JavaScript进阶篇-笔记

目录

[第1章 系好安全带,准备启航 1](#_Toc401504065)

[1-1让你认识JS 1](#_Toc401504066)

[任务 1](#_Toc401504067)

[1-2编程练习 2](#_Toc401504068)

[任务 3](#_Toc401504069)

[第2章 你要懂的规则(JS基础语法) 4](#_Toc401504070)

[2-1什么是变量 4](#_Toc401504071)

[任务 4](#_Toc401504072)

[2-2 给变量取个名字(变量命名) 4](#_Toc401504073)

[任务 6](#_Toc401504074)

[2-3 确定你的存在(变量声明) 7](#_Toc401504075)

[任务 7](#_Toc401504076)

[2-4多样化的我(变量赋值） 8](#_Toc401504077)

[任务 8](#_Toc401504078)

[2-5表达出你的想法(表达式) 9](#_Toc401504079)

[任务 11](#_Toc401504080)

[2-6我还有其它用途( +号操作符） 12](#_Toc401504081)

[任务 12](#_Toc401504082)

[2-7自加一，自减一 ( ++和- -) 13](#_Toc401504083)

[任务 13](#_Toc401504084)

[2-8较量较量(比较操作符) 14](#_Toc401504085)

[任务 15](#_Toc401504086)

[2-9我与你同在(逻辑与操作符） 15](#_Toc401504087)

[任务 16](#_Toc401504088)

[2-10我或你都可以 (逻辑或操作符） 17](#_Toc401504089)

[任务 18](#_Toc401504090)

[2-11是非颠倒(逻辑非操作符） 18](#_Toc401504091)

[任务 19](#_Toc401504092)

[2-12保持先后顺序(操作符优先级） 19](#_Toc401504093)

[任务 20](#_Toc401504094)

[2-13编程练习 21](#_Toc401504095)

[任务 21](#_Toc401504096)

[第3章 一起组团(数组) 21](#_Toc401504097)

[3-1一起组团（什么是数组） 21](#_Toc401504098)

[任务 22](#_Toc401504099)

[3-2组团，并给团取个名（如何创建数组） 23](#_Toc401504100)

[任务 24](#_Toc401504101)

[3-3谁是团里成员（数组赋值） 24](#_Toc401504102)

[任务 25](#_Toc401504103)

[3-4团里添加新成员（向数组增加一个新元素） 26](#_Toc401504104)

[任务 27](#_Toc401504105)

[3-5呼叫团里成员(使用数组元素) 27](#_Toc401504106)

[任务 28](#_Toc401504107)

[3-6了解成员数量(数组属性length) 29](#_Toc401504108)

[任务 29](#_Toc401504109)

[3-7二维数组 30](#_Toc401504110)

[任务 31](#_Toc401504111)

[3-8编程练习 32](#_Toc401504112)

[任务 32](#_Toc401504113)

[第4章 跟着我的节奏走(流程控制语句) 33](#_Toc401504114)

[4-1做判断（if语句） 33](#_Toc401504115)

[任务 34](#_Toc401504116)

[4-2二选一 （if...else语句） 34](#_Toc401504117)

[任务 35](#_Toc401504118)

[4-3多重判断（if..else嵌套语句） 36](#_Toc401504119)

[任务 37](#_Toc401504120)

[4-4多种选择（Switch语句) 38](#_Toc401504121)

[任务 41](#_Toc401504122)

[4-5重复重复（for循环） 42](#_Toc401504123)

[任务 43](#_Toc401504124)

[4-6反反复复(while循环) 44](#_Toc401504125)

[任务 44](#_Toc401504126)

[4-7来来回回(Do...while循环) 45](#_Toc401504127)

[任务 47](#_Toc401504128)

[4-8退出循环break 48](#_Toc401504129)

[任务 49](#_Toc401504130)

[4-9继续循环continue 50](#_Toc401504131)

[任务 51](#_Toc401504132)

[4-10编程练习 52](#_Toc401504133)

[任务 52](#_Toc401504134)

[第5章 小程序，大作用（函数） 53](#_Toc401504135)

[5-1什么是函数 53](#_Toc401504136)

[任务 54](#_Toc401504137)

[5-2定义函数 54](#_Toc401504138)

[任务 55](#_Toc401504139)

[5-3函数调用 56](#_Toc401504140)

[任务 57](#_Toc401504141)

[5-4有参数的函数 57](#_Toc401504142)

[任务 58](#_Toc401504143)

[5-5返回值的函数 59](#_Toc401504144)

[任务 59](#_Toc401504145)

[编程练习 60](#_Toc401504146)

[任务 60](#_Toc401504147)

[第6章 事件响应，让网页交互 63](#_Toc401504148)

[6-1什么是事件 63](#_Toc401504149)

[任务 63](#_Toc401504150)

[6-2鼠标单击事件( onclick ） 63](#_Toc401504151)

[任务 64](#_Toc401504152)

[6-3鼠标经过事件（onmouseover） 65](#_Toc401504153)

[任务 66](#_Toc401504154)

[6-4鼠标移开事件（onmouseout） 67](#_Toc401504155)

[任务 68](#_Toc401504156)

[6-5光标聚焦事件（onfocus） 69](#_Toc401504157)

[任务 69](#_Toc401504158)

[6-6失焦事件（onblur） 70](#_Toc401504159)

[任务 71](#_Toc401504160)

[6-7内容选中事件（onselect） 72](#_Toc401504161)

[任务 73](#_Toc401504162)

[6-8文本框内容改变事件（onchange） 74](#_Toc401504163)

[任务 75](#_Toc401504164)

[6-9加载事件（onload） 75](#_Toc401504165)

[任务 76](#_Toc401504166)

[6-10卸载事件（onunload） 77](#_Toc401504167)

[任务 78](#_Toc401504168)

[6-11编程练习 78](#_Toc401504169)

[任务 78](#_Toc401504170)

[第7章 JavaScript内置对象 79](#_Toc401504171)

[7-1什么是对象 79](#_Toc401504172)

[任务 80](#_Toc401504173)

[7-2 Date 日期对象 80](#_Toc401504174)

[任务 81](#_Toc401504175)

[7-3返回/设置年份方法 81](#_Toc401504176)

[任务 82](#_Toc401504177)

[7-4返回星期方法 83](#_Toc401504178)

[任务 83](#_Toc401504179)

[7-5返回/设置时间方法 84](#_Toc401504180)

[任务 84](#_Toc401504181)

[7-6 String 字符串对象 85](#_Toc401504182)

[任务 86](#_Toc401504183)

[7-7返回指定位置的字符 87](#_Toc401504184)

[任务 87](#_Toc401504185)

[7-8返回指定的字符串首次出现的位置 88](#_Toc401504186)

[任务 89](#_Toc401504187)

[7-9字符串分割split() 90](#_Toc401504188)

[任务 91](#_Toc401504189)

[7-10提取字符串substring() 92](#_Toc401504190)

[任务 93](#_Toc401504191)

[7-11提取指定数目的字符substr() 93](#_Toc401504192)

[任务 94](#_Toc401504193)

[7-12 Math对象 95](#_Toc401504194)

[7-13向上取整ceil() 96](#_Toc401504195)

[任务 97](#_Toc401504196)

[任务 98](#_Toc401504197)

[7-14向下取整floor() 99](#_Toc401504198)

[任务 100](#_Toc401504199)

[7-15四舍五入round() 100](#_Toc401504200)

[任务 101](#_Toc401504201)

[7-16随机数 random() 102](#_Toc401504202)

[任务 103](#_Toc401504203)

[7-17Array 数组对象 103](#_Toc401504204)

[任务 105](#_Toc401504205)

[7-18数组连接concat() 105](#_Toc401504206)

[任务 107](#_Toc401504207)

[使用concat()完成数组连接 107](#_Toc401504208)

[7-19指定分隔符连接数组元素join() 107](#_Toc401504209)

[任务 108](#_Toc401504210)

[7-20颠倒数组元素顺序reverse() 109](#_Toc401504211)

[任务 109](#_Toc401504212)

[使用reverse()颠倒数组顺序 110](#_Toc401504213)

[7-21选定元素slice() 110](#_Toc401504214)

[任务 111](#_Toc401504215)

[使用slice()选定数组元素 111](#_Toc401504216)

[7-22数组排序sort() 112](#_Toc401504217)

[任务 113](#_Toc401504218)

[7-23编程练习 114](#_Toc401504219)

[任务 114](#_Toc401504220)

[第8章 浏览器对象 115](#_Toc401504221)

[8-1 window对象 115](#_Toc401504222)

[任务 116](#_Toc401504223)

[8-2 JavaScript 计时器 117](#_Toc401504224)

[任务 118](#_Toc401504225)

[8-3计时器setInterval() 118](#_Toc401504226)

[任务 119](#_Toc401504227)

[8-4取消计时器clearInterval() 120](#_Toc401504228)

[任务 121](#_Toc401504229)

[8-5计时器setTimeout() 121](#_Toc401504230)

[任务 123](#_Toc401504231)

[8-6取消计时器clearTimeout() 124](#_Toc401504232)

[任务 125](#_Toc401504233)

[8-7 History 对象 126](#_Toc401504234)

[任务 127](#_Toc401504235)

[8-8返回前一个浏览的页面 127](#_Toc401504236)

[任务 128](#_Toc401504237)

[8-9返回下一个浏览的页面 129](#_Toc401504238)

[任务 129](#_Toc401504239)

[8-10返回浏览历史中的其他页面 130](#_Toc401504240)

[任务 130](#_Toc401504241)

[8-11 Location对象 131](#_Toc401504242)

[任务 132](#_Toc401504243)

[8-12 Navigator对象 133](#_Toc401504244)

[任务 133](#_Toc401504245)

[8-13 userAgent 134](#_Toc401504246)

[任务 135](#_Toc401504247)

[8-14 screen对象 136](#_Toc401504248)

[任务 136](#_Toc401504249)

[8-15屏幕分辨率的高和宽 136](#_Toc401504250)

[任务 137](#_Toc401504251)

[8-16屏幕可用高和宽度 138](#_Toc401504252)

[任务 138](#_Toc401504253)

[8-17编程练习 139](#_Toc401504254)

[任务 140](#_Toc401504255)

[第9章 DOM对象，控制HTML元素 141](#_Toc401504256)

[9-1认识DOM 141](#_Toc401504257)

[任务 144](#_Toc401504258)

[9-2 getElementsByName()方法 145](#_Toc401504259)

[任务 146](#_Toc401504260)

[9-3 getElementsByTagName()方法 147](#_Toc401504261)

[任务 148](#_Toc401504262)

[9-4区别getElementByID,getElementsByName,getElementsByTagName以人来举例说明，人有能标识身份的身份证，有姓名，有类别(大人、小孩、老人)等。 150](#_Toc401504263)

[任务 152](#_Toc401504264)

[9-5 getAttribute()方法 154](#_Toc401504265)

[任务 155](#_Toc401504266)

[9-6 setAttribute()方法 157](#_Toc401504267)

[任务 158](#_Toc401504268)

[9-7节点属性 159](#_Toc401504269)

[任务 160](#_Toc401504270)

