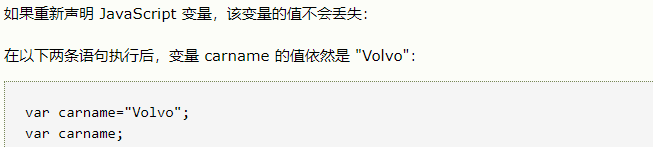
### 一条语句，多个变量



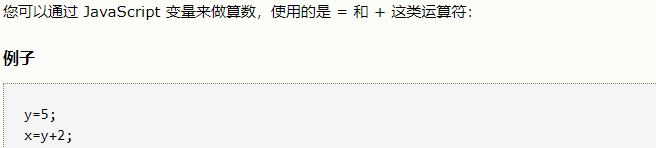
### value=undefined



### 重新声明JavaScript变量



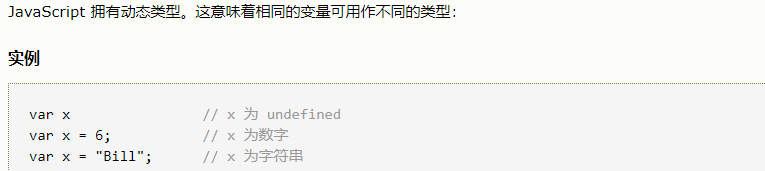
### JavaScript算数



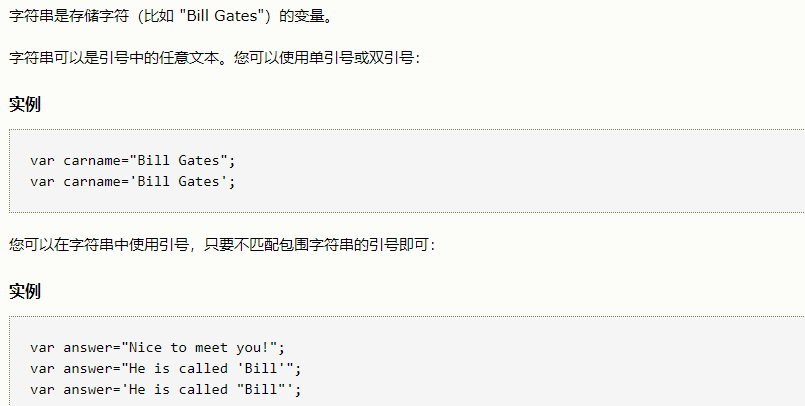
## JS数据类型



### JavaScript拥有动态类型



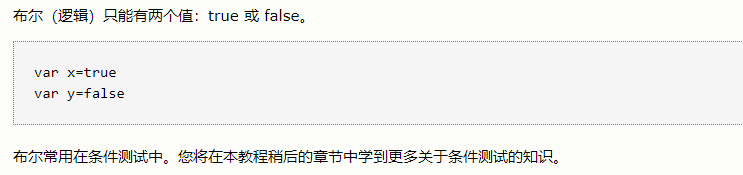
### JavaScript字符串



### JavaScript数字



### JavaScript布尔



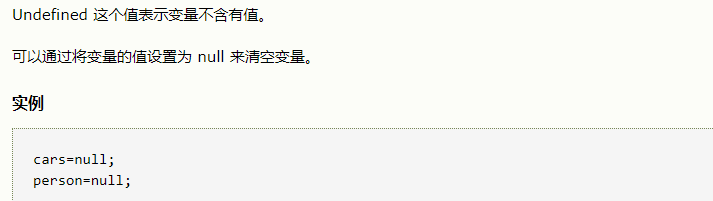
### JavaScript数组

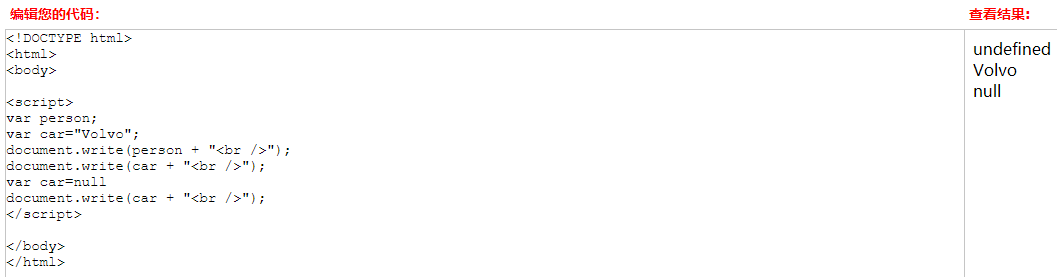


### JavaScript对象

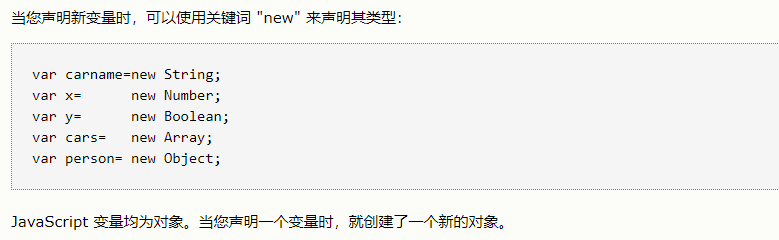


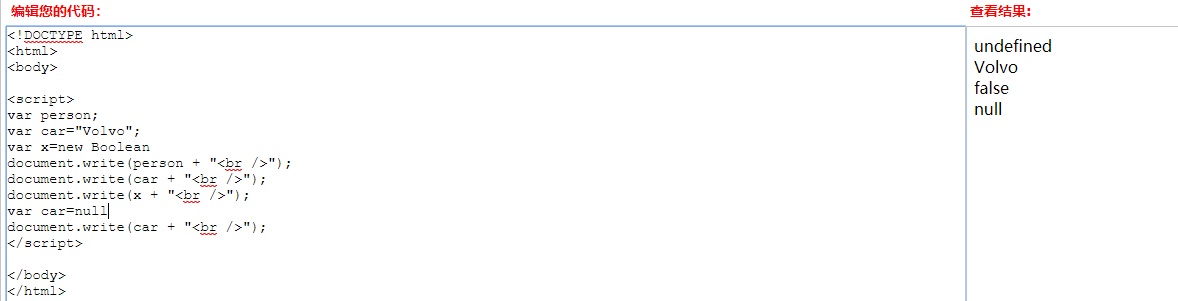
### undefined和null



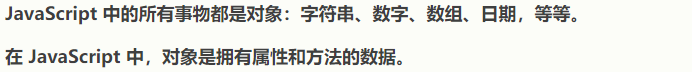


### 声明变量类型

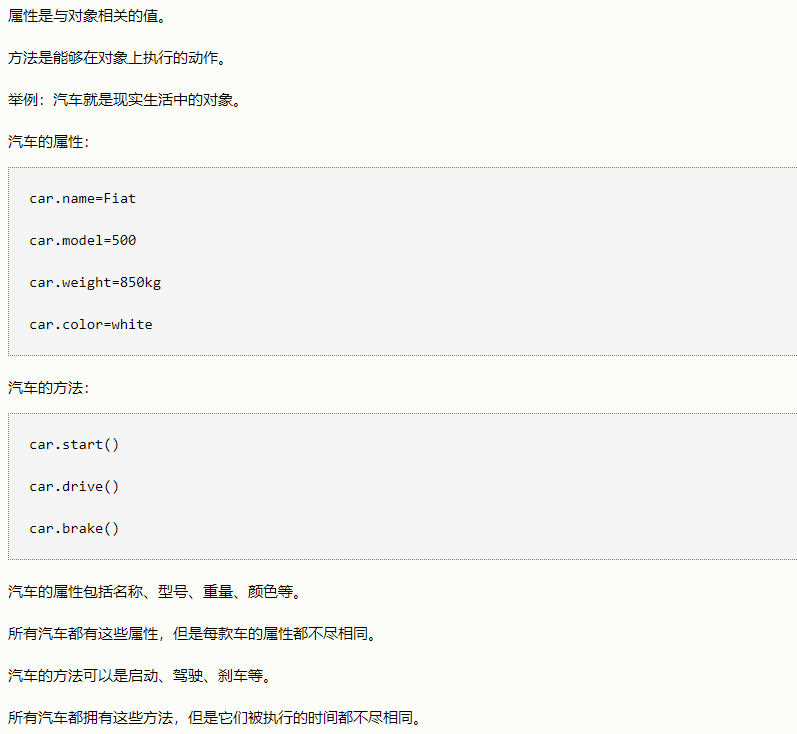




## JavaScript对象



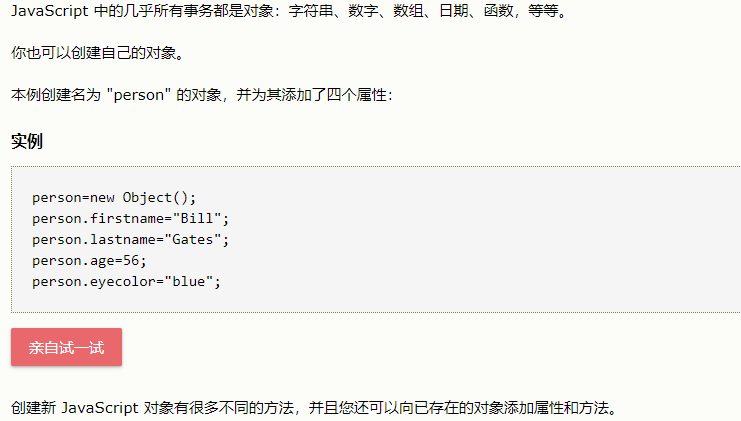
### 属性和方法



### JavaScript中的对象



### 创建JavaScript对象



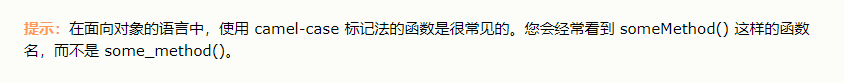
### 访问对象的属性



### 访问对象的方法



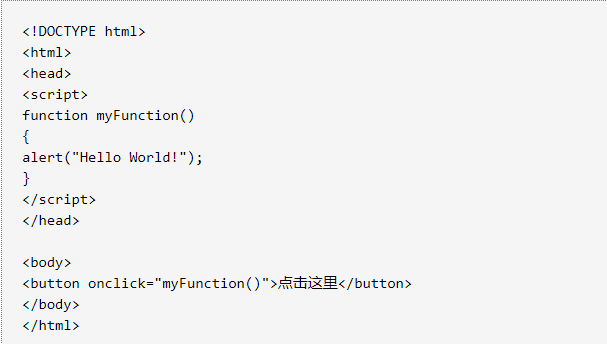
### 驼峰命名法



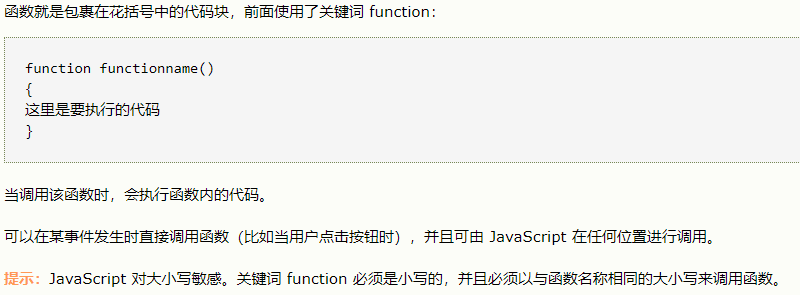
## JavaScript函数



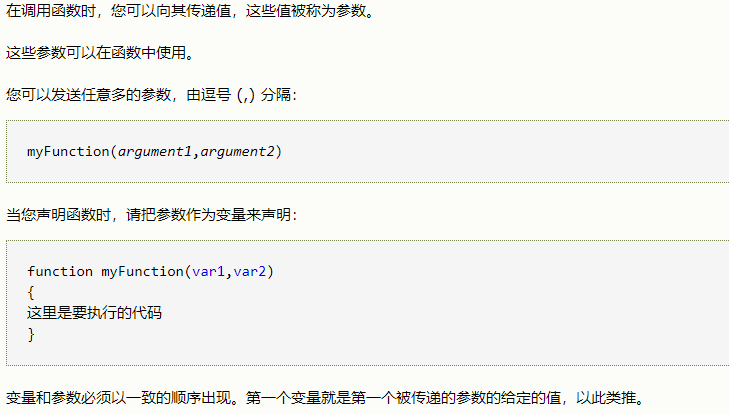
### 实例



### JavaScript函数语法



### 调用带参数的函数



实例：



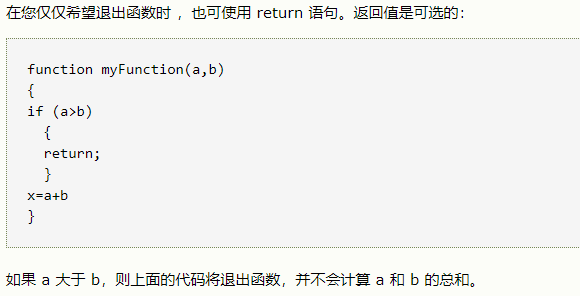
### 带有返回值的函数



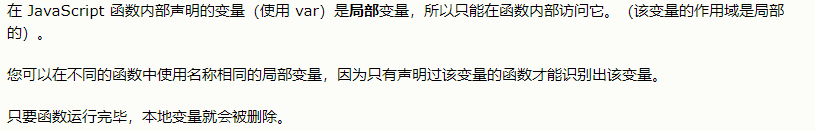
实例：



Return也可以用来退出函数：



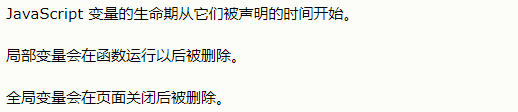