[9-8访问子结点childNodes 161](#_Toc401504271)

[任务 162](#_Toc401504272)

[9-9访问子结点的第一和最后项 164](#_Toc401504273)

[任务 164](#_Toc401504274)

[9-10访问父节点parentNode 165](#_Toc401504275)

[任务 166](#_Toc401504276)

[9-11访问兄弟节点 168](#_Toc401504277)

[任务 170](#_Toc401504278)

[9-12创建元素节点createElement 172](#_Toc401504279)

[任务 173](#_Toc401504280)

[9-13创建文本节点createTextNode 174](#_Toc401504281)

[任务 175](#_Toc401504282)

[9-14插入节点appendChild() 177](#_Toc401504283)

[任务 177](#_Toc401504284)

[9-15插入节点insertBefore() 178](#_Toc401504285)

[任务 180](#_Toc401504286)

[9-16删除节点removeChild() 181](#_Toc401504287)

[任务 182](#_Toc401504288)

[9-17替换元素节点replaceChild() 183](#_Toc401504289)

[任务 184](#_Toc401504290)

[9-18浏览器窗口可视区域大小 185](#_Toc401504291)

[任务 186](#_Toc401504292)

[9-19网页尺寸scrollHeight 187](#_Toc401504293)

[任务 187](#_Toc401504294)

[9-20网页尺寸offsetHeight 188](#_Toc401504295)

[任务 188](#_Toc401504296)

[9-21网页卷去的距离 189](#_Toc401504297)

[任务 190](#_Toc401504298)

[9-22编程练习 191](#_Toc401504299)

[任务 191](#_Toc401504300)

[10-1编程挑战 193](#_Toc401504301)

[任务 195](#_Toc401504302)

[要点： 197](#_Toc401504303)

[JS 中“===”与 “==”比较的区别 197](#_Toc401504304)

[++a与a++的区别 198](#_Toc401504305)

[JS中substr和substring的用法和区别 198](#_Toc401504306)

## ****第1章 系好安全带,准备启航****

### 1-1让你认识JS

你知道吗，Web前端开发师需要掌握什么技术?也许你已经了解HTML标记(也称为结构)，知道了CSS样式(也称为表示)，会使用HTML+CSS创建一个漂亮的页面，但这还不够，它只是静态页面而已。我们还需使用JavaScript增加行为，为网页添加动态效果。准备好，让JavaScript带你进入新境界吧!

**JavaScript能做什么？**

1.增强页面动态效果(如:下拉菜单、图片轮播、信息滚动等)

2.实现页面与用户之间的实时、动态交互(如:用户注册、登陆验证等)

**JS进阶篇学习什么？**

在**JavaScript入门篇**中，我们学习了如何插入JS、输出内容及简单的DOM操作，**JavaScript进阶篇**让您进一步的了解JS的变量、数组、函数、语法、对象、事件、DOM操作，制作简单的网页动态效果。

任务

我们先来回顾下JS入门篇的基础内容，在网页中插入JS代码，实现输出"JS进阶篇",并弹出对话框，内容为"关注JS高级篇"。

**注意:**

1. JS是区分大小写的，如：classname和ClassName是不一样的。同时注意方法、属性、变量等的大小写吆。

2. JS中的字符、符号等一定要在英文状态下输入吆。

**?**不会了怎么办

如果有问题，大家可以先学习下JS入门篇课程，**http://www.imooc.com/learn/36**。

<script type="text/javascript">

document.write("JS进阶篇");

alert("关注JS高级篇");

</script>

代码：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>我们互动下</title>

<script type="text/javascript">

document.write("JS进阶篇");

alert("关注JS高级篇");

</script>

</head>

<body>

</body>

</html>

### 1-2编程练习

小伙伴们，实现在html页面中的插入JavaScript脚本，打开页面时，在页面中显示一句话 : "系好安全带,准备启航--目标JS"，并弹出一个提示框："准备好了，起航吧！"

**温馨提示**: 完成任务后，请验证是否与实践要求一致，如一致，恭喜您，已经掌握此技能。否则，请重新学习课程内容吆，直到实践编写代码与实践要求一致。

任务

第一步:把注释语句注释。

第二步:编写代码，在页面中显示 “系好安全带,准备启航--目标JS”文字；

第三步:编写代码，在页面中弹出提示框“准备好了，起航吧！”

提示: 可以把弹框方法写在函数里。

第四步:使用引入JS外部文件的方式实现以上的任务。

**?**不会了怎么办

**注意:**本章是回顾" JS入门篇 " 内容，如有问题，请学习" JS入门篇 " ,**网址:** http://www.imooc.com/view/36

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>系好安全带,准备启航</title>

<!--引入外部文件的方式-->

<script type="text/javascript">

//多行注释

/\*我是多行注释！

我需要隐藏，

否则会报错哦！

\*/

//在页面中显示文字

document.write("系好安全带,准备启航--目标JS"+"<br>")

//页面中弹出提示框

function duihua(){

var txt=confirm("请点击我，准备好了吗？");

if(txt==true)

{

document.write("准备好了，起航吧！！");

}

else{document.write("请重新学习课程内容吆!");}

}

//单行注释

//我是单行注释，我也要隐藏起来！

</script>

</head>

<body>

<input type="button" value="请点击我，准备好了吗" onclick="duihua()">

</input>

</body>

</html>

## ****第2章 你要懂的规则(JS基础语法)****

### 2-1什么是变量

什么是变量? 从字面上看，变量是可变的量；从编程角度讲，变量是用于存储某种/某些数值的存储器。我们可以把变量看做一个盒子,盒子用来存放物品,物品可以是衣服、玩具、水果...等。

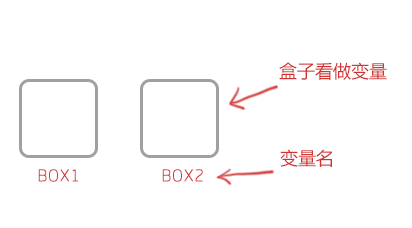
[](http://img.mukewang.com/529860210001304d04700234.jpg)

### 任务

此节没有任务，点击提交进行下节学习吧。

### [2-2 给变量取个名字(变量命名)](http://www.imooc.com/code/117)

我们为了区分盒子，可以用BOX1,BOX2等名称代表不同盒子，BOX1就是盒子的名字（也就是变量的名字）。

[](http://img.mukewang.com/529c05d40001140c04120228.jpg)

我们赶快给变量取个好名字吧!**变量名字可以任意取，**只不过取名字要遵循一些规则:

1.必须以字母、下划线或美元符号开头，后面可以跟字母、下划线、美元符号和数字。如下:

正确:

mysum

\_mychar

$numa1

错误:

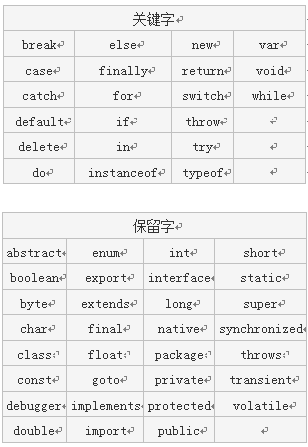
6num //开头不能用数字

%sum //开头不能用除(\_ $)外特殊符号,如(% + /等)

sum+num //开头中间不能使用除(\_ $)外特殊符号，如(% + /等)

2.变量名区分大小写，如:A与a是两个不同变量。

3.不允许使用JavaScript关键字和保留字做变量名。

[](http://img.mukewang.com/529c07c000014f5103080447.jpg)

### 任务

在右边编辑器的第7行补充代码,在var 后，填写变量名，如"myname"。

**?**不会了怎么办

在**半角**及**英文**状态下输入变量名

变量名以字母、下划线和$美元符号开头

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>变量命名</title>

<script type="text/JavaScript">

var mychar() ;

</script>

</head>

<body>

</body>

</html>

### [2-3 确定你的存在(变量声明)](http://www.imooc.com/code/120)

我们要使用盒子装东西,是不是先要找到盒子,那在编程中，这个过程叫声明变量,找盒子的动作，如何表示：

**声明变量语法:** **var 变量名;**

var就相当于找盒子的动作，在JavaScript中是关键字（即保留字），这个关键字的作用是声明变量，并为"变量"准备位置(即内存）。

var mynum ; //声明一个变量mynum

当然，我们可以一次找一个盒子，也可以一次找多个盒子，所以Var还可以一次声明多个变量，变量之前用","逗号隔开。

var num1,mun2 ; //声明一个变量num1

**注意:变量也可以不声明，直接使用，但为了规范，需要先声明，后使用。**

### 任务

在右边编辑器的第7行声明了一个变量mya，代码程序有问题，我们来修改下:

1使用正确声明语法，将vra修改为var

2.将变量名3mya修改为mya

**?**不会了怎么办

变量声明使用var

变量名以字母或者下划线或$开头

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>声明变量</title>

<script type="text/javascript">

var mya;

mya=5;

alert("mya的值是:"+mya);

</script>

</head>

<body>

</body>

</html>

### 2-4多样化的我(变量赋值）

我们可以把变量看做一个盒子,盒子用来存放物品,那如何在变量中存储内容呢?

我们使用"="号给变量存储内容,看下面的语句:

var mynum = 5 ; //声明变量mynum并赋值。

这个语句怎么读呢？ 给变量mynum赋值，值为5。我们也可以这样写:

var mynum; //声明变量mynum

mynum = 5 ; //给变量mynum赋值

**注:这里**"="**号的作用是给变量赋值，不是等于号。**

盒子可以装衣服、玩具、水果...等。其实，变量是无所不能的容器，你可以把任何东西存储在变量里，如数值、字符串、布尔值等，例如：

var num1 = 123; // 123是数值

var num2 = "一二三"; //"一二三"是字符串

var num3=true; //布尔值true（真），false(假)

其中，num1变量存储的内容是数值；num2变量存储的内容是字符串，字符串需要用一对引号""括起来，num3变量存储的内容是布尔值(true、false)。

### 任务

定义两个变量，并赋值，同时在网页中显示出变量的值。

1.在右边编辑器中第7行，定义num1变量，并赋值数值10。

2.在右边编辑器中第8行，定义num2变量，并赋值内容为字符串"JS"。

**?**不会了怎么办

定义变量var

变量赋值使用=号

赋值内容为字符串，一定要记住字符串需要用一对引号""括起来

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>变量赋值</title>

<script language="javascript">

var num1=10;

var num2="JS";

document.write("num1的值是:"+num1+"<br>");

document.write("num2的值是:"+num2);

</script>

</head>

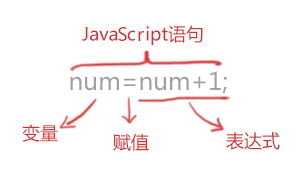
<body>

</html>

### 2-5表达出你的想法(表达式)

表达式与数学中的定义相似，表达式是指具有一定的值、用操作符把常数和变量连接起来的代数式。**一个表达式可以包含常数或变量。**

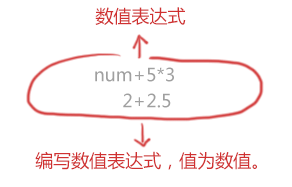
我们先看看下面的JavaScript语句：

[](http://img.mukewang.com/52b9227d0001acc703000174.jpg)