### 局部JavaScript变量



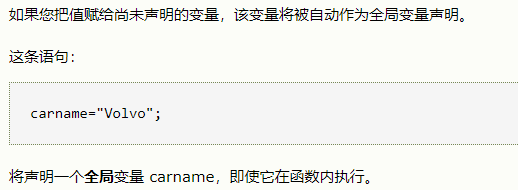
### 全局JavaScript变量



### JavaScript变量的生存期

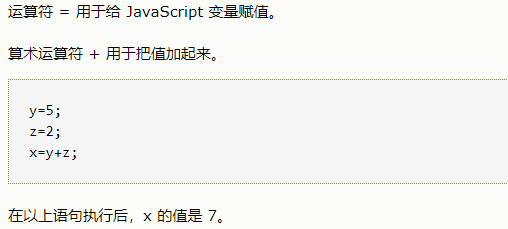


### 向未声明的JavaScript变量来分配值



## 十一、JavaScript运算符





### JavaScript算数运算符



### JavaScript赋值运算符



### 用于字符串的+运算符



### 对字符串和数字进行加法运算

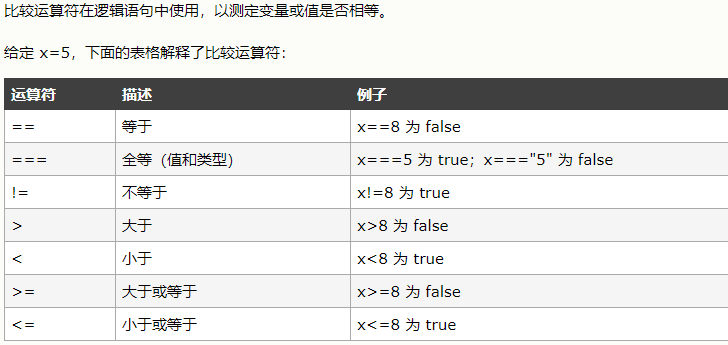




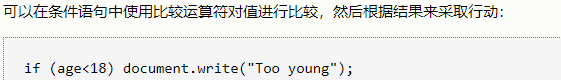
## 十二、JavaScript比较和逻辑运算符



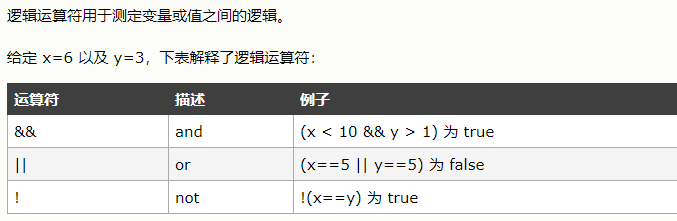
### 比较运算符



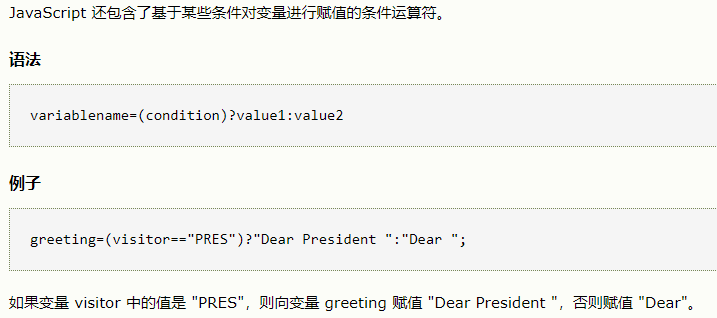
### 如何使用



### 逻辑运算符



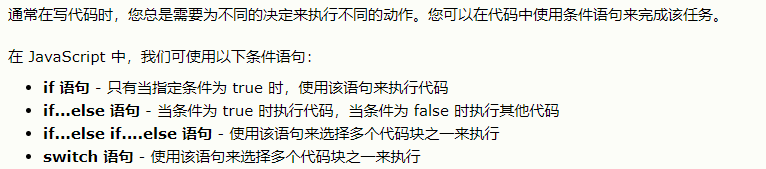
### 条件运算符



## 十三、JavaScript if…else语句



### 条件语句



### if语句



### if…else语句



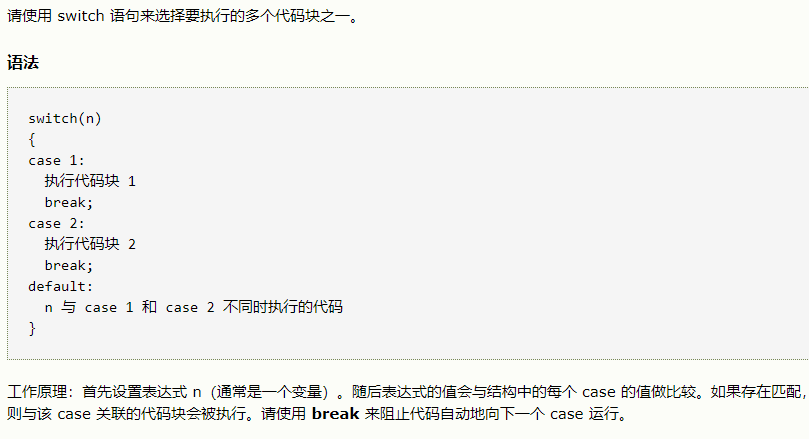
### if…else if…else语句



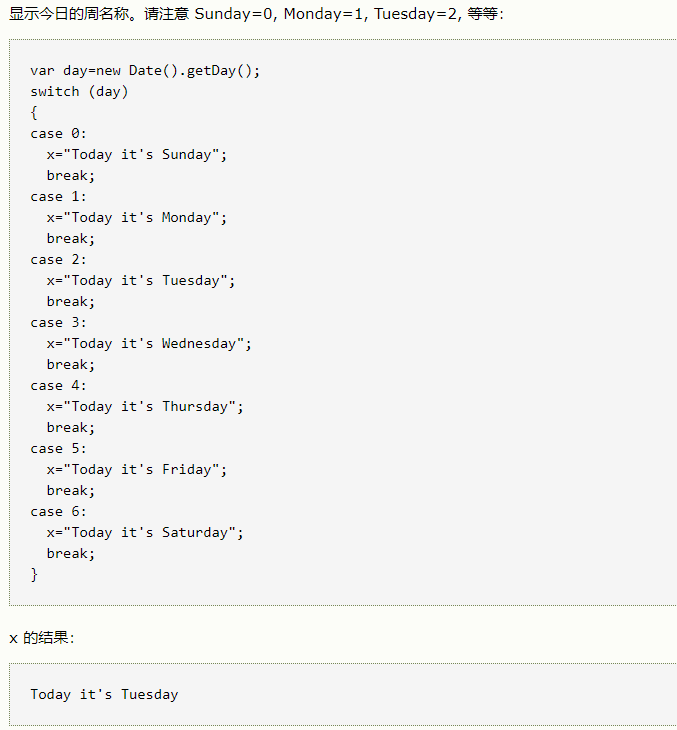
## 十四、JavaScript switch语句



### JavaScript switch语句



### JavaScript switch实例



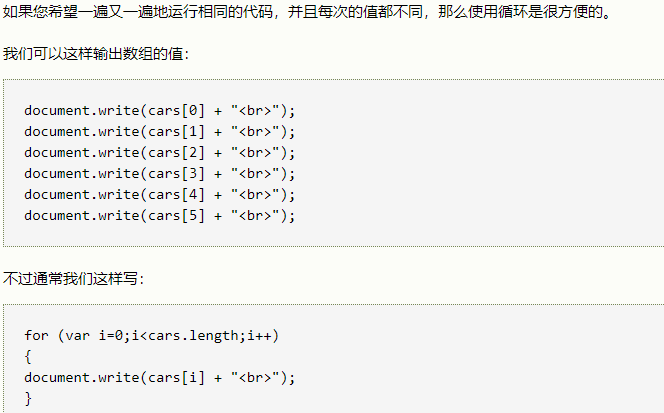
### default关键词



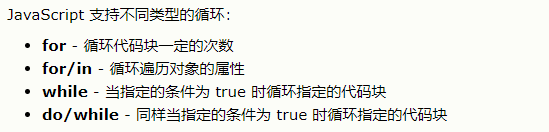
## 十五、JavaScript for循环



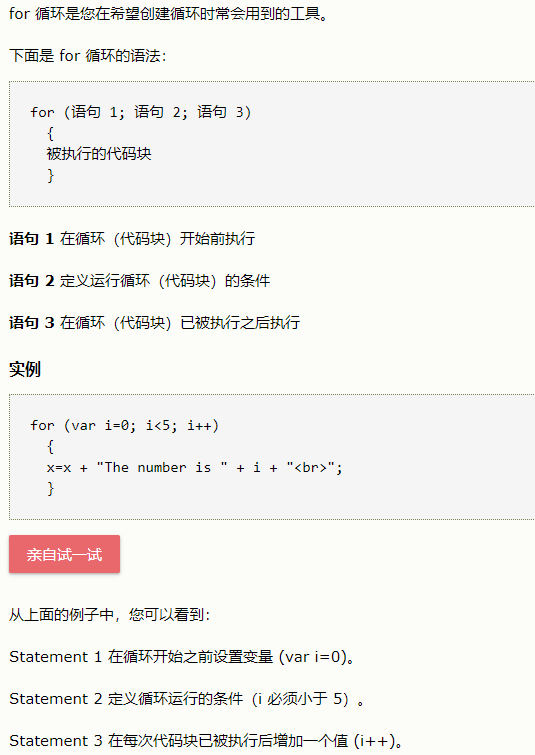
### JavaScript循环



### 不同类型的循环



### for循环



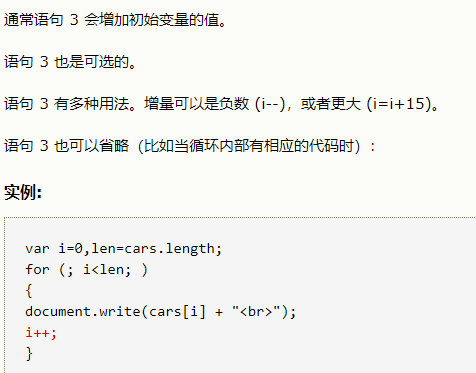
#### 语句1



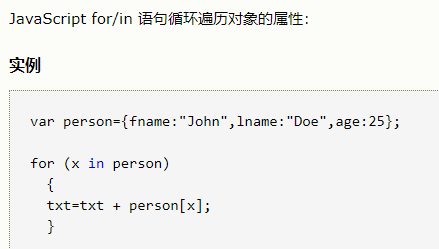
#### 语句2



#### 语句3



### 4、for/in循环



## 十六、JavaScript while循环



### while循环



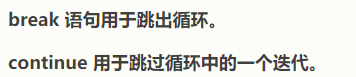
### do/while 循环



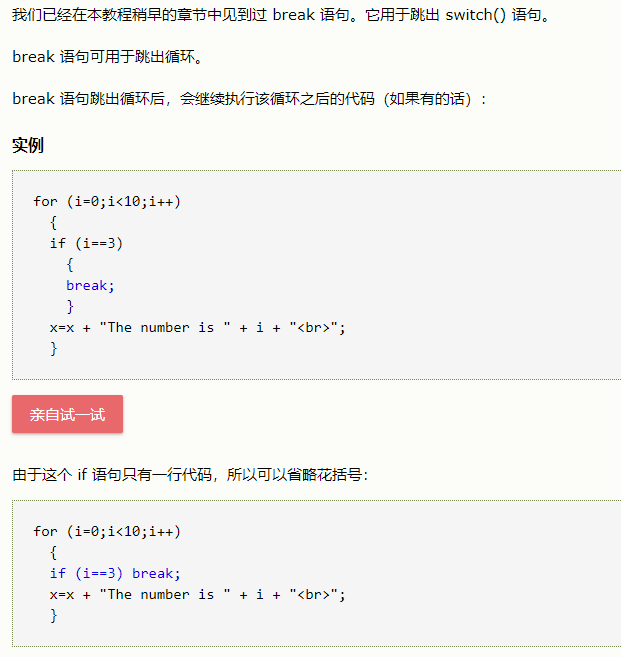
### 比较for循环和while循环



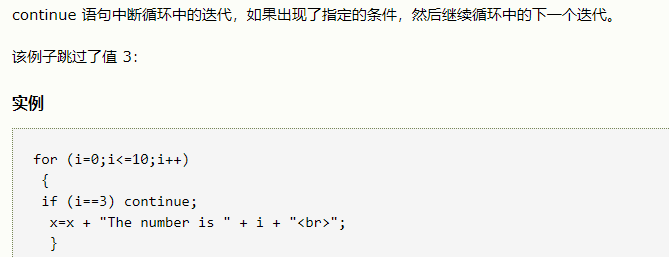
## 十七、JavaScript break和continue



### break语句



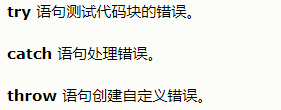
### continue语句



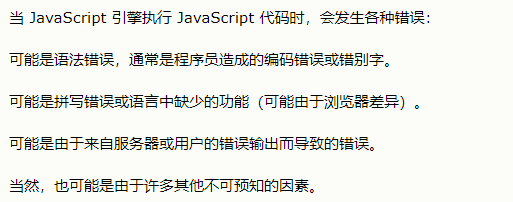
### JavaScript标签



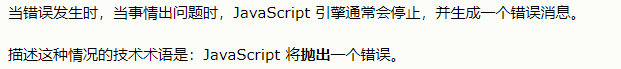
## 十八、JavaScript错误-Throw、try、catch



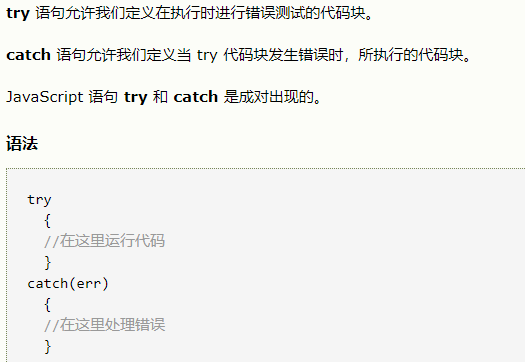
### 错误一会发生



### JavaScript抛出错误



### JavaScript测试和捕捉

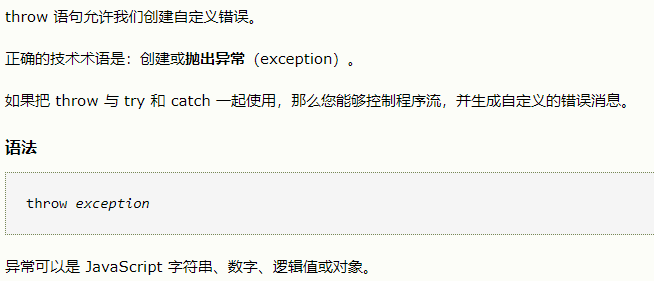


实例：

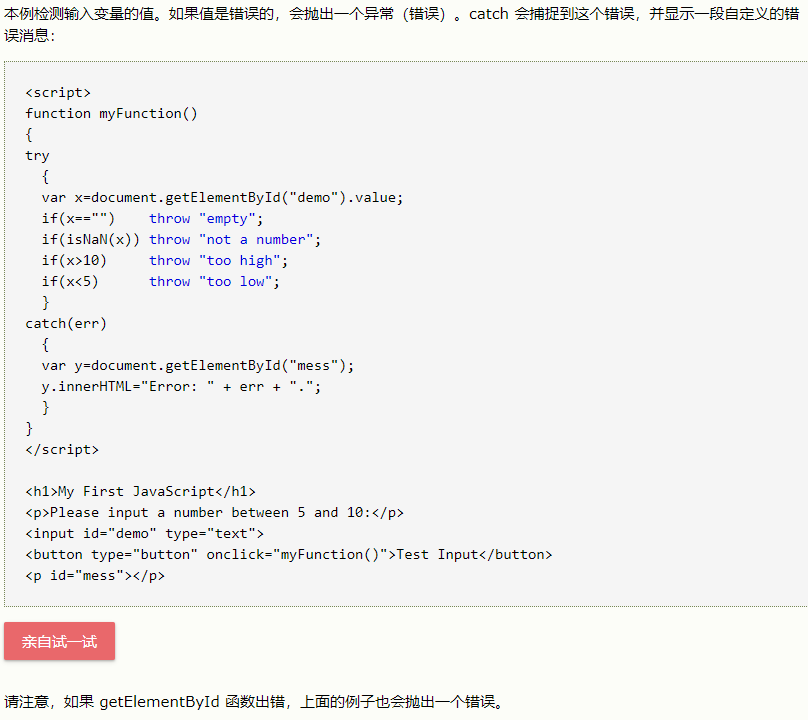


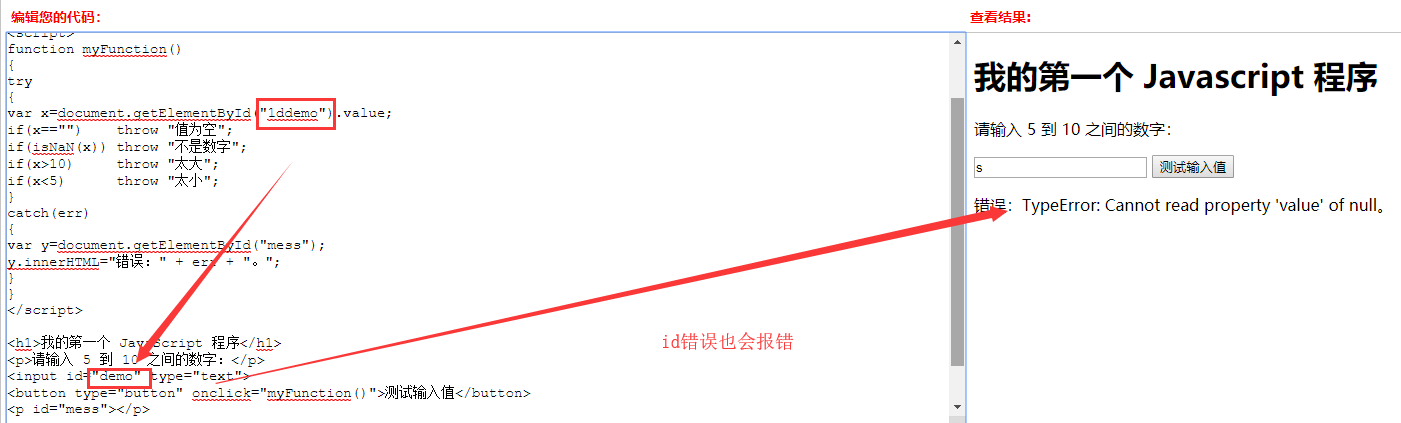


### Throw语句



实例：

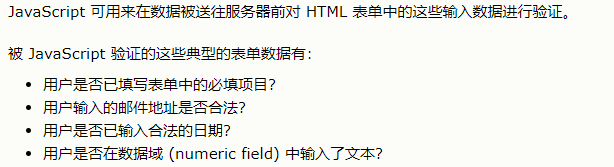




## 十九、JavaScript表单验证



### 1、JavaScript表单验证



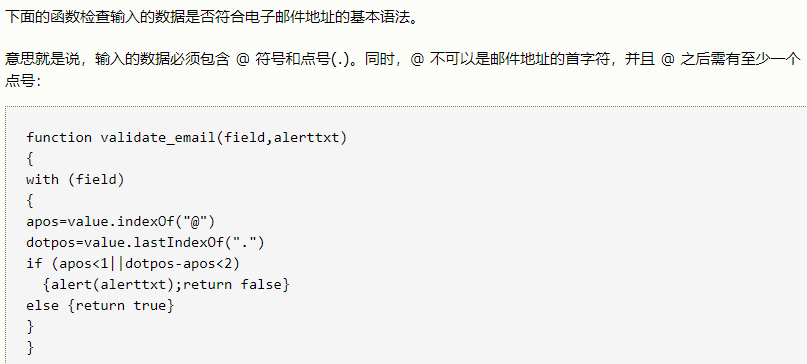
### 必填（必选）项目



下面是连同HTML表单的代码：



### Emile验证



下面是连同HTML表单的代码：



# JavaScript HTML DOM

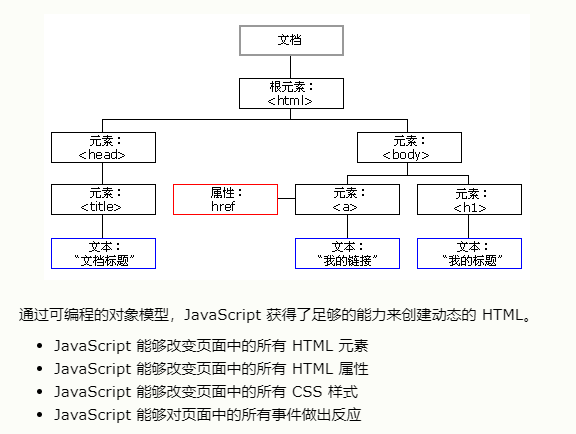
## 一、DMO简介



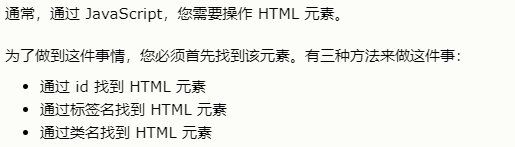
### 1、HTML DOM（文档对象模型）

当网页被加载时，浏览器会创建页面的文档对象模型（Document Object Model）。HTML DOM 模型被构造为对象的树。

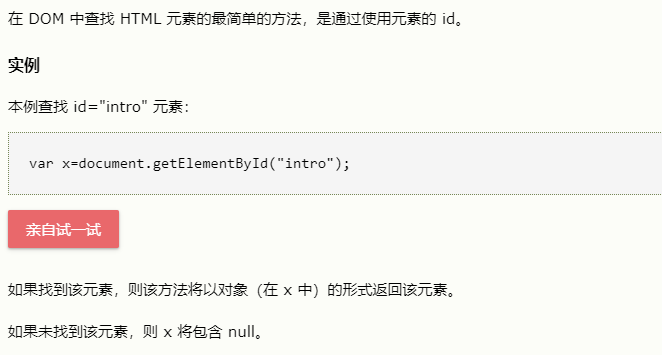
HTML DOM树：



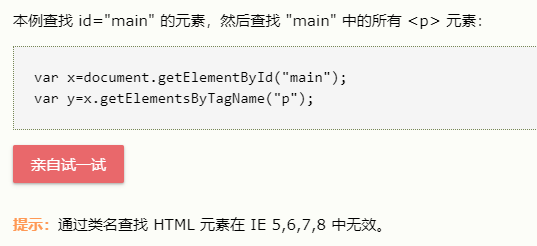
### 2、查找HTML元素



### 3、通过id查找HTML元素



### 4、通过标签名查找HTML元素



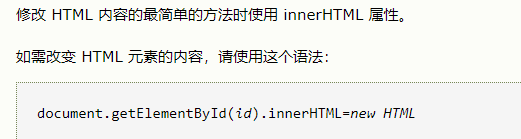