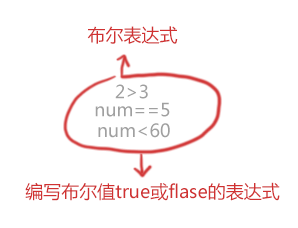
生活中“再见”表达方法很多，如:英语(goodbye）、网络语（88）、肢体语（挥挥手）等。在JavaScript表达式无处不在，所以一定要知道可以表达哪些内容，看看下面几种情况:

[](http://img.mukewang.com/529c21600001d02302800228.jpg)

注意:串表达式中mychar是变量

[](http://img.mukewang.com/52b922a00001445b02920187.jpg)

注意:数值表达式中num是变量

[](http://img.mukewang.com/529c218f000101d402980228.jpg)

注意:布尔表达式中num是变量

### 任务

小明有10元钱，买了一个本，花了5元，小红说:"你剩下的钱加上我的6元钱，就可以买个铅笔盒了。

1.在右边编辑器的第7行,定义变量num1,并计算小明剩下多少钱。

2.在右边编辑器的第8行，定义变量num2,并计算小红花多少钱买铅笔盒。

**?**不会了怎么办

表达式中可以是变量、字符、数值等。

小明剩多少钱：var num1=10-5;

小红花多少钱：var num2=num1+6;

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>表达式</title>

<script type="text/javascript">

var num1=10-5;

var num2=num1+6;

document.write("小明还剩:"+num1+"元"+"<br>");

document.write("小红花:"+num2+"元买个铅笔盒");

</script>

</head>

<body>

</body>

</html>

### 2-6我还有其它用途( +号操作符）

操作符是用于在JavaScript中指定一定动作的符号。

（1）操作符

看下面这段JavaScript代码。

sum = numa + numb;

其中的"="和"+"都是操作符。

JavaScript中还有很多这样的操作符，例如，算术操作符(+、-、\*、/等)，比较操作符(<、>、>=、<=等)，逻辑操作符(&&、||、！)。

**注意: “=” 操作符是赋值，不是等于。**

(2) "+"操作符

算术运算符主要用来完成类似加减乘除的工作，在JavaScript中，“+”不只代表加法，还可以连接两个字符串，例如：

mystring = "Java" + "Script"; // mystring的值“JavaScript”这个字符串

### 任务

使用"+"号连连看，看在网页中能输出什么内容吧!

1. 第7行输入1+5;

2. 第8行输入"中"+"国";

**?**不会了怎么办

+号两边是数值，功能为计算，值为数值。

+号两边一侧为字符串，功能为连接作用，值为字符串。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>+操作符</title>

<script language="javascript">

var mynum=1+5;

var mychar="中"+"国";

document.write("mynum的值:"+mynum+"<br>");

document.write("mychar的值:"+mychar);

</script>

</head>

<body>

</body>

</html>

### 2-7自加一，自减一 ( ++和- -)

算术操作符除了(+、-、\*、/)外，还有两个非常常用的操作符，自加一“++”；自减一“--”。首先来看一个例子：

mynum = 10;

mynum++; //mynum的值变为11

mynum--; //mynum的值又变回10

上面的例子中，mynum++使mynum值在原基础上增加1，mynum--使mynum在原基础上减去1,其实也可以写成:

mynum = mynum + 1;//等同于mynum++

mynum = mynum - 1;//等同于mynum--

### 任务

试一试，让变量numa自加一，让变量numb自减一

1. 在右边编辑器中第9行输入numa++;

2. 在右边编辑器中第10行输入numb--;

**?**不会了怎么办

numa++ : 变量numa值加1后，再赋值给numa

numb-- : 变量numb值减1后，再赋值给numb

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>++/--</title>

<script language="javascript">

var numa=1;

var numb=7;

numa++;

numb--;

document.write("numa的值:"+numa+"<br>");

document.write("numb的值:"+numb);

</script>

</head>

<body>

</body>

</html>

### 2-8较量较量(比较操作符)

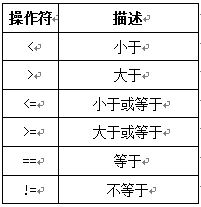
我们先来做道数学题，数学考试成绩中，小明考了90分，小红考了95分，问谁考的分数高?

答: 因为“95 > 90”，所以小红考试成绩高。

其中大于号">" 就是比较操作符，小红考试成绩和小明考试成绩就是操作数，并且是两个操作数。

也就是说两个操作数通过比较操作符进行比较，得到值为真（true）和假(false)。

在JavaScript中，这样的比较操作符有很多，这些操作符的含义如下:

[](http://img.mukewang.com/532a3c150001c65802010207.jpg)  
看看下面例子:

var a = 5;//定义a变量，赋值为5

var b = 9; //定义b变量，赋值为9

document.write (a<b); //a小于b的值吗? 结果是真(true)

document.write (a>=b); //a大于或等于b的值吗? 结果是假(false)

document.write (a!=b); //a不等于b的值吗? 结果是真(true)

document.write (a==b); //a等于b的值吗? 结果是假(false)

### 任务

实验考试成绩中，numa考了1分，numb考了7分，我们来比较下它们的分数。

1. 在右边编辑器中第10行输入numa>numb

2. 在右边编辑器中第11行输入numa!=numb

**?**不会了怎么办

在半角及英文状态下输入>、!=等比较操作符

numa、numb要小写，因为JS中区分大小写

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>比较操作符</title>

<script language="javascript">

var numa,numb,jq1,jq2;

numa=1;

numb=7;

jq1=numa>numb;

jq2=numa!=numb;

document.write("numa大于numb的分数吗？"+jq1+"<br>")

document.write("numa不等于numb的分数吗？"+ jq2);

</script>

</head>

<body>

</body>

</html>

### 2-9我与你同在(逻辑与操作符）

数学里面的“a>b”，在JavaScript中还表示为a>b；数学中的“b大于a，b小于c”是“a<b<c”，那么在JavaScript中可以用&&表示，如下：

b>a && b<c //“&&”是并且的意思, 读法"b大于a"并且" b小于c "

好比我们参加高考时,在进入考场前,必须出示准考证和身份证,两者缺一不可，否则不能参加考试，表示如下:

if(有准考证 &&有身份证)

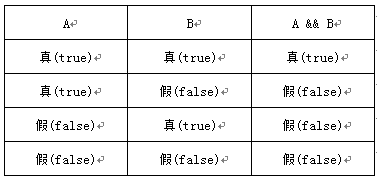
{

进行考场考试

}

 “&&”是逻辑与操作符，只有“&&”两边值同时满足(同时为真)，整个表达式值才为真。

**逻辑与操作符值表:**

**[](http://img.mukewang.com/52a16f880001d27803770180.jpg)**

### 任务

JavaScript考试成绩中，numa（表示笔试）考了60分，numb（表示实践）考了70分，只有两项成绩都大于或等于60分，该课程考核就通过。

1. 在右边编辑器中第10行,if语句的括号内输入numa>=60 && numb>=60

**?**不会了怎么办

使用&&完成与操作

numa、numb都是小写的，因为JS中区分大小写

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>逻辑与</title>

<script type="text/javascript">

var numa,numb;

numa=60;

numb=70;

if(numa>=60&&numb>=60)

{

document.write("JavaScript考核通过！"+"<br>");

}

</script>

</head>

<body>

</body>

</html>

### 2-10我或你都可以 (逻辑或操作符）

"||"逻辑或操作符，相当于生活中的“或者”，当两个条件中有任一个条件满足，“逻辑或”的运算结果就为“真”。

例如：本周我们计划出游,可是周一至周五工作,所以周六或者周日哪天去都可以。即两天中只要有一天有空，就可以出游了。

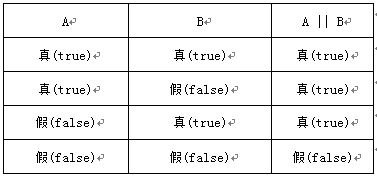
var a=3;

var b=5;

var c;

c=b>a ||a>b; //b>a是true，a>b是false，c是true

**逻辑或操作符值表:**

[](http://img.mukewang.com/530a9d2b0001a33e03770178.jpg)

### 任务

在右边编辑器中第10行补充代码，在“jq1=”后输入numa>numb || numa==numb，看看结果是什么。

**?**不会了怎么办

numa、numb都是小写的，因为JS中区分大小写

使用||完成或操作

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>逻辑或</title>

<script type="text/javascript">

var numa,numb,jq1;

numa=50;

numb=55;

jq1= numa>numb||numa==numb ;

document.write("jq1的值是:"+jq1+"<br>")

</script>

</head>

<body>

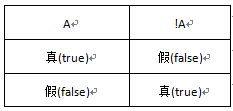
</body>

</html>

### 2-11是非颠倒(逻辑非操作符）

"!"是逻辑非操作符，也就是"不是"的意思,非真即假，非假即真。好比小华今天买了一个杯子，小明说:"杯子是白色的"，小亮说:“杯子是红色的”，小华说："小明说的不是真话，小亮说的不是假话"。猜猜小华买的什么颜色的杯子，答案：红色杯子。

**逻辑非操作符值表:**

**[](http://img.mukewang.com/52a1760c000159a702330111.jpg)**

看看下面代码，变量c的值是什么:

var a=3;

var b=5;

var c;

c=!(b>a); // b>a值是true,! (b>a）值是false

c=!(b<a); // b<a值是false, ! (b<a）值是true

### 任务

在右边编辑器中第10行补充代码，在“jq1= ”后输入!(numa<numb),看看结果是什么。

**?**不会了怎么办

numa、numb都是小写的，因为JS中区分大小写

使用!完成非操作

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>逻辑非</title>

<script language="javascript">

var numa,numb,jq1;

numa=60;

numb=70;

jq1= !(numa<numb) ;

document.write("jq1的值是:"+jq1+"<br>")

</script>

</head>

<body>

</body>

</html>

### 2-12保持先后顺序(操作符优先级）

我们都知道，除法、乘法等操作符的优先级比加法和减法高，例如：

var numa=3;

var numb=6

jq= numa + 30 / 2 - numb \* 3; // 结果为0

如果我们要改变运算顺序，需添加括号的方法来改变优先级:

var numa=3;

var numb=6

jq= ((numa + 30) / (2 - numb)) \* 3; //结果是-24.75

**操作符之间的优先级（高到低）:**

算术操作符 → 比较操作符 → 逻辑操作符 → "="赋值符号

如果同级的运算是按从左到右次序进行,多层括号由里向外。

var numa=3;

var numb=6;

jq= numa + 30 >10 && numb \* 3<2; //结果为false

### 任务

考考大家，先计算右边代码编辑器中第10行和第11行语句的结果,然后运行代码,看看结果是否一样。

**?**不会了怎么办

操作符之间的优先级为:

算术操作符 → 比较操作符 → 逻辑操作符 → "="赋值符号

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>操作符优先级</title>

<script language="javascript">

var numa,numb,jq1;

numa=5;

numb=2;

jq1=numa + 30 >10 && numb \* 3<20;

jq2=((numa + 30) / (7-numb)) \* 3

document.write("jq1的值是:"+jq1+"<br>");

document.write("jq2的值是:"+jq2);