## 二、JavaScript HTML DOM-改变HTML



### 1、改变HTML输出流



### 2、改变HTML内容



实例：





### 3、改变HTML属性



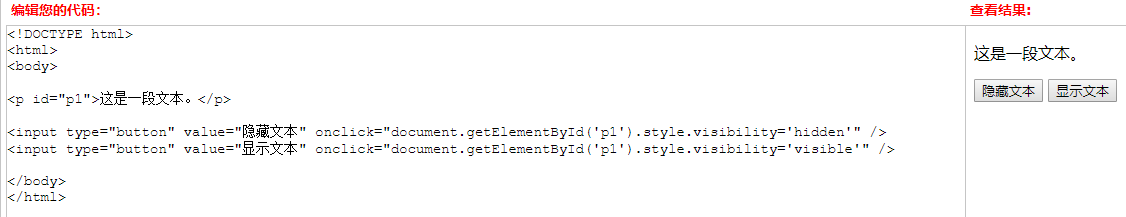
## 三、JavaScript HTML DOM- 改变css



### 1、改变HTML样式



### 2、显示/隐藏元素



## 四、JavaScript HTML DOM事件



### 1、对事件做出反应





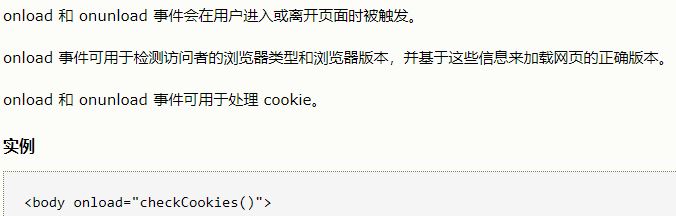
### 2、HTML事件属性

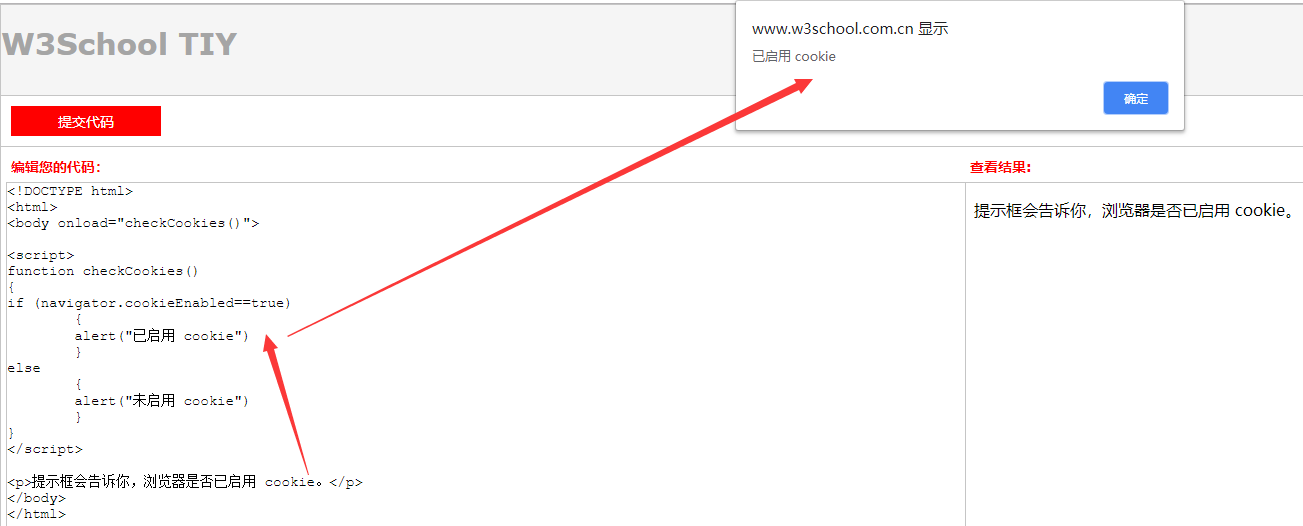


### 3、使用HTML DOM来分配事件

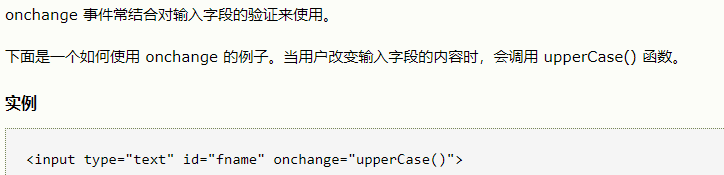


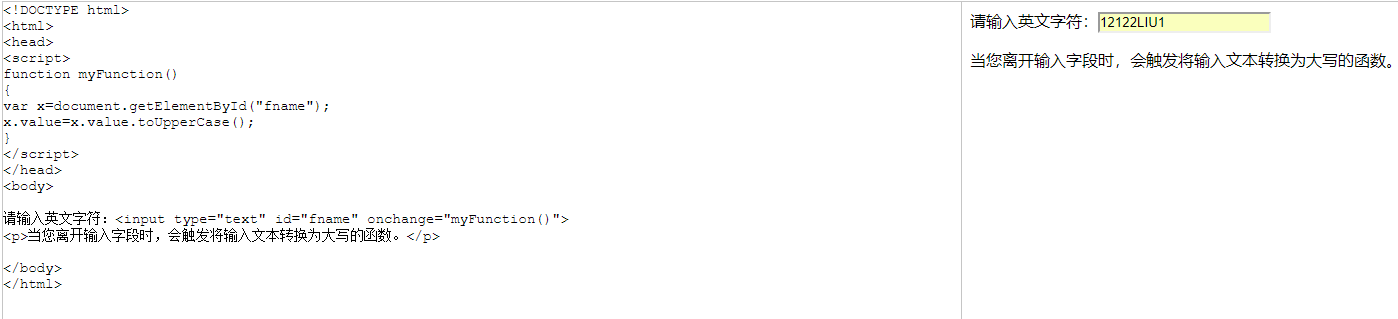
### 4、onload和onunload事件



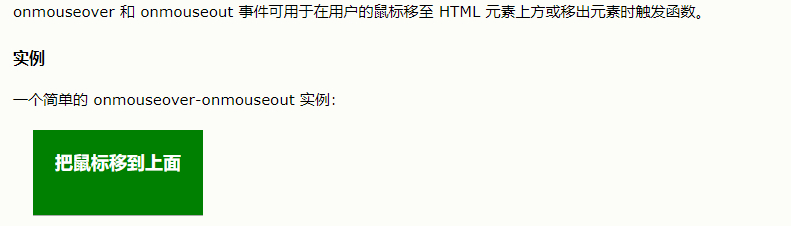


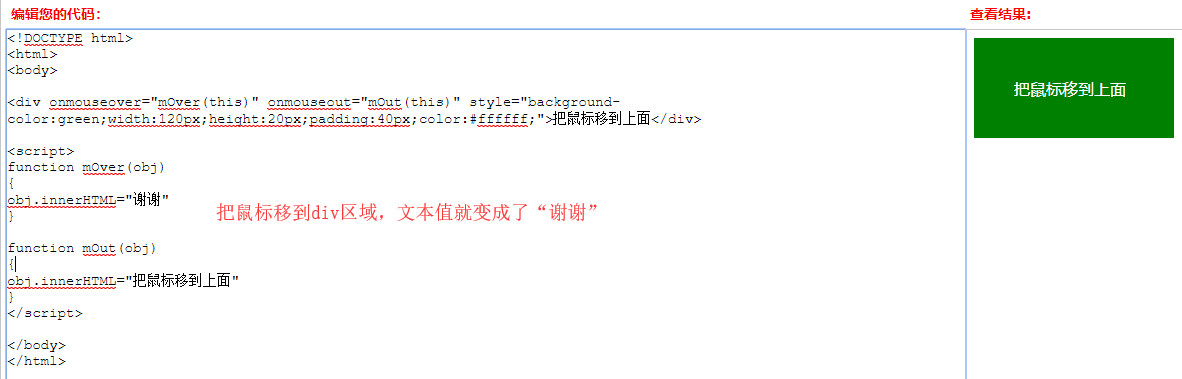
### 5、onchange事件





### 6、onmouseover和onmouseout事件



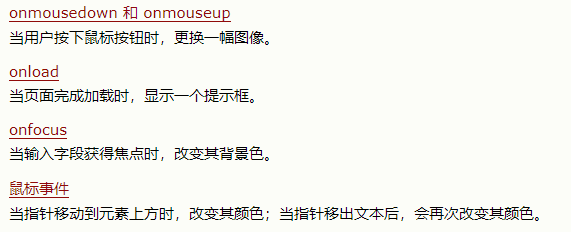


### 7、onmousedown、onmouseup和onclick事件





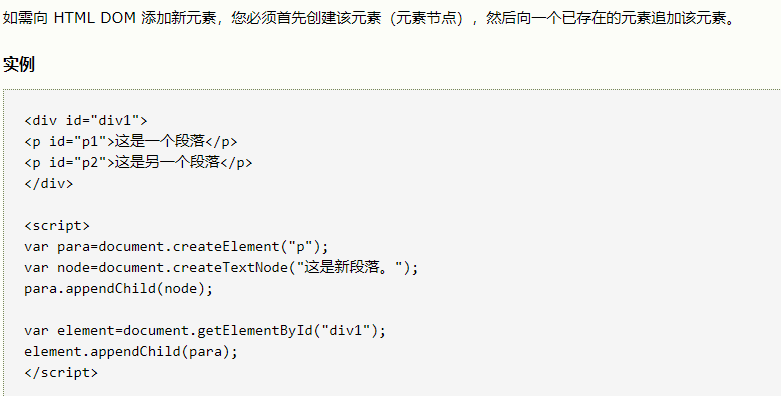
### 8、更多实例

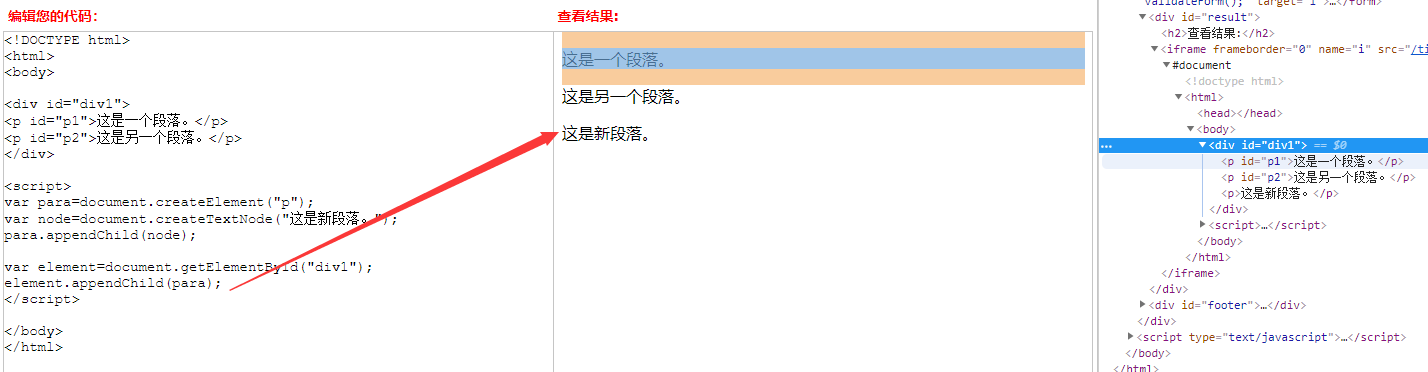


## 五、JavaScript HTML DOM 元素（节点）



### 1、创建新的HTML元素

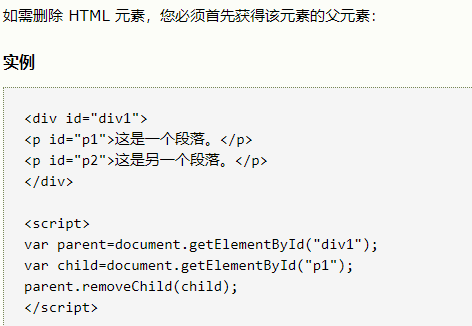




例子解释：



### 2、删除已有的HTML元素

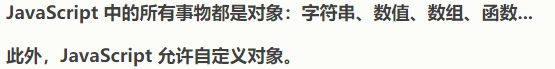


例子解释：

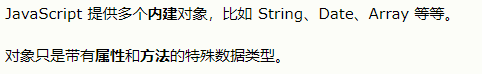


# JavaScript对象

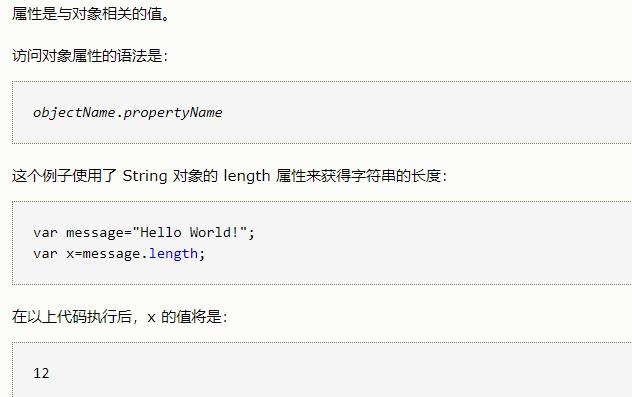
## 一、JavaScript对象



### 1、JavaScript对象



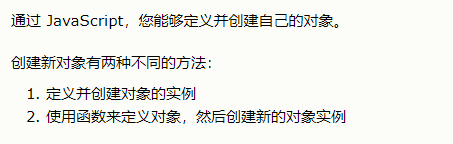
### 2、访问对象的属性



### 3、访问对象的方法



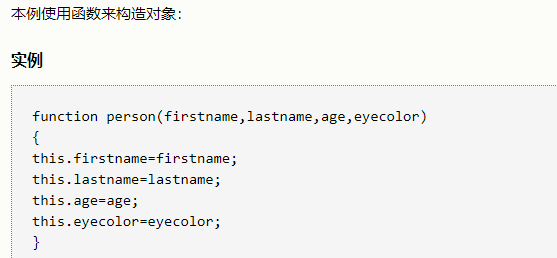