</script>

</head>

<body>

</body>

</html>

### 2-13编程练习

 考考大家的数学，计算以下计算公式的结果。然后在浏览器中运行一下，看看结果是否跟你的结果一致。

### 任务

第一步： 在  ？ 处填写你的答案。

第二步： 填写完成后，运行一下，看看是不是跟你填写的结果一致。

第三步： 如果你有不一致的地方，思考一下为什么跟你的不一致呢。

**?**不会了怎么办

代码：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>JS基础</title>

<script type="text/javascript">

var a,b,sum;

var a = 5;

var b = 100%7;

sum = a > b && a\*b > 0 ;

document.write( "我认为 a 的值是:" + 5 + " b的值是:" + 2 + "sum 的值是:" +true+"<br/>");

document.write( "答案是,第一轮计算后，a 为："+ a +";b为："+b +";第一次计算sum为："+ sum +"<br/>");

sum = ( (++a) + 3 ) / (2 - (--b) ) \* 3;

document.write( "再一次计算后，我认为 a 的值是:" + 6 + " b的值是:" + 1+ "sum 的值是:" + 27 +"<br/>");

document.write( "答案是，第二轮计算后，a 为：" + a + ";b为：" + b +";第一次计算sum为："+ sum +",sum的类型也发生变化了。");

</script>

</head>

<body>

</body>

</html>

## ****第3章 一起组团(数组)****

### 3-1一起组团（什么是数组）

我们知道变量用来存储数据，一个变量只能存储一个内容。假设你想存储10个人的姓名或者存储20个人的数学成绩，就需要10个或20个变量来存储，如果需要存储更多数据，那就会变的更麻烦。我们用数组解决问题，一个数组变量可以存放多个数据。好比一个团，团里有很多人，如下我们使用数组存储5个学生成绩。

[](http://img.mukewang.com/52c9ff5c0001085a05460266.jpg)

数组是一个值的集合，每个值都有一个索引号，从0开始，每个索引都有一个相应的值，根据需要添加更多数值。

### 任务

编辑器中定义数组myarr,并赋值3个学生的成绩， 验证结果窗口中是否是80，60，99。

**?**不会了怎么办

大家不要着急，接下来就进入数组的操作。

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>什么是数组</title>

<script type="text/javascript">

var myarr=new Array(); //定义数组

myarr[0]=80;

myarr[1]=60;

myarr[2]=99;

document.write("第一个人的成绩是:"+myarr[0]+";<br>");

document.write("第二个人的成绩是:"+myarr[1]+";<br>");

document.write("第三个人的成绩是:"+myarr[2]+";<br>");

</script>

</head>

<body>

</html>

### 3-2组团，并给团取个名（如何创建数组）

使用数组之前首先要创建，而且需要把数组本身赋至一个变量。好比我们出游，要组团，并给团定个名字“云南之旅”。

**创建数组语法：**

var myarray=new Array();

[](http://img.mukewang.com/52ca004b0001c81103980228.jpg)          
 我们创建数组的同时，还可以为数组指定长度，长度可任意指定。

var myarray= new Array(8); //创建数组，存储8个数据。

**注意：**  
1.创建的新数组是空数组，没有值，如输出，则显示undefined。  
2.虽然创建数组时，指定了长度，但实际上数组都是变长的，也就是说即使指定了长度为8，仍然可以将元素存储在规定长度以外。

### 任务

1.在右边编辑器的第7行，使用new Array()创建一个新数组，并保存在myarr变量中  
2.验证编辑器的第8行输出数组第一个值，在结果窗口中是否是undefined。

**?**不会了怎么办

var myarr=new Array();

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>创建数组</title>

<script type="text/JavaScript">

var myarr=new Array();

document.write("数组的第一个值："+myarr[0]);

</script>

</head>

<body>

</body>

</html>

### 3-3谁是团里成员（数组赋值）

数组创建好，接下来我们为数组赋值。我们把数组看似旅游团的大巴车，大巴车里有很多位置，每个位置都有一个号码，顾客要做在哪个位置呢？

第一步：组个大巴车

第二步：按票对号入座

大巴车的1号座位是张三

大巴车的2号座位是李四

**数组的表达方式：**

第一步：创建数组var myarr=new Array();

第二步：给数组赋值

myarr[1]=" 张三";

myarr[2]=" 李四";

下面创建一个数组，用于存储5个人的数学成绩。

var myarray=new Array(); //创建一个新的空数组

myarray[0]=66; //存储第1个人的成绩

myarray[1]=80; //存储第2个人的成绩

myarray[2]=90; //存储第3个人的成绩

myarray[3]=77; //存储第4个人的成绩

myarray[4]=59; //存储第5个人的成绩

**注意：**数组每个值有一个索引号，从0开始。

我们还可以用简单的方法创建上面的数组和赋值：

第一种方法：

var myarray = new Array(66,80,90,77,59);//创建数组同时赋值

第二种方法：

var myarray = [66,80,90,77,59];//直接输入一个数组（称 “字面量数组”）

**注意：**数组存储的数据可以是任何类型（数字、字符、布尔值等）

### 任务

1.在右边编辑器中第7行创建数组，长度为3，保存在myarr变量中。  
2.在右边编辑器中第8、9、10行为数组中每项赋值，分别为88，90，68。

**?**不会了怎么办

**第一种方法：**

第7行创建数组：var myarr=new Array(3);

第8行myarr[0]=88;

第9行myarr[1]=90;

第10行myarr[2]=68;

**第二种方法：**

var myarr=new Array(88，90，68);

**第三种方法：**

var myarr=[88，90，68];

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>数组赋值</title>

<script language="javascript">

var myarr=new Array(3);

myarr[0]=88

myarr[1]=90

myarr[2]=68

document.write("数组第一个值是:"+myarr[0]+"<br>");

document.write("数组第二个值是:"+myarr[1]+"<br>");

document.write("数组第三个值是:"+myarr[2]+"<br>");

</script>

</head>

<body>

</body>

</html>

### 3-4团里添加新成员（向数组增加一个新元素）

上一节中，我们使用myarray变量存储了5个人的成绩，现在多出一个人的成绩，如何存储呢？   
[](http://img.mukewang.com/52ca00eb0001dd4805120199.jpg)

只需使用下一个未用的索引，任何时刻可以不断向数组增加新元素。

myarray[5]=88; //使用一个新索引，为数组增加一个新元素

### 任务

数组中已有三个数值88，90，68，为数组新增加一个元素（第四个），值为99。

**?**不会了怎么办

myarray[3]=99;

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>增加新元素</title>

<script language="javascript">

var myarray=new Array();

myarray[0]=88;

myarray[1]=90;

myarray[2]=68;

myarray[3]=98;

document.write("数组第一个值是:"+myarray[0]+"<br>");

document.write("数组第二个值是:"+myarray[1]+"<br>");

document.write("数组第三个值是:"+myarray[2]+"<br>");

document.write("数组新增加的值是:"+myarray[3]+"<br>");

</script>

</head>

<body>

</body>

</html>

### 3-5呼叫团里成员(使用数组元素)

我们知道数组中的每个值有一个索引号，从0开始，如下图， myarray变量存储6个人的成绩：  
 [](http://img.mukewang.com/52ca012800016f3505460234.jpg)  
要得到一个数组元素的值，只需引用数组变量并提供一个索引，如：  
**第一个人的成绩表示方法：**myarray[0]  
**第三个人的成绩表示方法:** myarray[2]

### 任务

变量myarr保存4个人的姓名，现在看看第二个人的姓名是什么?  
1.在右边编辑器的第12行补充代码，显示第二个人的姓名。（?位置写代码）

**?**不会了怎么办

 myarr[1]里存储第二个人的姓名

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>使用数组</title>

<script language="javascript">

var myarr=new Array();

myarr[0]="小红";

myarr[1]="小明";

myarr[2]="小亮";

myarr[3]="小川";

document.write("第二人的姓名是:"+ myarr[1]);

</script>

</head>

<body>

</body>

</html>

### 3-6了解成员数量(数组属性length)

如果我们想知道数组的大小，只需引用数组的一个属性length。Length属性表示数组的长度，即数组中元素的个数。

**语法：**

myarray.length; //获得数组myarray的长度

**注意：**因为数组的索引总是由0开始，所以一个数组的上下限分别是：0和length-1。如数组的长度是5，数组的上下限分别是0和4。

var arr=[55,32,5,90,60,98,76,54];//包含8个数值的数组arr

document.write(arr.length); //显示数组长度8

document.write(arr[7]); //显示第8个元素的值54

同时，JavaScript数组的length属性是可变的，这一点需要特别注意。

arr.length=10; //增大数组的长度

document.write(arr.length); //数组长度已经变为10

数组随元素的增加，长度也会改变，如下:

var arr=[98,76,54,56,76]; // 包含5个数值的数组

document.write(arr.length); //显示数组的长度5

arr[15]=34; //增加元素，使用索引为15,赋值为34

alert(arr.length); //显示数组的长度16

### 任务

在右边编辑器中，已创建数组mynum，并赋值65，90，88，98。  
1. 在右边编辑器的第8行，使用length属性显示数组长度。

**?**不会了怎么办

使用length属性。

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>数组长度</title>

<script language="javascript">

var mynum=new Array(65,90,88,98);

document.write("数组的长度是:"+ mynum.length );

</script>

</head>

<body>

</body>

</html>

### 3-7二维数组

一级数组，我们看成一组盒子，每个盒子只能放一个内容。

一维数组的表示: myarray[ ]

二维数组，我们看成一组盒子，不过每个盒子里还可以放多个盒子。

二维数组的表示: myarray[ ][ ]

**注意:**二维数组的两个维度的索引值也是从0开始，两个维度的最后一个索引值为长度-1。

**1. 二维数组的定义方法一**

var myarr=new Array(); //先声明一维

for(var i=0;i<2;i++){ //一维长度为2

myarr[i]=new Array(); //在声明二维

for(var j=0;j<3;j++){ //二维长度为3

myarr[i][j]=i+j; // 赋值，每个数组元素的值为i+j

}

}

**注意:**关于for 循环语句，请看第四章4-5 。

将上面二维数组，用表格的方式表示:

[](http://img.mukewang.com/537957a20001c24c03200210.jpg)

2. **二维数组的定义方法二**

var Myarr = [[0 , 1 , 2 ],[1 , 2 , 3, ]]

**3. 赋值**

myarr[0][1]=5; //将5的值传入到数组中，覆盖原有值。

**说明:** myarr[0][1] ,0 表示表的行，1表示表的列。

### 任务

试一试，创建二维数组(一维长度3，二维长度5)，值为一维数组和二维数组索引值的积，如myarr[2][5]=2\*5。

在右边编辑器的script标签内，创建该二维数组。

**?**不会了怎么办

1. var Myarr = [[0 , 0 , 0 , 0 ,0 ],[0 , 1 ,2 , 3 ,4 ],[0 , 2 , 4 , 6 ,8 ]]

2.

var myarr=new Array(); //先声明一维

for(var i=0;i<3;i++){ //一维长度为2

myarr[i]=new Array(); //在声明二维

for(var j=0;j<5;j++){ //二维长度为3

myarr[i][j]=i\*j; // 赋值，每个数组元素的值为i\*j

}

}

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>二维数组</title>

<script type="text/javascript">

var myarr=new Array();

for(var i=0;i<3;i++){

myarr[i]=new Array();

for(var j=0;j<5;j++){

myarr[i][j]=i\*j;

}

}

</script>

</head>

<body>

</body>

</html>

### 3-8编程练习

使用Javascript语言，把以下数组

   var  arr = ['\*','##',"\*\*\*","&&","\*\*\*\*","##\*"];

   arr[7] = "\*\*";

在页面显示如下图所示的图案：

\*

\*\*

\*\*\*

\*\*\*\*

### 任务

第一步：定义一个数组，存储要展示的图形。

提示:

var arr = ['\*','##',"\*\*\*","&&","\*\*\*\*","##\*"];

arr[7] = "\*\*";

第二步：预想一下arr 数组的长度是多少？

提示: 使用alert弹出数组的长度，看看是否跟自己的答案一样。

第三步：将数组内容输出，完成达到的效果。

提示: 想在不同行显示不同数组内容，别忘记<br/>换行标签。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>数组</title>

<script type="text/javascript">

//创建数组

var arr = ['\*','##',"\*\*\*","&&","\*\*\*\*","##\*"];

arr[7] = "\*\*";

//显示数组长度

alert("数组长度为："+arr.length)

//将数组内容输出，完成达到的效果。

document.write(arr[0]+"<br/>"+arr[7]+"<br/>"+arr[2]+"<br/>"+arr[4]);

</script>

</head>

<body>

</body>

</html>

## ****第4章 跟着我的节奏走(流程控制语句)****

### 4-1做判断（if语句）

if语句是基于条件成立才执行相应代码时使用的语句。

**语法:**

if(条件)

{ 条件成立时执行代码}

**注意：if小写，大写字母（IF）会出错！**

假设你应聘web前端技术开发岗位，如果你会HTML技术，你面试成功，欢迎加入公司。代码表示如下:

<script type="text/javascript">

var mycarrer = "HTML";

if (mycarrer == "HTML")

{

document.write("你面试成功，欢迎加入公司。");

}

</script>

### 任务

请在右边编辑器的第7行，使用if语句补充代码，实现判断，是否购买笔，条件:只要笔是黑色(black)，就购买。

**?**不会了怎么办

1.使用if语句判断

2.条件为mypen == "black"

**代码：**

<!DOCTYPE HTML>

<html><head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>if语句</title>

<script type="text/JavaScript">

var mypen = "black"; //定义变量mypen，存储笔的黑色。

if(mypen=="black")

{

document.write("笔是黑色的，所以购买");

}

</script>

</head>

<body>

</body>

</html>

### 4-2二选一 （if...else语句）

if...else语句是在指定的条件成立时执行代码，在条件不成立时执行else后的代码。

**语法:**

if(条件)

{ 条件成立时执行的代码}

else

{条件不成立时执行的代码}

假设你应聘web前端技术开发岗位，如果你会HTML技术，你面试成功，欢迎加入公司，否则你面试不成功，不能加入公司。

**代码表示如下:**

<script type="text/javascript">

var mycarrer = "HTML"; //mycarrer变量存储技能

if (mycarrer == "HTML")

{ document.write("你面试成功，欢迎加入公司。"); }

else //否则，技能不是HTML

{ document.write("你面试不成功，不能加入公司。");}

</script>

### 任务

今天去看电影，如果有票就可以看电影，没有票电影看不成。在右边编辑器的第8、12行补充代码，使用if...else进行判断。

**?**不会了怎么办

if语句()里填写判断条件

else执行条件不成立语句

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>if...else</title>

<script type="text/JavaScript">

function myfunction(){

var ticket=document.getElementById("thedo").value;

if(ticket==true)

{

document.getElementById("show\_results").innerHTML="有票，可以看电影。";

}else{

document.getElementById("show\_results").innerHTML="没票，电影看不成。";

}

}

</script>

</head>

<body>

<select id="thedo" name="">

<option value="0">没有票</option>

<option value="1">有票</option>

</select>

<input type="submit" onclick="myfunction()" value="ok">

<p id="show\_results"></p>

</body>

</html>

### 4-3多重判断（if..else嵌套语句）

要在多组语句中选择一组来执行，使用if..else嵌套语句。

**语法:**

**if(**条件1**)**

{ 条件1成立时执行的代码}

**else if(**条件2**)**

{ 条件2成立时执行的代码}

...

**else if(**条件n**)**

{ 条件n成立时执行的代码}

**else**

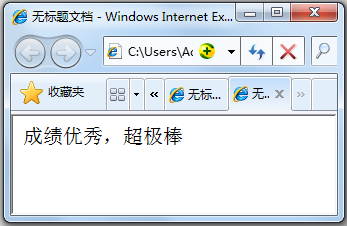
{ 条件1、2至n不成立时执行的代码}

假设数学考试，小明考了86分，给他做个评价，60分以下的不及格，60(包含60分)-75分为良好，75(包含75分)-85分为很好，85(包含75分)-100优秀。

**代码表示如下:**

[](http://img.mukewang.com/541799000001aada05260280.jpg)

**结果:**

[](http://img.mukewang.com/52d121a70001de7503470226.jpg)

### 任务

联合国世界卫生组织对年龄的划分标准， 44岁以下为青年；45岁至59·岁为中年人。60岁至89岁为老年人；90岁以上为长寿老年人。赵红今年99岁了，他属于那个年龄阶段的。使用if...else嵌套语句进行判断，补充右边代码编辑器中第12、14行代码，完成判断。

**?**不会了怎么办

多条件判断else if

最后判断使用else

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>if...else</title>

<script type="text/JavaScript">

var myage =99;//赵红的年龄为99

if(myage<=44)

{document.write("青年");}

else if(myage<=59)

{document.write("中年人");}

else if (myage<=89)

{document.write("老年人");}

else

{document.write("长寿老年人");}

</script>

</head>

<body>

</body>

</html>

### 4-4多种选择（Switch语句)

当有很多种选项的时候，switch比if else使用更方便。

**语法:**

**switch(表达式)**

{

**case值1:**

执行代码块 1

**break;**

**case值2:**

执行代码块 2

**break;**

**...**

**case值n:**

执行代码块 n

**break;**

**default:**

与 case值1 、 case值2...case值n 不同时执行的代码

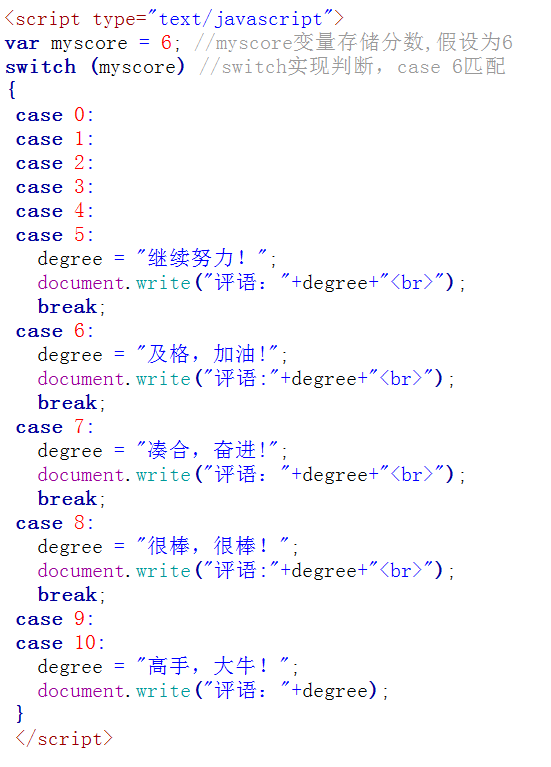
}

**语法说明:**

Switch必须赋初始值，值与每个case值匹配。满足执行该 case 后的所有语句，并用break语句来阻止运行下一个case。

假设评价学生的考试成绩，10分满分制，我们按照每一分一个等级将成绩分等，并根据成绩的等级做出不同的评价。

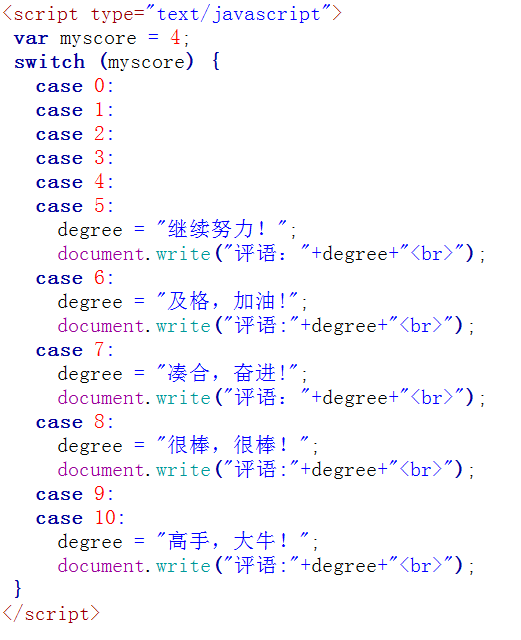
**代码如下:**

[](http://img.mukewang.com/52d1293c0001394705510767.jpg)

**执行结果:**

**评语: 及格，加油!**

**注意:记得在case所执行的语句后添加上一个break语句。否则就直接继续执行下面的case中的语句，看以下代码:**

[](http://img.mukewang.com/52d128b80001995105210640.jpg)

**执行结果:**

评语: 继续努力!

评语: 及格，加油!