### 4、创建JavaScript对象



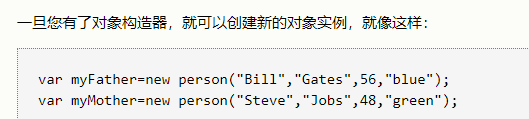
### 5、创建直接的实例



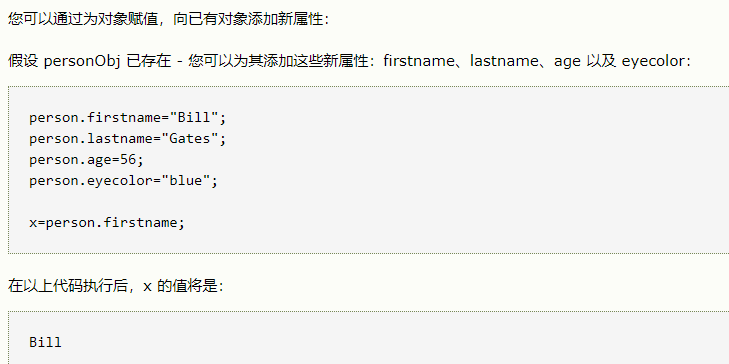
### 6、使用对象构造器



### 7、创建JavaScript对象实例



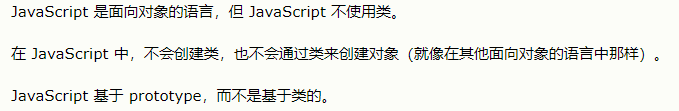
### 8、把属性添加到JavaScript对象



### 9、把方法添加到JavaScript对象



### 10、JavaScript类



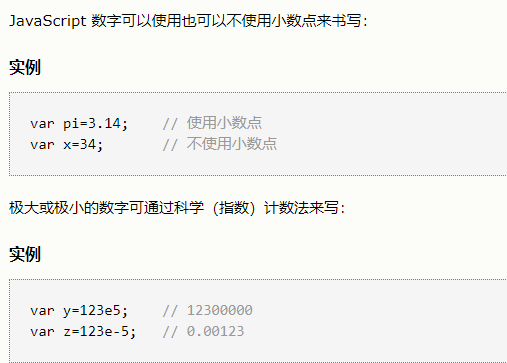
### 11、JavaScript for…in 循环



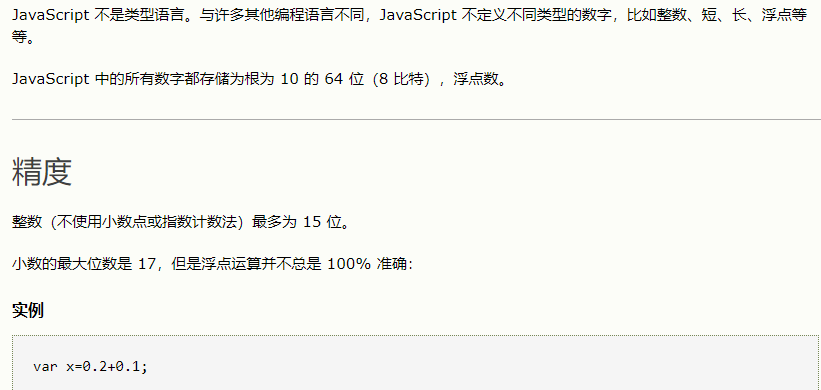
## 二、JavaScript Number对象（数字）

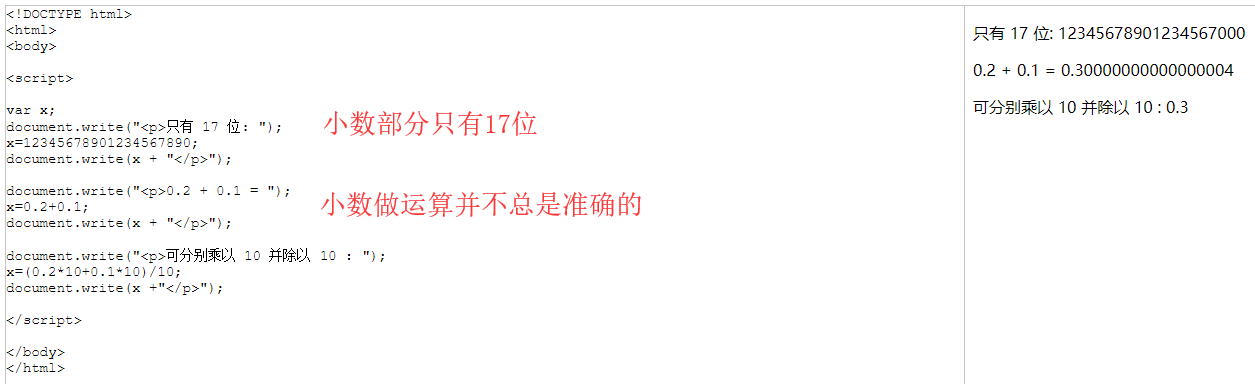


### 1、JavaScript数字

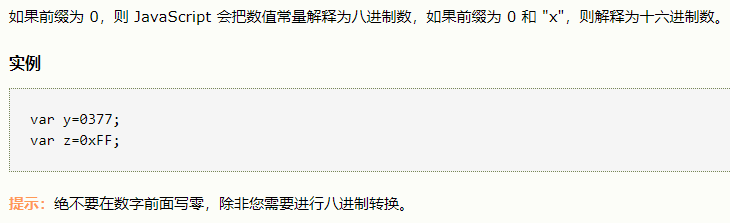


### 2、所有JavaScript数字均为64位

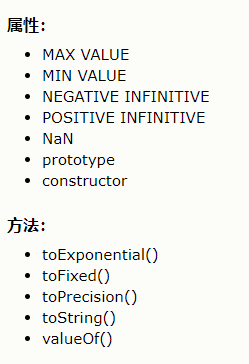




### 3、八进制和十六进制



### 4、数字属性和方法

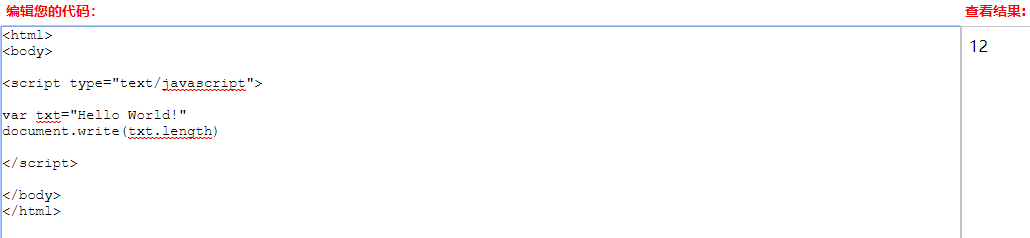


### 三、JavaScript String （字符串）对象

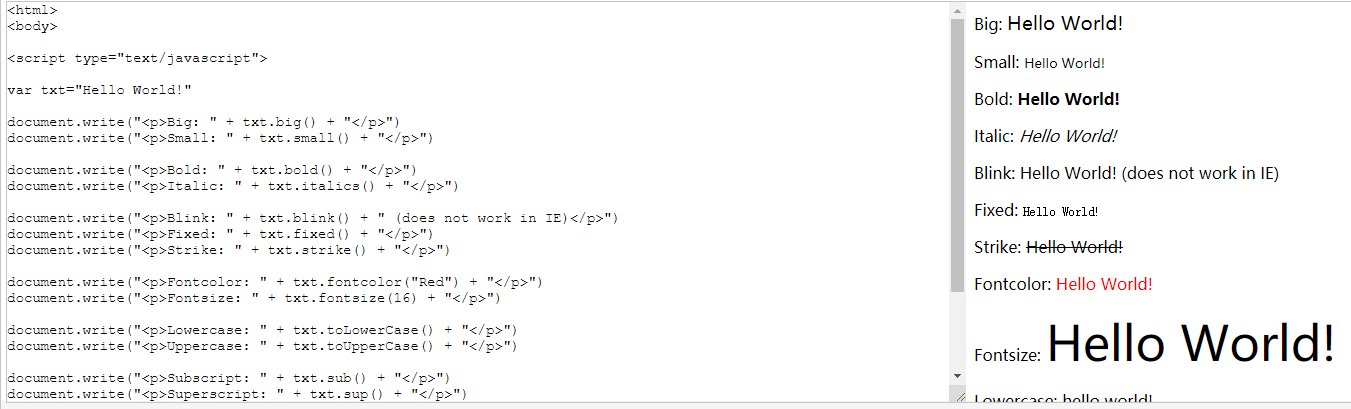


### 1、JavaScript字符串对象实例

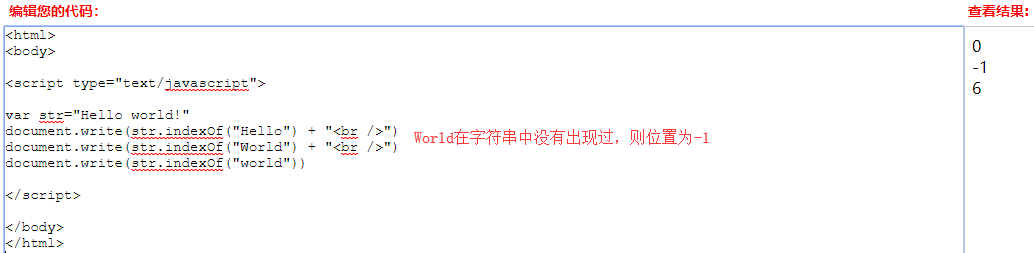
#### （1）使用长度属性计算字符串的长度



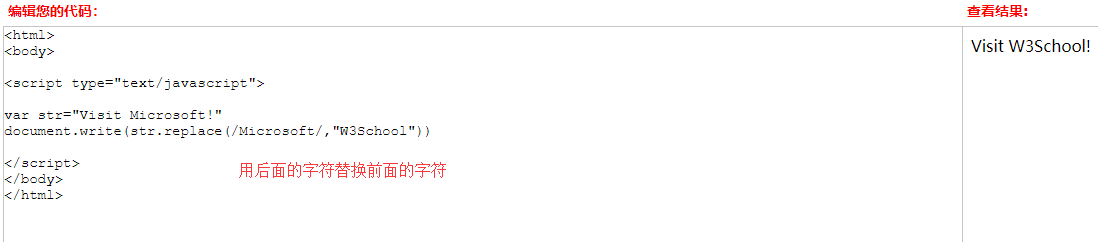
#### （2）为字符串添加样式



#### （3）indexOf（）方法定位字符第一次出现的位置



### （4）用replace（）方法在字符串中替换某些字符



### 2、字符串对象

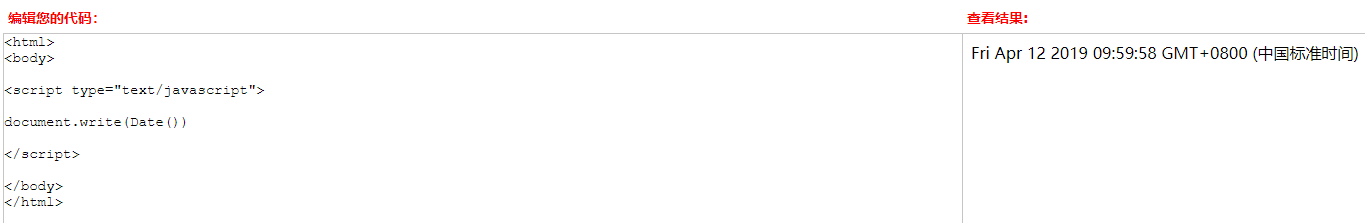


## 四、JavaScript Date（日期）对象

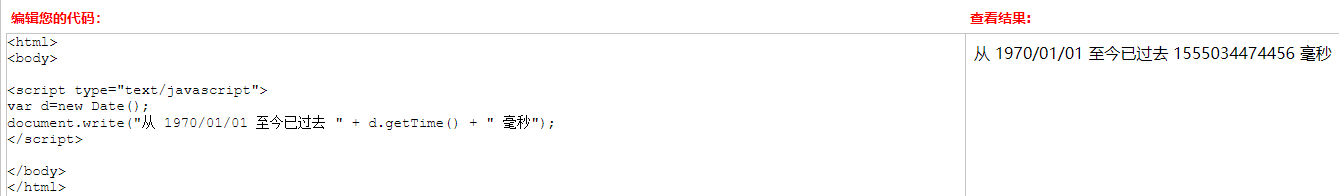


### 1、JavaScript 日期对象实例

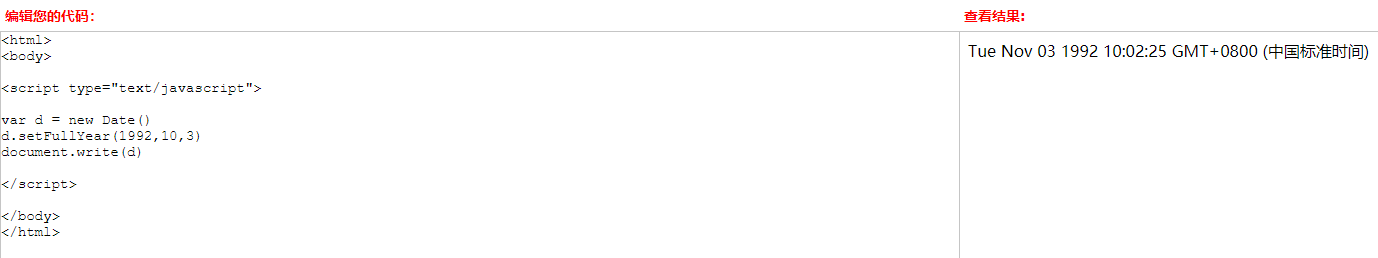
#### （1）使用Date（）方法获得当日的日期



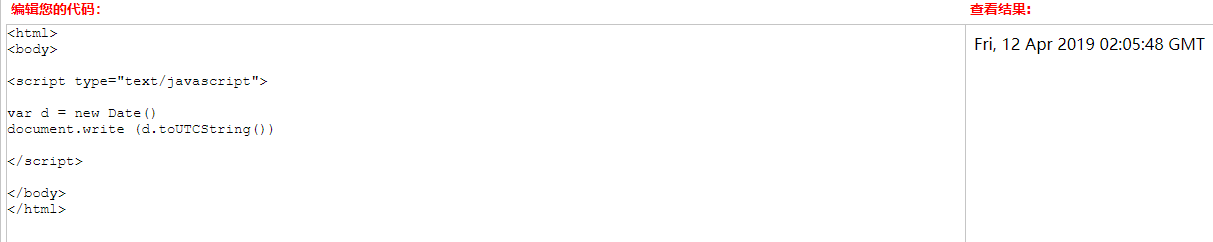
#### （2）getTime() 返回从 1970 年 1 月 1 日至今的毫秒数



#### （3）使用 setFullYear() 设置具体的日期

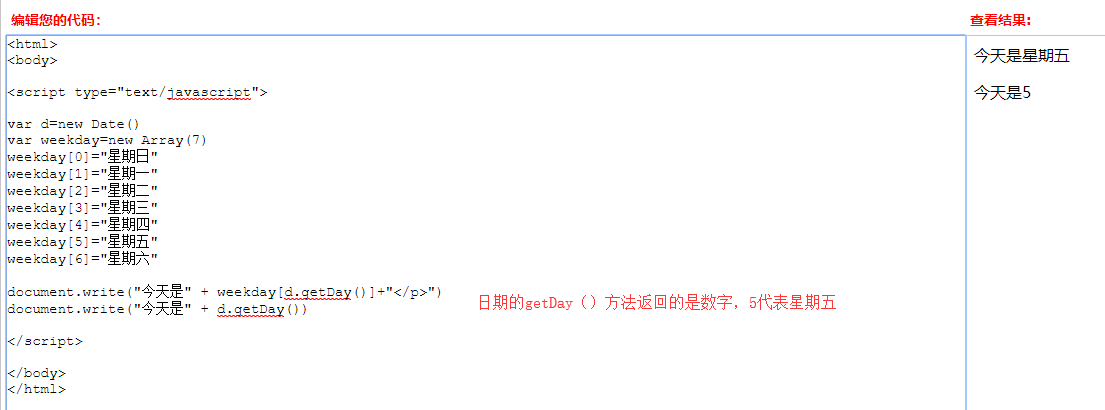


#### （4）使用 toUTCString() 将当日的日期（根据 UTC）转换为字符串

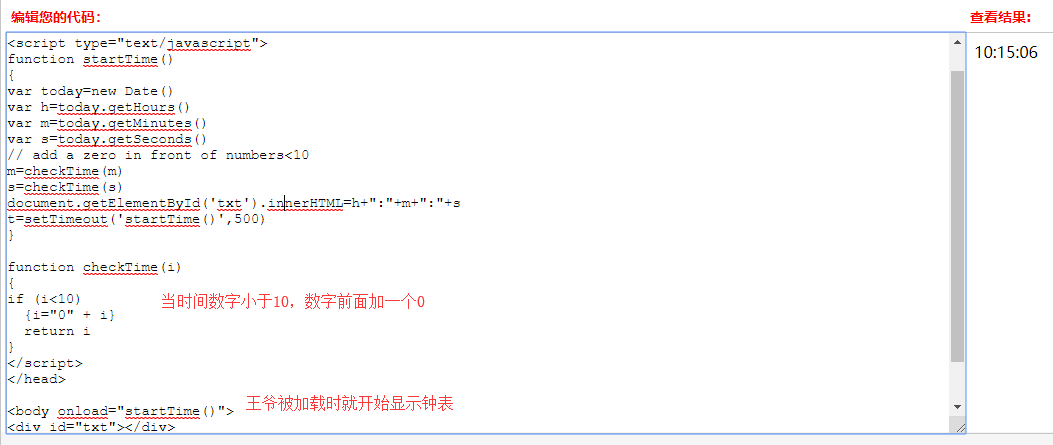


#### （5）使用 getDay() 和数组来显示星期，而不仅仅是数字

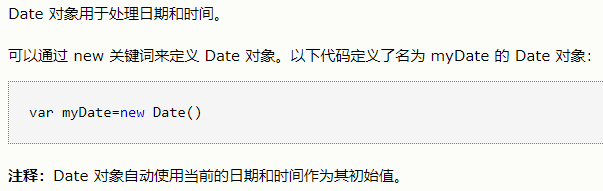




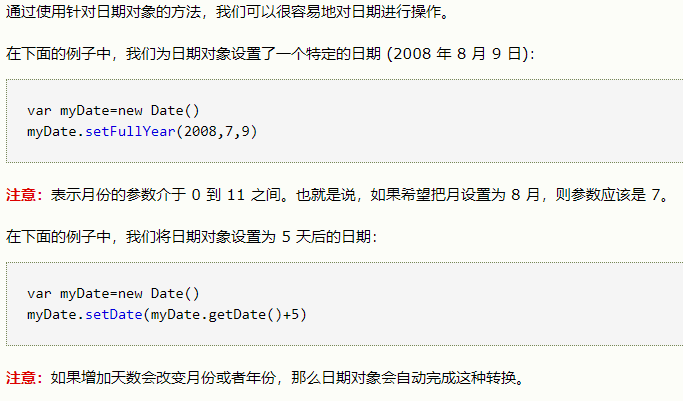
#### （6）在网页上显示一个钟表



### 2、定义日期



### 3、操作日期



### 4、比较日期

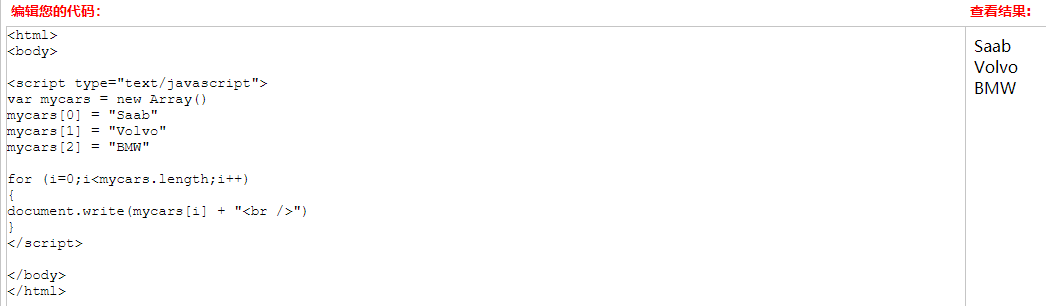


## 五、JavaScript Array（数组）对象

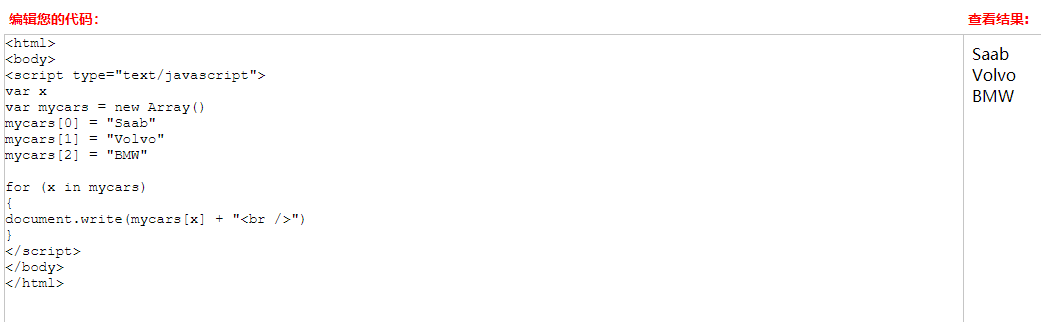


### 1、实例

#### （1）创建数组



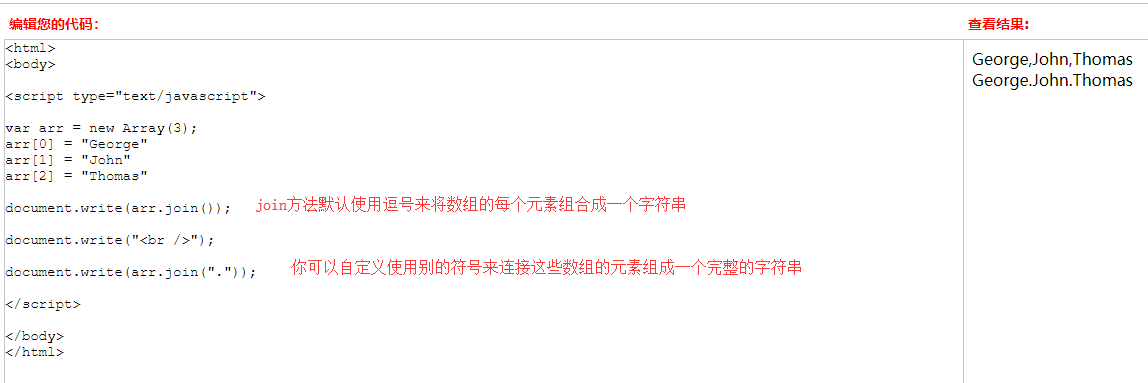
#### （2）使用 for...in 声明来循环输出数组中的元素



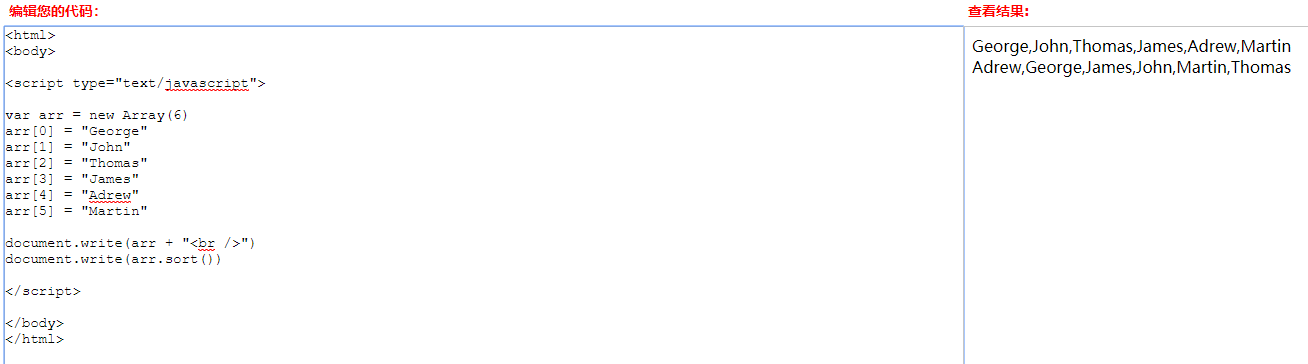
#### （3）使用 concat() 方法来合并两个数组



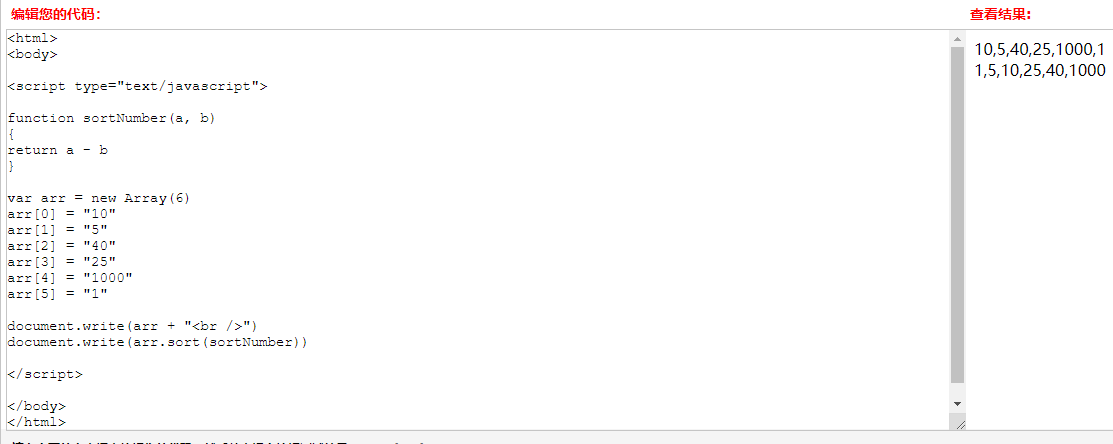
#### （4）使用 join() 方法将数组的所有元素组成一个字符串



#### （5）使用 sort() 方法从字面上对数组进行排序



#### （6）使用 sort() 方法从数值上对数组进行排序



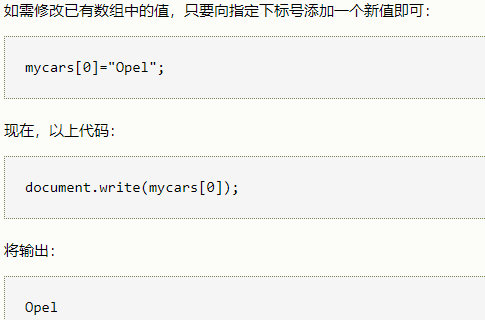
### 2、定义数组



### 3、访问数组



### 4、修改已有数组中的值



## 六、JavaScript Boolean（逻辑）对象



### 1、实例

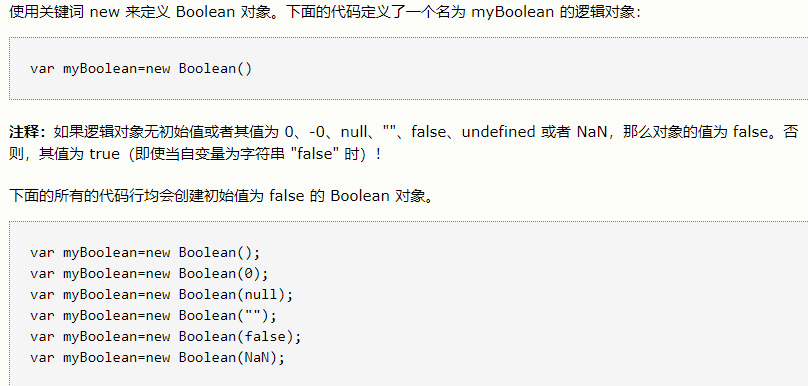


### 2、Boolean对象

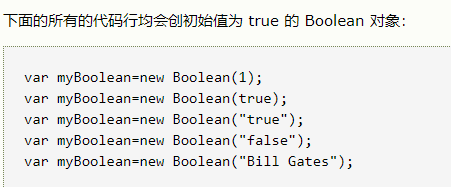
您可以将 Boolean 对象理解为一个产生逻辑值的对象包装器。

Boolean（逻辑）对象用于将非逻辑值转换为逻辑值（true 或者 false）。

#### （1）创建Boolean对象



#### （2）实例

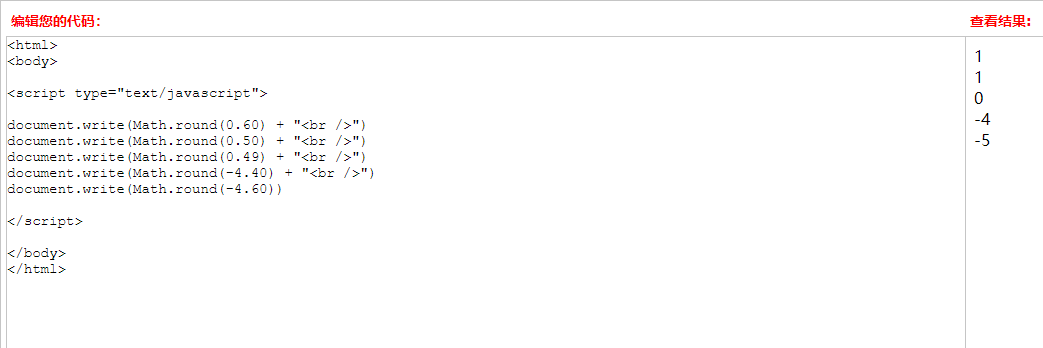


## 七、JavaScript Math（算术）对象

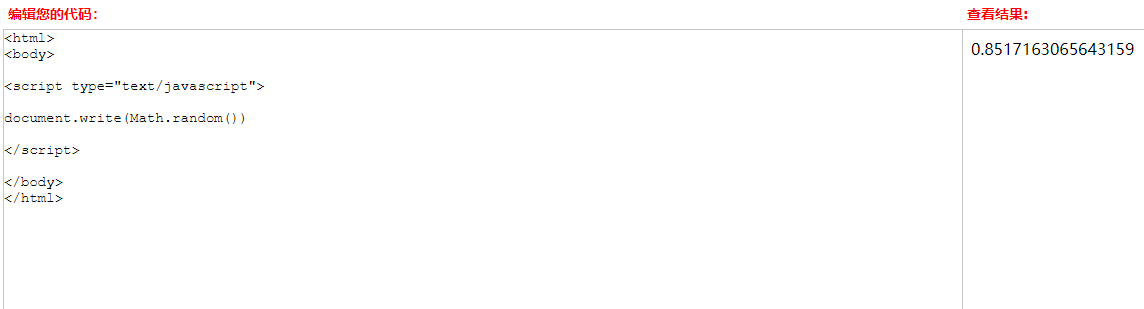


### 1、实例

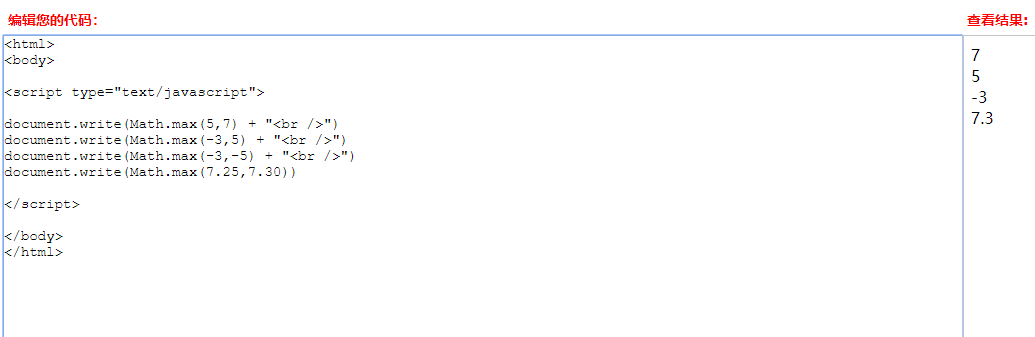
#### （1）round（）将数字四舍五入为最接近的整数



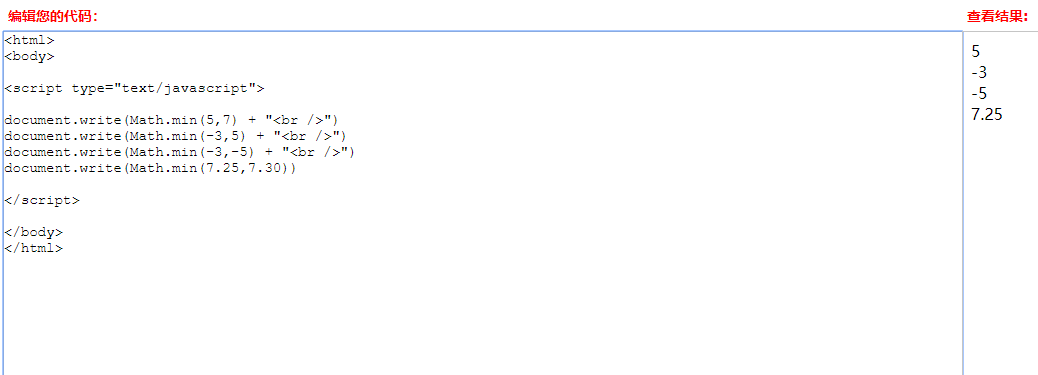
#### （2）使用 random() 来返回 0 到 1 之间的随机数



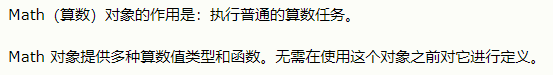
#### （3）使用 max() 来返回两个给定的数中的较大的数



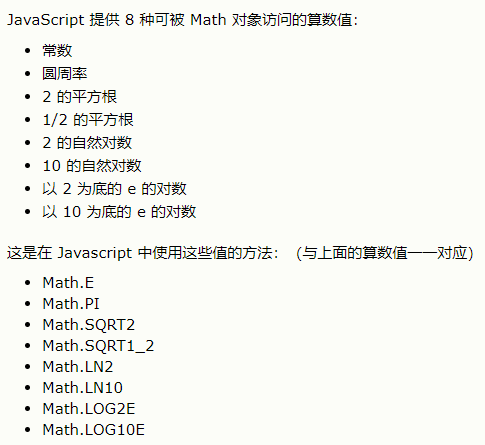
#### （4）使用 min() 来返回两个给定的数中的较小的数



### 2、Math对象



### 3、算数值



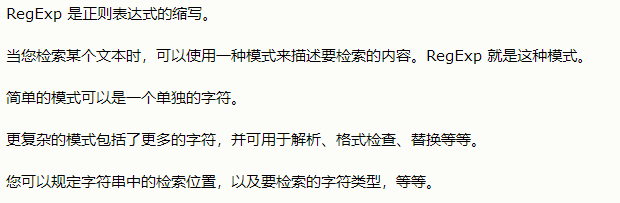
### 4、算数方法



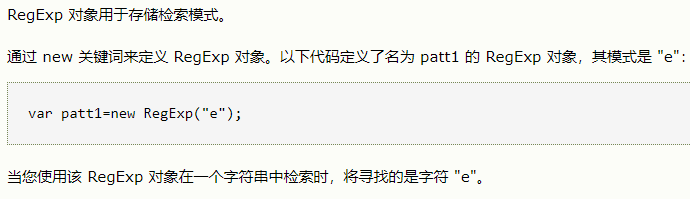
## 八、JavaScript RegExp对象（正则表达式）



### 1、什么是RegExp



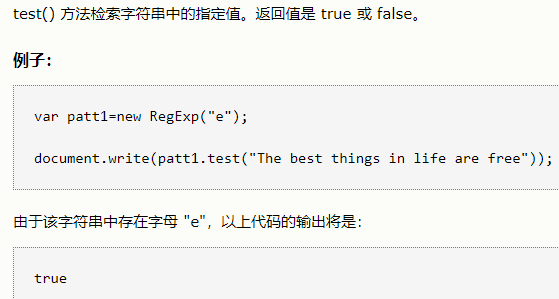
### 2、定义RegExp



### 3、RegExp对象的方法

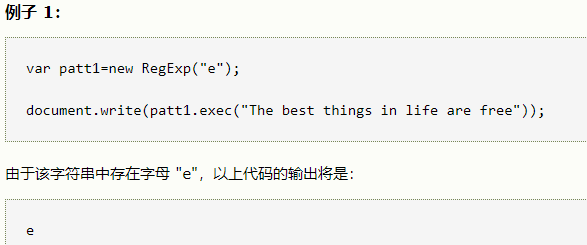
RegExp 对象有 3 个方法：test()、exec() 以及 compile()。