评语: 凑合，奋进

评语: 很棒，很棒

评语: 高手，大牛

在上面的代码中，没有break停止语句，如果成绩是4分，则case 5后面的语句将会得到执行，同样，case6、7-10后面的语句都会得到执行。

### 任务

我们来做个周计划，周一、二学习理念知识，周三、四到企业实践，周五总结经验，周六、日休息和娱乐。请根据星期几做出选择，使用Switch语句完成，补充右边代码编辑器中第8、20、21行代码，完成星期几的判断。

**?**不会了怎么办

使用switch进行选择

case进行条件匹配

break 结束当前 case中的语句

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>switch</title>

<script type="text/JavaScript">

var myweek = 4;//myweek表示星期几变量

switch(myweek)

{

case 1:

case 2:

document.write("学习理念知识");

break;

case 3:

case 4:

document.write("到企业实践");

break;

case 5:

document.write("总结经验");

break;

default:

document.write("周六、日休息和娱乐");

}

</script>

</head>

<body>

</body>

</html>

### 4-5重复重复（for循环）

很多事情不只是做一次，要重复做。如打印10份试卷，每次打印一份，重复这个动作，直到打印完成。这些事情，我们使用循环语句来完成，循环语句，就是重复执行一段代码。

**for语句结构：**

**for**(初始化变量;循环条件;循环迭代)

{

  循环语句

 }

假如，一个盒子里有6个球，我们每次取一个，重复从盒中取出球，直到球取完为止。

<script type="text/javascript">

var num=1;

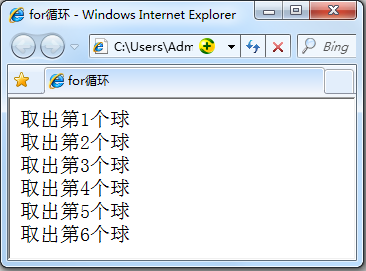
for (num=1;num<=6;num++)  //初始化值；循环条件；循环后条件值更新

{  document.write("取出第"+num+"个球<br />");

}

</script>

**结果:**

**[](http://img.mukewang.com/52d649f700019de303660271.jpg)**

**执行思路:**

[](http://img.mukewang.com/5371cf4200019cc406640707.jpg)

### 任务

我们有1，2，3...10不同面值的钱，使用for语句完成合计，看看我们一共有多少钱，请补充第8行代码，实现该功能。

**?**不会了怎么办

for语句，缺少"循环条件"，mymeny<=10

for(mymeny=1;mymeny<=10;mymeny++)

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>if...else</title>

<script type="text/JavaScript">

var mymeny,sum=0;//mymeny变量存放不同面值，sum总计

for(mymeny=1;mymeny<=10;mymeny++)

{

sum= sum + mymeny;

}

document.write("sum合计:"+sum);

</script>

</head>

<body>

</body>

</html>

### 4-6反反复复(while循环)

和for循环有相同功能的还有while循环, while循环重复执行一段代码，直到某个条件不再满足。

**while语句结构：**

**while(判断条件)**

{

循环语句

}

使用while循环，完成从盒子里取球的动作，每次取一个，共6个球。

<script type="text/javascript">

var num=0; //初始化值

while (num<=6) //条件判断

{

document.write("取出第"+num+"个球<br />");

num=num+1; //条件值更新

}

</script>

### 任务

我们使用while语句，输出5个数字。补充第8行代码。

**?**不会了怎么办

while(mynum<=5)或while(mynum<6)

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>while</title>

<script type="text/javascript">

var mynum =1;//mynum初值化数值为1

while (mynum<=5)

{

document.write("数字:"+mynum+"<br/>");

mynum=mynum+1;

}

</script>

</head>

<body>

</body>

</html>

### 4-7来来回回(Do...while循环)

do while结构的基本原理和while结构是基本相同的，但是它保证循环体至少被执行一次。因为它是先执行代码，后判断条件。

**do...while语句结构：**

**do**

{

    循环语句

}

**while**(判断条件)

我们试着输出5个数字。

<script type="text/javascript">

   num= 1;

   do

   {

     document.write("数值为:" +  num+"<br />");

     num++; //更新条件

   }

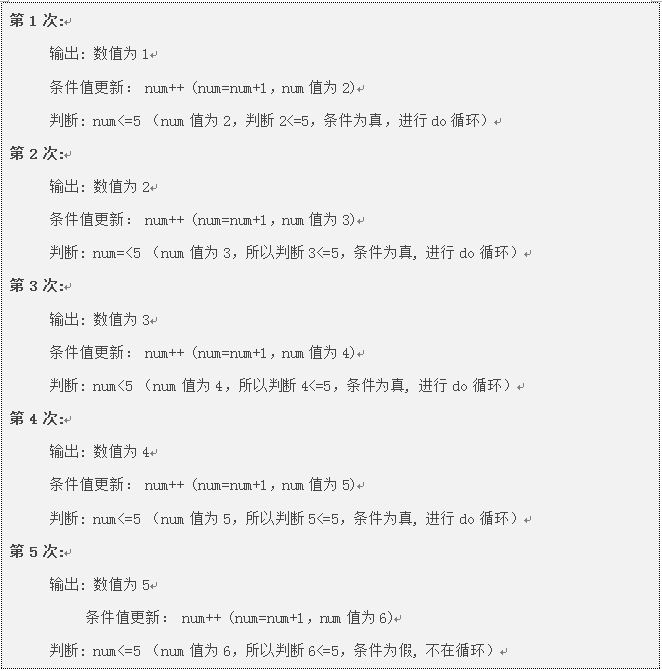
   while (num<=5)

</script>

**执行结果:**

[](http://img.mukewang.com/52dc805c0001da3703680274.jpg)

为什么呢?我们来看下执行思路：

[](http://img.mukewang.com/52dc808b0001bff006620671.jpg)

### 任务

我们使用do...while语句，输出6个数字。补充第8行和13行代码。

**?**不会了怎么办

do...while语句，因为至少被执行一次，所以要注意循环条件。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>do...while</title>

<script type="text/javascript">

var mynum =6;//mynum初值化数值为6

do

{

document.write("数字:"+mynum+"<br/>");

mynum=mynum-1;

}

while(mynum<=6 && mynum>=1)

</script>

</head>

<body>

</body>

</html>

### 4-8退出循环break

在while、for、do...while、while循环中使用break语句退出当前循环，直接执行后面的代码。

**格式如下：**

**for**(初始条件;判断条件;循环后条件值更新)

{

**if**(特殊情况)

{**break;**}

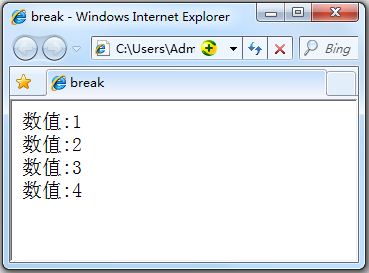
循环代码

}

当遇到特殊情况的时候，循环就会立即结束。看看下面的例子，输出10个数，如果数值为5，就停止输出。

[](http://img.mukewang.com/52dc866d0001512004990273.jpg)

**执行结果:**

[](http://img.mukewang.com/52dc85920001b91503690273.jpg)

**注:**当num=5的时候循环就会结束，不会输出后面循环的内容。

### 任务

考试成绩输出，如果成绩及格继续输出下个成绩，如果成绩不及格，退出并且后面成绩不输出， 我们使用break语句，退出循环。补充第14行代码。

**?**不会了怎么办

break语句，退出循环。

**代码：**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>break</title>

<script type="text/JavaScript">

var mynum =new Array(70,80,66,90,50,100,89);//定义数组mynum并赋值

var i=0;

while(i<mynum.length)

{

if(mynum[i]<60)

{

document.write("成绩"+mynum[i]+"不及格，不用循环了"+"<br>");

break;

}

document.write("成绩:"+mynum[i]+"及格，继续循环"+"<br>");

i=i+1;

}

</script>

</head>

<body>

</body>

</html>

### 4-9继续循环continue

continue的作用是仅仅跳过本次循环，而整个循环体继续执行。

**语句结构：**

**for**(初始条件;判断条件;循环后条件值更新)

{

if(特殊情况)

{ **continue;** }

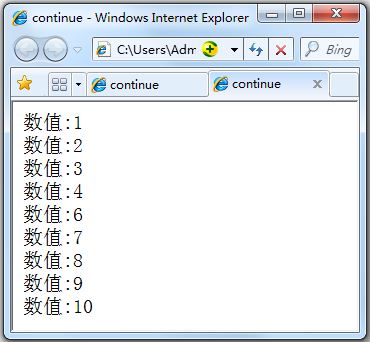
循环代码

}

上面的循环中，当特殊情况发生的时候，本次循环将被跳过，而后续的循环则不会受到影响。好比输出10个数字，如果数字为5就不输出了。

[](http://img.mukewang.com/52dc89270001212905110297.jpg)

**执行结果:**

**[](http://img.mukewang.com/52dc88e800017b8d03700342.jpg)**

**注:**上面的代码中，num=5的那次循环将被跳过。

### 任务

考试成绩输出，如果成绩及格继续输出下个成绩，如果成绩不及格，则不输出该成绩，我们使用continue语句完成,补充第14行代码。

**?**不会了怎么办

continue语句，退出本次循环，继续执行后面的循环。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>continue</title>

<script type="text/JavaScript">

var mynum =new Array(70,80,66,90,50,100,89);//定义数组mynum并赋值

var i;

for(i=0;i<mynum.length;i++)

{

if(mynum[i]<60)

{

document.write("成绩不及格，不输出！"+"<br>");

continue;

}

document.write("成绩:"+mynum[i]+"及格，输出!"+"<br>");

}

</script>

</head>

<body>

</body>

</html>

### 4-10编程练习

在一个大学的编程选修课班里，我们得到了一组参加该班级的学生数据，分别是姓名、性别、年龄和年级，接下来呢，我们要利用JavaScript的知识挑出其中所有是大一的女生的的名字哦。

**学生信息如下：**

    ('小A','女',21,'大一'),  ('小B','男',23,'大三'),

    ('小C','男',24,'大四'),  ('小D','女',21,'大一'),

    ('小E','女',22,'大四'),  ('小F','男',21,'大一'),

    ('小G','女',22,'大二'),  ('小H','女',20,'大三'),

    ('小I','女',20,'大一'),  ('小J','男',20,'大三')

### 任务

第一步： 把这些数据组成一个数组，方便我们之后操作哦。

提示: 使用二维数组。

第二步： 筛选数据吧，首先找出都是大一的所有信息 ;

第三步： 最后再一次筛选上一步得到的数据，打印出都是女生的姓名 ;

提示: 可以用switch 或 if 语句进行筛选。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title></title>

<script type="text/javascript">

var infos = [

['小A','女',21,'大一'],

['小B','男',23,'大三'],

['小C','男',24,'大四'],

['小C','男',24,'大四'],

['小D','女',21,'大一'],

['小E','女',22,'大四'],

['小F','男',21,'大一'],

['小G','女',22,'大二'],

['小H','女',20,'大三'],

['小I','女',20,'大一'],

['小J','男',20,'大三']

];

for(var i = 0; i < infos.length; i++) {

if(infos[i][3] === "大一" && infos[i][1] === "女") {

document.write(infos[i][0] + "<br />");

}

}

</script>

</head>

<body>

</body>

</html>

## ****第5章 小程序，大作用（函数）****

### 5-1什么是函数

函数的作用，可以写一次代码，然后反复地重用这个代码。

如:我们要完成多组数和的功能。

var sum;

sum = 3+2;

alert(sum);

sum=7+8 ;

alert(sum);

.... //不停重复两行代码

如果要实现8组数的和，就需要16行代码，实现的越多，代码行也就越多。所以我们可以把完成特定功能的代码块放到一个函数里，直接调用这个函数，就省去重复输入大量代码的麻烦。

**使用函数完成:**

function add2(a,b){

sum = a + b;

alert(sum);

} // 只需写一次就可以

add2(3,2);

add2(7,8);

.... //只需调用函数就可以

### 任务

此节没的任务，快快进入下节函数的学习吧!