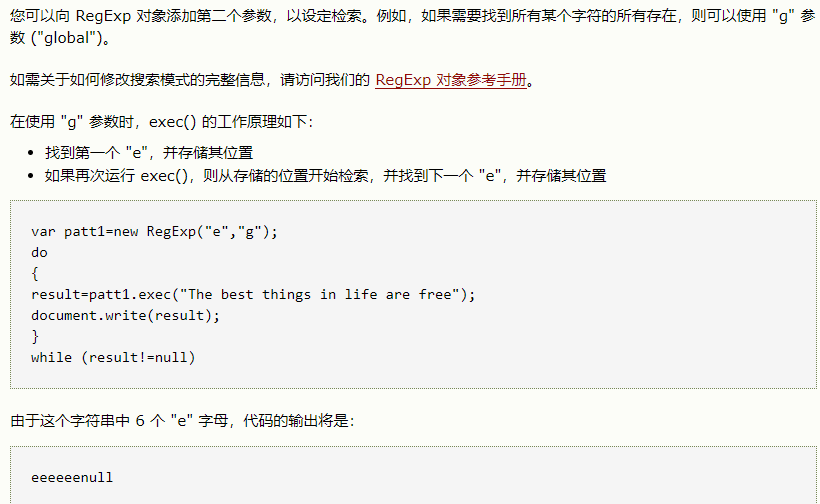
#### （1）test（）



#### （2）exec（）

exec() 方法检索字符串中的指定值。返回值是被找到的值。如果没有发现匹配，则返回 null





#### （3）compile（）



# JavaScript Window-浏览器对象模型

## 一、JavaScript Window-浏览器对象模型



### 1、浏览器对象模型（BOM）



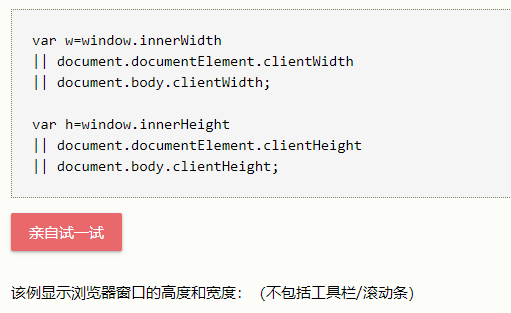
### 2、Windows对象



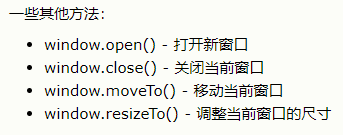
### 3、windows尺寸



实例：



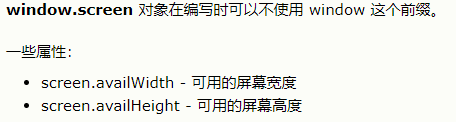
### 4、其他windows方法



## 二、JavaScript window Screen



### 1、window screen



### 2、window screen可用宽度



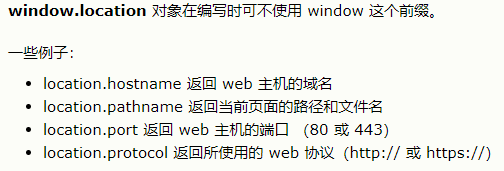
### 3、window screen可用高度



## 三、JavaScript Window Location



### 1、Window Location



### 2、Window Location Href



### 3、Window Location Pathname



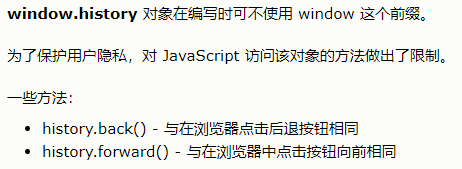
### 4、Window Location Assign



## 四、JavaScript Window Hostory



### 1、Window History



### 2、Window History Back



### 3、Window History Forward

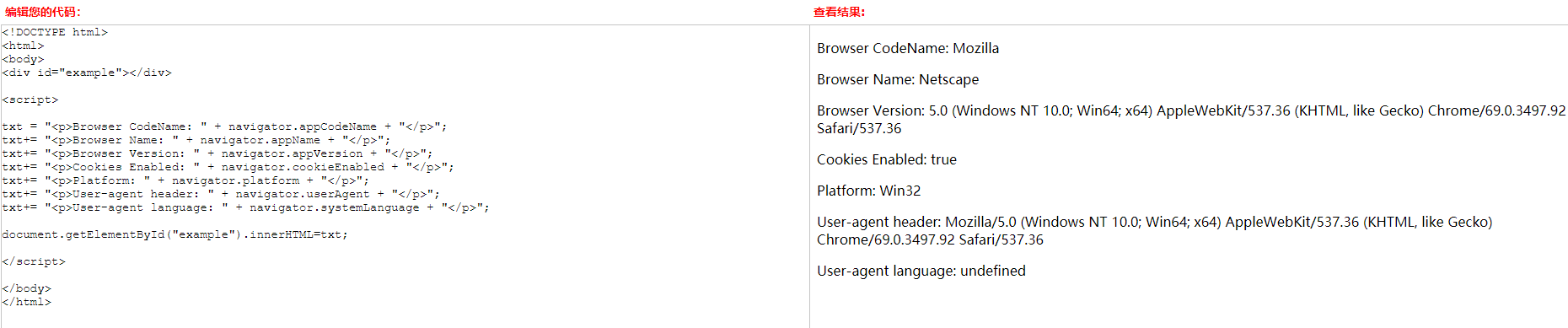


## 五、JavaScript Window Navigator



### 1、Window Navigator





### 2、检测



## 六、JavaScript PopupAlert（消息框）



### 1、实例

#### （1）警告框



#### （2）带有折行的警告框

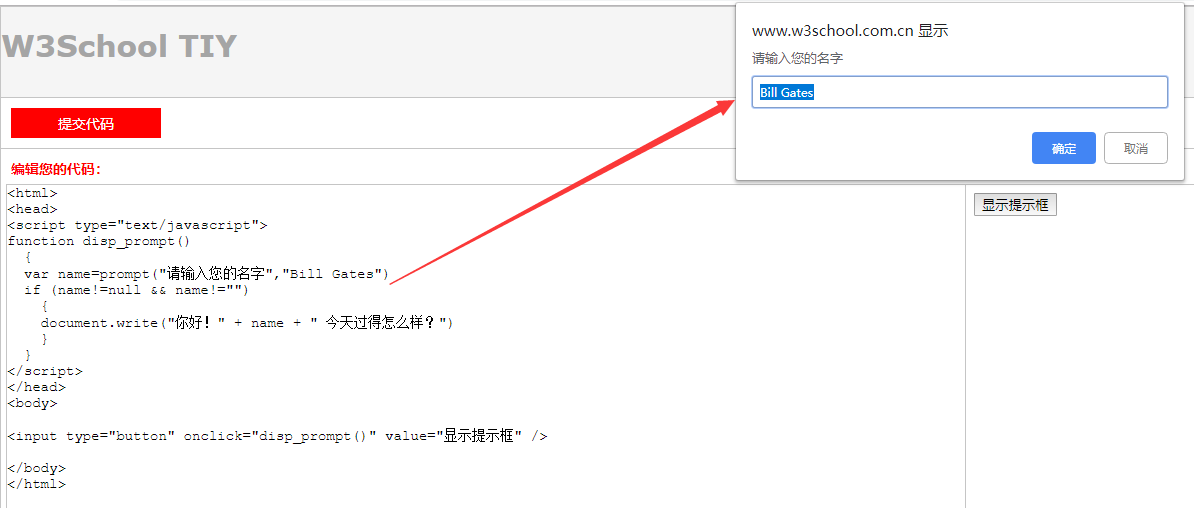


#### （3）确认框

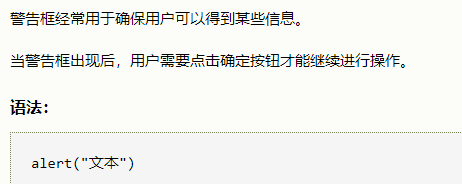
确认框实例化之后，该对象的值等于确认框的返回值，true或者false，即用户点击确认或者取消



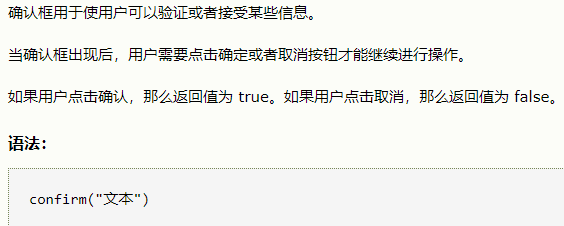
#### （4）提示框



### 2、警告框



### 3、确认框



### 4、提示框



## 七、JavaScript Timing（JavaScript 计时）

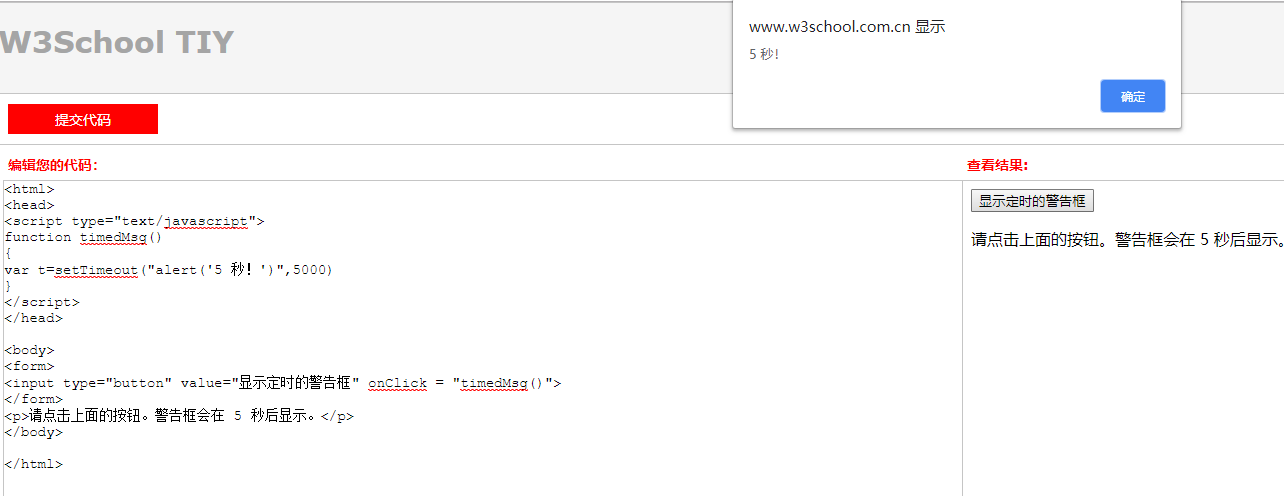


### 1、实例：

#### （1）简单实例

单击本例中的按钮后，会在 5 秒后弹出一个警告框

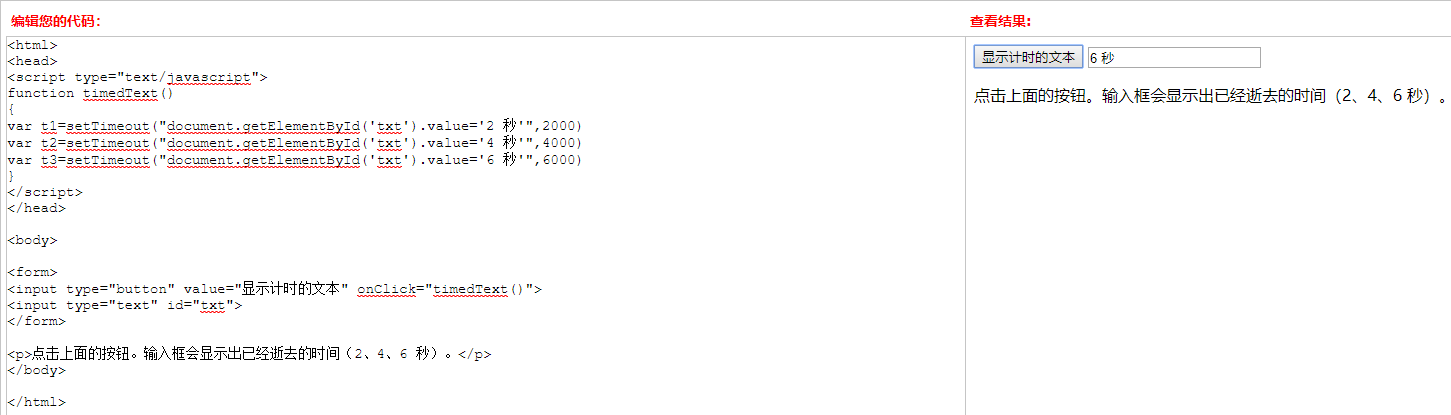
var t=setTimeout("alert('5 秒！')",5000)



#### （2）另一个简单的实例

本例中的程序会执行 2 秒、4 秒和 6 秒的计时。

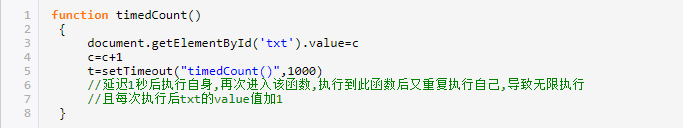
var t1=setTimeout("document.getElementById('txt').value='2 秒'",2000)