### 5-2定义函数

如何定义一个函数呢？看看下面的格式：

**function** **函数名( )**

{

函数体;

}

function定义函数的关键字，“函数名”你为函数取的名字，“函数体”替换为完成特定功能的代码。

我们完成对两个数求和并显示结果的功能。并给函数起个有意义的名字：“add2”，代码如下：

<script type="text/javascript">

function add2(){

sum = 3 + 2;

alert(sum);

}

​add2();

</script>

**结果:**

**[](http://img.mukewang.com/5338c81600019e9101600189.jpg)**

### 任务

补充右边编辑器第7行，编写一个函数，实现二个数的差，函数名为sub2。

**?**不会了怎么办

1.定义函数使用function

2.function 后面是函数名

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>定义函数</title>

<script type="text/javascript">

function sub2() //定义函数

{

sub=5-2;

alert("5和2的差:"+sub);

}

</script>

</head>

<body>

<form>

<input type="button" value="点击我" onclick="sub2()" />

</form>

</body>

</html>

### 5-3函数调用

函数定义好后，是不能自动执行的，需要调用它,直接在需要的位置写函数名。

**第一种情况:在<script>标签内调用。**

<script type="text/javascript">

function add2()

  {

sum = 1 + 1;

alert(sum);

}

**add2();**//调用函数，直接写函数名。

</SCRIPT>

**第二种情况:**在HTML文件中调用，如通过点击按钮后调用定义好的函数。

<html>

<head>

<script type="text/javascript">

function add2()

  {

sum = 5 + 6;

alert(sum);

}

</script>

</head>

<body>

<form>

<input type="button" value="click it" onclick="**add2()**"> //按钮,onclick点击事件，直接写函数名

</form>

</body>

</html>

注意:鼠标事件会在后面讲解。

### 任务

**补充右边编辑器第15行，实现如下功能:**

网页中有一按钮(名字"点点我")，当点击按钮后调用函数tcon(),弹出对话框"恭喜你学会函数调用了!"。

**?**不会了怎么办

函数调用，在需要的位置直接写函数名。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>函数调用</title>

<script type="text/javascript">

function tcon()

{

alert("恭喜你学会函数调用了!");

}

</script>

</head>

<body>

<form>

<input type="button" value="点点我" onclick="tcon()">

</form>

</body>

</html>

### 5-4有参数的函数

上节中add2()函数不能实现任意指定两数相加。其实，定义函数还可以如下格式：

**function 函数名**(参数1,参数2)

{

函数代码

}

**注意:参数可以多个，根据需要增减参数个数。参数之间用(逗号，）隔开。**

按照这个格式，函数实现任意两个数的和应该写成：

function add2(x,y)

{

sum = x + y;

document.write(sum);

}

x和y则是函数的两个参数，调用函数的时候，我们可通过这两个参数把两个实际的加数传递给函数了。

例如，add2(3，4)会求3+4的和，add2(60,20)则会求出60和20的和。

### 任务

**补充右边编辑器代码，实现如下功能:**

1. 补充右边编辑器第7行代码，定义函数实现三个数的和，函数名为add3。

2. 补充右边编辑器第12、13行代码，计算5、8、3/7、1、4两组三个数的和。

**?**不会了怎么办

1.定义函数要有参数，这样就可以重复使用并传不同的值。

 add3(x,y,z)

2.求不同数的和，在调用函数要记的传数值。

 add3(5,8,3);

 add3(7,1,4);

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>函数传参</title>

<script type="text/JavaScript">

function add3(x,y,z)

{

sum = x + y +z;

document.write(x+"、"+y+"、"+z+"和:"+sum+"<br/>");

}

add3(5,8,3);

add3(7,1,4);

</script>

</head>

<body>

</body>

</html>

### 5-5返回值的函数

**思考:**上一节函数中，通过"document.write"把结果输出来，如果想对函数的结果进行处理怎么办呢？

我们只要把"document.write(sum)"这行改成如下代码：

function add2(x,y)

{

sum = x + y;

**return** sum; //返回函数值,return后面的值叫做返回值。

}

还可以通过变量存储调用函数的返回值，代码如下:

result = add2(3,4);//语句执行后,result变量中的值为7。

**注意:函数中参数和返回值不只是数字，还可以是字符串等其它类型。**

### 任务

**补充右边编辑器代码，实现如下功能:**

1. 定义函数实现两个数的积，函数名为app2,补充右边编辑器第10行代码，返回函数值。

2. 补充右边编辑器第12、13行代码，计算5、6/2、3两组两个数的积。分别保存在req1和req2变量中。

3. 补充右边编辑器第14行代码，计算req1和req2和的值，保存在变量sumq中。

**?**不会了怎么办

1.函数返回值，使用return。

return sum;

2. 返回值可以存储在变量中。

var req1=app2(5,6);

var req2=app2(2,3);

var sumq=req1+req2;

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>返回值函数</title>

<script type="text/javascript">

function app2(x,y)

{ var sum,x,y;

sum = x \* y;

return sum;

}

req1=app2(5,6);

req2=app2(2,3);

sumq=req1+req2;

document.write("req1的值:"+req1+"<br/>");

document.write("req2的值:"+req2+"<br/>");

document.write("req1的值:"+req1+"与"+"req2的值:"+req2+"的和:"+sumq);

</script>

</head>

<body>

</body>

</html>

### 编程练习

使用javascript代码写出一个函数：实现传入两个整数后弹出较大的整数。

### 任务

第一步： 编写代码完成一个函数的定义吧。

第二步： 我们来补充函数体中的控制语句，完成函数功能吧。

提示:再想一想，两个整数比较有三种情况，大于，等于或小于，所以我们需要控制语句判断(if...else if)。

第三步：  写好的函数，我们就可以任意调用函数了。通过函数调用实现两组数值中，返回较大值吧。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>函数</title>

<script type="text/javascript">

//定义函数

function size(x,y){

if(x>y){

return x;

}

else if(x<y){

return y;

}

else{

document.write("一样大")

}

}

//函数体，判断两个整数比较的三种情况

a=size(5,4);

b=size(6,3);

//调用函数，实现下面两组数中，返回就大值。

document.write(" 5 和 4 的较大值是:"+a+"<br>");

document.write(" 6 和 3 的较大值是:"+b );

</script>

</head>

<body>

</body>

</html>

## ****第6章 事件响应，让网页交互****

### [6-1](http://www.imooc.com/code/635)什么是事件

JavaScript 创建动态页面。事件是可以被 JavaScript 侦测到的行为。 网页中的每个元素都可以产生某些可以触发 JavaScript 函数或程序的事件。

比如说，当用户单击按钮或者提交表单数据时，就发生一个鼠标单击（onclick）事件，需要浏览器做出处理，返回给用户一个结果。

**主要事件表:**

[](http://img.mukewang.com/53e198540001b66404860353.jpg)

任务

此节没有任务，快快进入下节学习鼠标事件。

### 6-2鼠标单击事件( onclick ）

onclick是鼠标单击事件，当在网页上单击鼠标时，就会发生该事件。同时onclick事件调用的程序块就会被执行，通常与按钮一起使用。

比如，我们单击按钮时，触发 onclick 事件，并调用两个数和的函数add2()。代码如下：

<html>

<head>

  <script type="text/javascript">

  function add2(){

  var numa,numb,sum;

  numa=6;

  numb=8;

  sum=numa+numb;

  document.write("两数和为:"+sum); }

  </script>

</head>

<body>

<form>

<input name="button" type="button" value="点击提交" onclick="add2()" />

</form>

</body>

</html>

**注意:**在网页中，如使用事件，就在该元素中设置事件属性。

任务

补充右边编辑器第12行，当鼠标点击"确定"按钮后，调用openwin()函数。

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>单击事件 </title>

<script type="text/javascript">

function openwin(){

window.open('http://www.imooc.com','\_blank','height=600,width=400,top=100,toolbar=no,left=0,menubar=no,scrollbars=no,status=no');}

</script>

</head>

<body>

<form>

<input name="点击我" type="button" value="点击我"/>

</form>

</body>

</html>

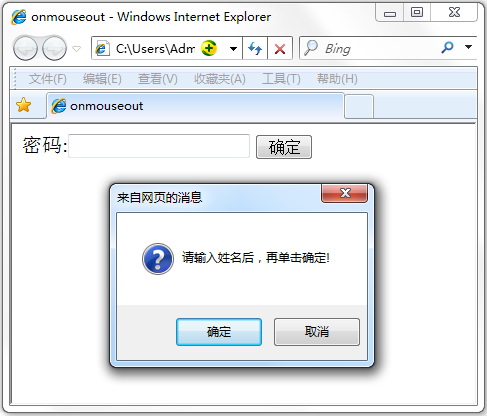
### 6-3鼠标经过事件（onmouseover）

鼠标经过事件，当鼠标移到一个对象上时，该对象就触发onmouseover事件，并执行onmouseover事件调用的程序。

现实鼠标经过"确定"按钮时，触发onmouseover事件，调用函数info()，弹出消息框，代码如下:

[](http://img.mukewang.com/53e196e500013f1807700354.jpg)

**运行结果:**



任务

补充右边编辑器第13行，当鼠标经过"确定"按钮后，调用message()函数，弹出消息对话框。

**?**不会了怎么办

使用onmouseover事件,为input标签添加属性 onmouseover="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title> 鼠标经过事件 </title>

<script type="text/javascript">

function message(){

confirm("请输入密码后，再单击确定!"); }

</script>

</head>

<body>

<form>

密码:<input name="password" type="password" >

<input name="确定" type="button" value="确定" onmouseover="message()" />

</form>

</body>

</html>

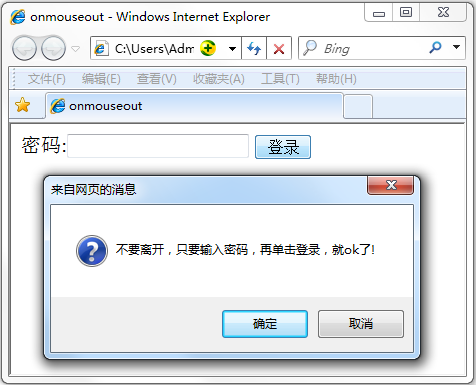
### 6-4鼠标移开事件（onmouseout）

鼠标移开事件，当鼠标移开当前对象时，执行onmouseout调用的程序。

当把鼠标移动到"登录"按钮上，然后再移开时，触发onmouseout事件，调用函数message()，代码如下:

[](http://img.mukewang.com/53e195580001a0bc07730356.jpg)

**运行结果:**

[](http://img.mukewang.com/53e195bf00010d1804760385.jpg)

任务

补充右边编辑器第12行，当鼠标移开"点击我"的按钮时，调用message()函数，弹出消息对话框。

**?**不会了怎么办

使用onmouseout事件,为a标签添加属性onmouseout="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>鼠标移开事件 </title>

<script type="text/javascript">

function message(){

alert("不要移开，点击后进行慕课网!"); }

</script>

</head>

<body>

<form>

<a href="http://www.imooc.com" onmouseout="message()">点击我</a>

</form>

</body>

</html>

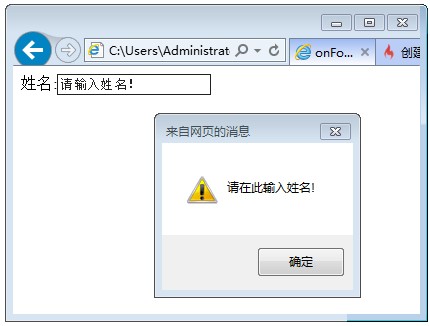
### 6-5光标聚焦事件（onfocus）

当网页中的对象获得聚点时，执行onfocus调用的程序就会被执行。

如下代码, 当将光标移到文本框内时，即焦点在文本框内，触发onfocus 事件，并调用函数message()。

[](http://img.mukewang.com/53e19337000113d108390338.jpg)