#### （3）在一个无穷循环中的计时事件

在本例中，单击开始计时按钮后，程序开始从 0 以秒计时。

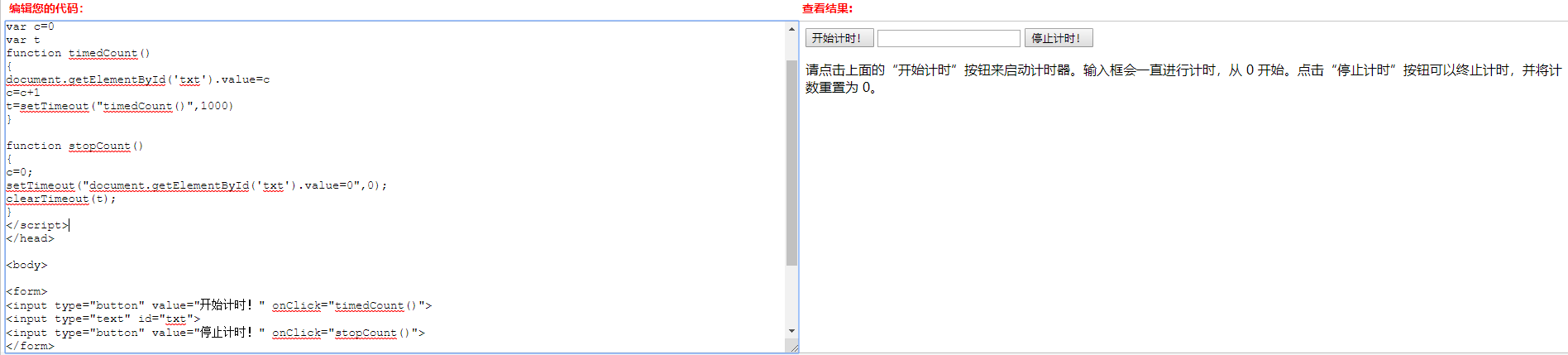




#### （4）带有停止按钮的无穷循环中的计时事件

在本例中，点击计数按钮后根据用户输入的数值开始倒计时，点击停止按钮停止计时。

clearTimeout(t); //停止计时

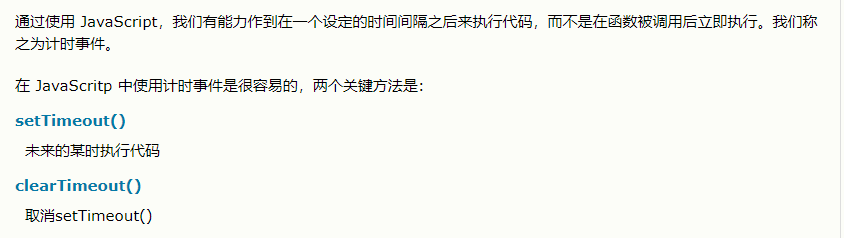


#### （5）使用计时事件制作的钟表

<http://www.w3school.com.cn/tiy/t.asp?f=jseg_timing_clock>

获取当前时间，并开始计时

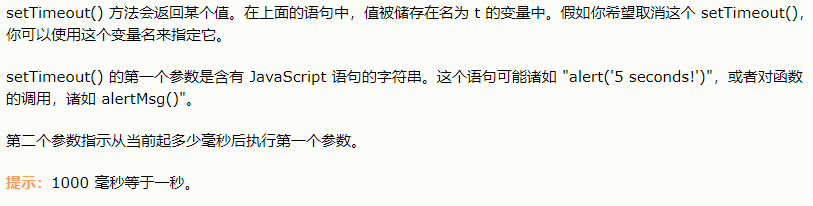
### 2、JavaScript计时事件



### 3、setTimeout（）

#### （1）语法

var t=setTimeout("javascript语句",毫秒)



#### （2）实例



### 4、clearTimeout（）

#### （1）语法

clearTimeout(setTimeout\_variable)

#### （2）实例



## 八、JavaScript cookies



### 1、实例

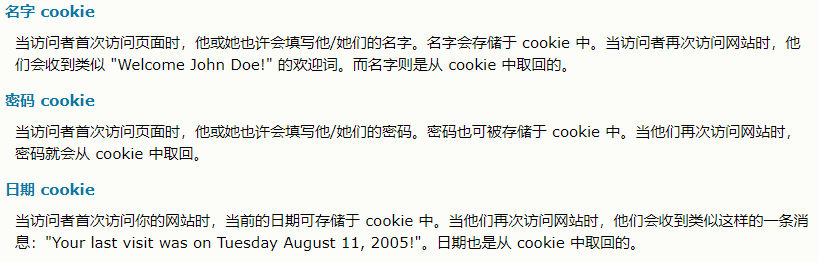
利用用户在提示框中输入的数据创建一个 JavaScript Cookie，当该用户再次访问该页面时，根据 cookie 中的信息发出欢迎信息。



### 2、什么是cookie

cookie 是存储于访问者的计算机中的变量。每当同一台计算机通过浏览器请求某个页面时，就会发送这个 cookie。你可以使用 JavaScript 来创建和取回 cookie 的值。

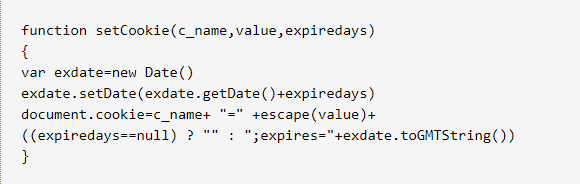
**有关cookie的例子：**



### 3、创建和存储cookie

在这个例子中我们要创建一个存储访问者名字的 cookie。当访问者首次访问网站时，他们会被要求填写姓名。名字会存储于 cookie 中。当访问者再次访问网站时，他们就会收到欢迎词。

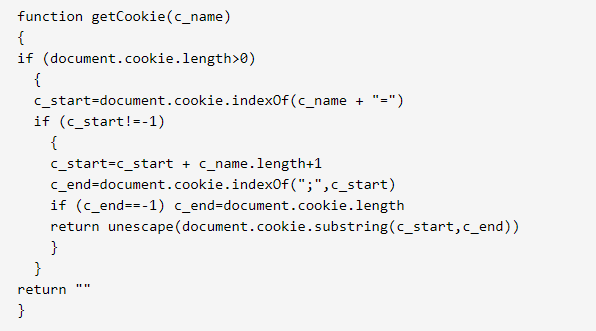
**首先，我们会创建一个可在 cookie 变量中存储访问者姓名的函数：**



上面这个函数中的参数存有 cookie 的名称、值以及过期天数。

在上面的函数中，我们首先将天数转换为有效的日期，然后，我们将 cookie 名称、值及其过期日期存入 document.cookie 对象。

**之后，我们要创建另一个函数来检查是否已设置 cookie：**



上面的函数首先会检查 document.cookie 对象中是否存有 cookie。假如 document.cookie 对象存有某些 cookie，那么会继续检查我们指定的 cookie 是否已储存。如果找到了我们要的 cookie，就返回值，否则返回空字符串。

**最后，我们要创建一个函数，这个函数的作用是：如果 cookie 已设置，则显示欢迎词，否则显示提示框来要求用户输入名字**。



这是所有的代码：

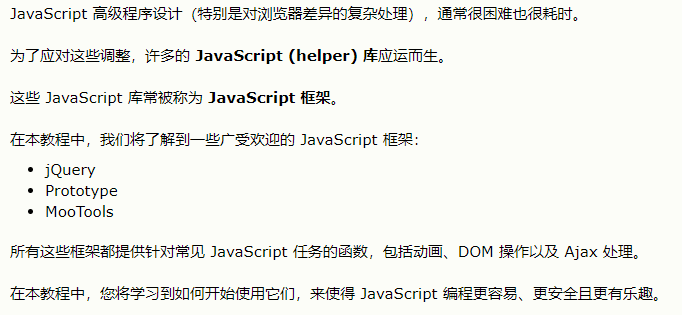
http://www.w3school.com.cn/js/js\_cookies.asp

# JavaScript 库

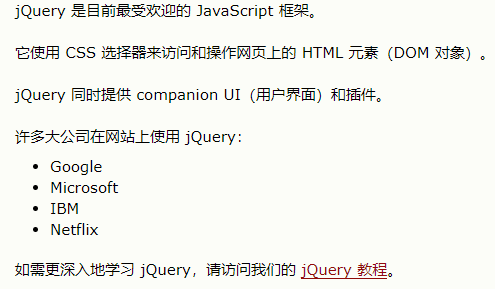
## 一、JavaScript 库



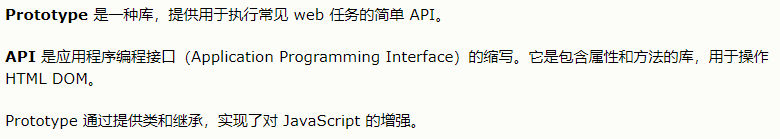
### 1、JavaScript 框架（库）



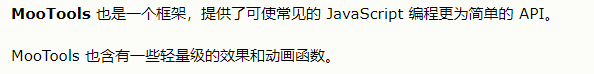
### 2、JQuery



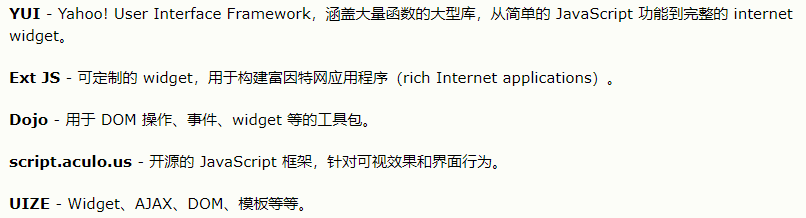
### 3、prototype



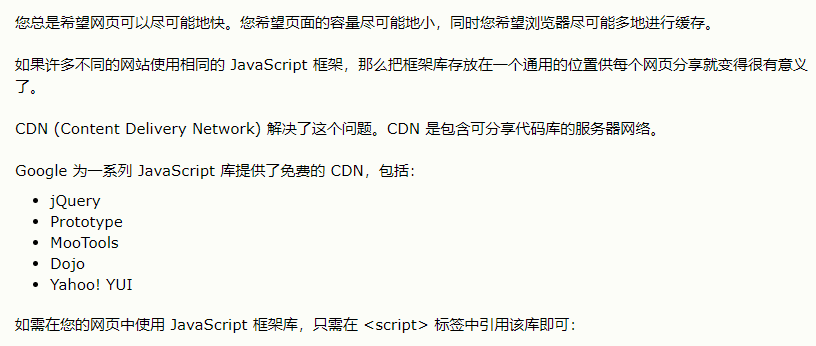
### 4、MooTools



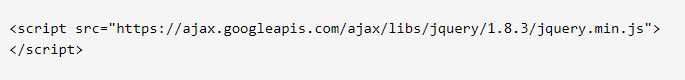
### 5、其他框架



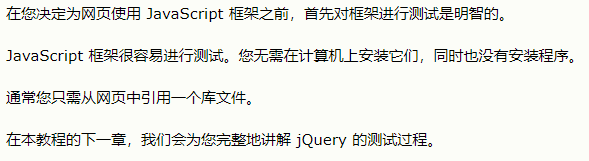
### 6、CDN内容分发网络



**引用JQuery：**



### 7、使用框架

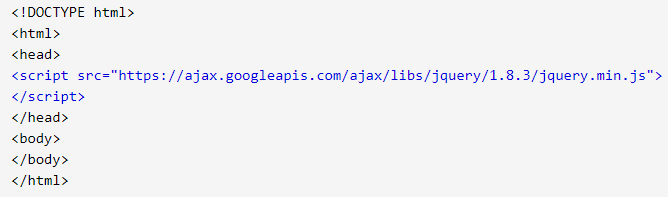


## 二、JavaScript ---测试JQuery

### 1、引用JQuery

如需测试 JavaScript 库，您需要在网页中引用它。

为了引用某个库，请使用 <script> 标签，其 src 属性设置为库的 URL：



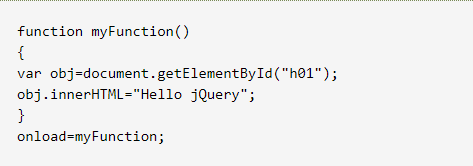
### 2、JQuery描述

主要的 jQuery 函数是 $() 函数（jQuery 函数）。如果您向该函数传递 DOM 对象，它会返回 jQuery 对象，带有向其添加的 jQuery 功能。

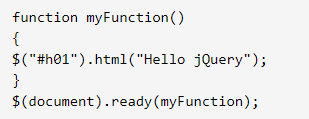
jQuery 允许您通过 CSS 选择器来选取元素。

**在 JavaScript 中，您可以分配一个函数以处理窗口加载事件：**

（1）JavaScript方式



（2）等价的JQuery方式



上面代码的最后一行，HTML DOM 文档对象被传递到 jQuery ：$(document)。

当您向 jQuery 传递 DOM 对象时，jQuery 会返回以 HTML DOM 对象包装的 jQuery 对象。

jQuery 函数会返回新的 jQuery 对象，其中的 ready() 是一个方法。

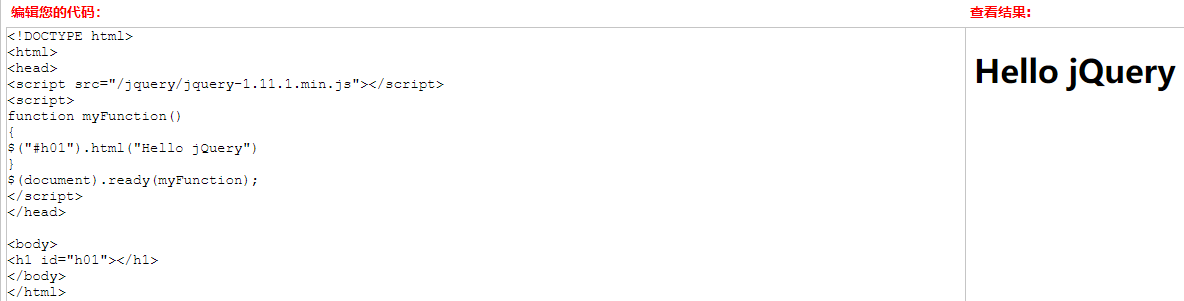
由于在 JavaScript 中函数就是变量，因此可以把 myFunction 作为变量传递给 jQuery 的 ready 方法。

**提示：**jQuery 返回 jQuery 对象，与已传递的 DOM 对象不同。jQuery 对象拥有的属性和方法，与 DOM 对象的不同。您不能在 jQuery 对象上使用 HTML DOM 的属性和方法。

### 3、测试JQuery

#### （1）例子1





#### （2）例子2