**运行结果：**

[](http://img.mukewang.com/5312c5660001821a04300326.jpg)

任务

补充右边编辑器第14行，当下拉列表得到焦点时，调用message()函数，就弹出对话框“"请选择，您现在的职业！”。

**?**不会了怎么办

1. 添加聚焦事件

2. onfocus ="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title> 光标聚焦事件 </title>

<script type="text/javascript">

function message(){

alert("请选择，您现在的职业！");

}

</script>

</head>

<body>

请选择您的职业：<br>

<form>

<select name="career" onfocus="message()">

<option>学生</option>

<option>教师</option>

<option>工程师</option>

<option>演员</option>

<option>会计</option>

</select>

</form>

</body>

</html>

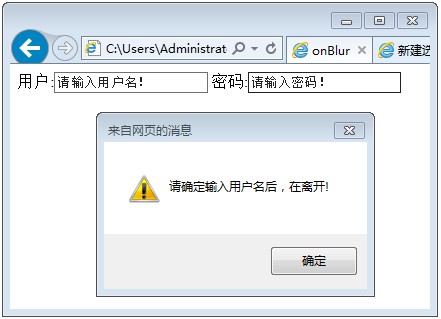
### 6-6失焦事件（onblur）

onblur事件与onfocus是相对事件，当光标离开当前获得聚焦对象的时候，触发onblur事件，同时执行被调用的程序。

如下代码, 网页中有用户和密码两个文本框。当前光标在用户文本框内时（即焦点在文本框），在光标离开该文本框后（即失焦时），触发onblur事件，并调用函数message()。

[](http://img.mukewang.com/53e191d00001dfe508560326.jpg)

**运行结果：**

**[](http://img.mukewang.com/5312da710001968704410319.jpg)**

任务

补充右边编辑器第13行，当光标从密码文本框离开时，触发onblur 事件，调用message()函数。

**?**不会了怎么办

1. 使使用onblur事件

2. onblur="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title> 失焦事件 </title>

<script type="text/javascript">

function message(){

alert("请确定已输入密码后，在移开!"); }

</script>

</head>

<body>

<form>

用户:<input name="username" type="text" value="请输入用户名！" >

密码:<input name="password" type="text" value="请输入密码！" onblur="message()" >

</form>

</body>

</html>

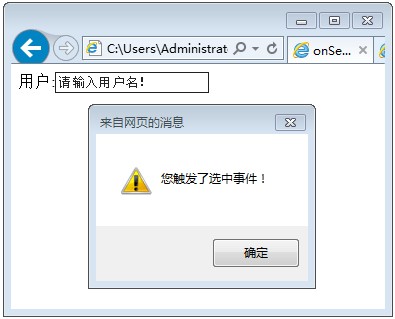
### 6-7内容选中事件（onselect）

选中事件，当文本框或者文本域中的文字被选中时，触发onselect事件，同时调用的程序就会被执行。

如下代码,当选中用户文本框内的文字时，触发onselect 事件，并调用函数message()。

[](http://img.mukewang.com/53e190b70001ffa908560325.jpg)

**运行结果：**

**[](http://img.mukewang.com/5312d7ba00013fff03950319.jpg)**

任务

补充右边编辑器第13行，当选中个人简介文本框中文字时，触发onselect事件，并弹出对话框。

**?**不会了怎么办

1. 使用onselect事件

2. onselect="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title> 内容选中事件 </title>

<script type="text/javascript">

function message(){

alert("您触发了选中事件！"); }

</script>

</head>

<body>

<form>

个人简介：<br>

<textarea name="summary" cols="60" rows="5" onselect="message()">请写入个人简介，不少于200字！</textarea>

</form>

</body>

</html>

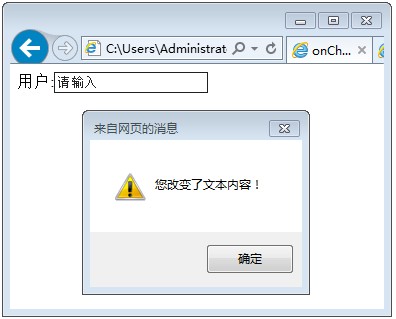
### 6-8文本框内容改变事件（onchange）

通过改变文本框的内容来触发onchange事件，同时执行被调用的程序。

如下代码,当用户将文本框内的文字改变后，弹出对话框“您改变了文本内容！”。

[](http://img.mukewang.com/5312d59c00011cd311490444.jpg)

**运行结果：**

**[](http://img.mukewang.com/5312d5c600012c3703960319.jpg)**

任务

补充右边编辑器第13行，实现改变文本框内容时，调用message()函数，弹出对话框“您改变了文本内容！”。

**?**不会了怎么办

1. 使用onchange事件

2.  onchange="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title> 文本框内容改变事件 </title>

<script type="text/javascript">

function message(){

alert("您改变了文本内容！"); }

</script>

</head>

<body>

<form>

个人简介：<br>

<textarea name="summary" cols="60" rows="5" onchange="message()">请写入个人简介，不少于200字！</textarea>

</form>

</body>

</html>

### 6-9加载事件（onload）

事件会在页面加载完成后立即发生，同时执行被调用的程序。  
注意：1. 加载页面时，触发onload事件，事件写在<body>标签内。

      2. 此节的加载页面，可理解为打开一个新页面时。  
如下代码,当加载一个新页面时，弹出对话框“加载中，请稍等…”。

[](http://img.mukewang.com/5312e3c10001bd9908920372.jpg)

**运行结果：**

**[](http://img.mukewang.com/5312e3eb0001e8a103930318.jpg)**

任务

补充右边编辑器第10行，实现加载页面时，弹出对话框“加载中，请稍等…”。

**?**不会了怎么办

1. 使用 onload事件

2. onLoad ="message()"

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title> 加载事件 </title>

<script type="text/javascript">

function message(){

alert("加载中，请稍等…"); }

</script>

</head>

<body onload ="message()">

欢迎学习JavaScript。

</body>

</html>

### 6-10卸载事件（onunload）

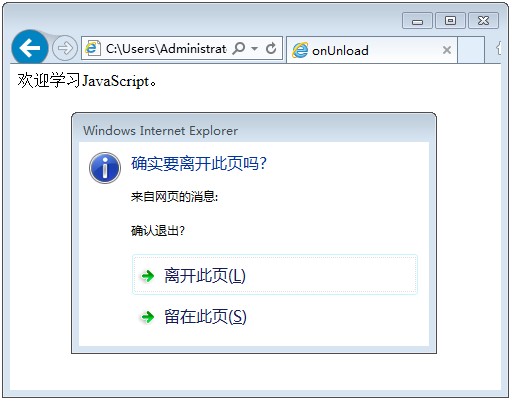
当用户退出页面时（页面关闭、页面刷新等），触发onUnload事件，同时执行被调用的程序。

**注意：不同浏览器对onunload事件支持不同。**

如下代码,当退出页面时，弹出对话框“您确定离开该网页吗？”。

[](http://img.mukewang.com/5312ee6b0001f89408950418.jpg)

**运行结果：（IE浏览器）**

**[](http://img.mukewang.com/5312ef5c0001a85d05120401.jpg)**

### 任务

由于浏览器不同及演示局限性，此节没有任务，大家可以拷贝代码，在本机上演示。

### 6-11编程练习

使用JS完成一个简单的计算器功能。实现2个输入框中输入整数后，点击第三个输入框能给出2个整数的加减乘除。

提示：获取元素的值设置和获取方法为：例：赋值：document.getElementById(“id”）.value = 1； 取值：var 啊= document.getElementById(“id”）.value；

任务

第一步： 创建构建运算函数count()。

第二步： 获取两个输入框中的值和获取选择框的值。

提示：document.getElementById( id名 ).value 获取或设置 id名的值。

第三步： 获取通过下拉框来选择的值来改变加减乘除的运算法则。

提示：使用switch判断运算法则。

第四步：  通过 = 按钮来调用创建的函数，得到结果。

**注意:** 使用parseInt()函数可解析一个字符串,并返回一个整数。

**?**不会了怎么办

## ****第7章 JavaScript内置对象****

### 7-1什么是对象

JavaScript 中的所有事物都是对象，如:字符串、数值、数组、函数等，每个对象带有**属性**和**方法**。

**对象的属性：**反映该对象某些特定的性质的，如：字符串的长度、图像的长宽等；

**对象的方法：**能够在对象上执行的动作。例如，表单的“提交”(Submit)，时间的“获取”(getYear)等；

JavaScript 提供多个内建对象，比如 String、Date、Array 等等，使用对象前先定义，如下使用数组对象：

var objectName =new Array();//使用new关键字定义对象

**或者**

var objectName =[];

**访问对象属性的语法:**

objectName.propertyName

如使用 Array 对象的 length 属性来获得数组的长度：

var myarray=new Array(6);//定义数组对象

var myl=myarray.length;//访问数组长度length属性

**以上代码执行后，myl的值将是：6**

**访问对象的方法：**

objectName.methodName()

如使用string 对象的 toUpperCase() 方法来将文本转换为大写：

var mystr="Hello world!";//创建一个字符串

var request=mystr.toUpperCase(); //使用字符串对象方法

以上代码执行后，request的值是**：HELLO WORLD!**

### 任务

补充右边编辑器第5行，计算数组myarray的长度,并保存在变量mynum中，使代码实现获取数组长度并输出结果。

**?**不会了怎么办

1.计算数组的长度，使用length属性。

2.var mynum=myarray.length;

### 7-2 Date 日期对象

日期对象可以储存任意一个日期，并且可以精确到毫秒数（1/1000 秒）。

定义一个时间对象 :

var Udate=new Date();

**注意:**使用关键字new，Date()的首字母必须大写。

使 Udate 成为日期对象，并且已有初始值：**当前时间(当前电脑系统时间)**。

如果要自定义初始值，可以用以下方法：

var d = new Date(2012, 10, 1); //2012年10月1日

var d = new Date('Oct 1, 2012'); //2012年10月1日

我们最好使用下面介绍的“方法”来严格定义时间。

**访问方法语法：**“<日期对象>.<方法>”

Date对象中处理时间和日期的常用方法：

[](http://img.mukewang.com/53d1c6ed000151aa04190327.jpg)

### 任务

**此节没有任务，快快进入下节，学习Date对象的方法。**

**?**不会了怎么办

**此节没有任务，快快进入下节，学习Date对象的方法。**

### 7-3返回/设置年份方法

get/setFullYear() 返回/设置年份，用四位数表示。

var mydate=new Date();//当前时间2014年3月6日

document.write(mydate+"<br>");//输出当前时间

document.write(mydate.getFullYear()+"<br>");//输出当前年份

mydate.setFullYear(81); //设置年份

document.write(mydate+"<br>"); //输出年份被设定为 0081年。

**注意:**不同浏览器， mydate.setFullYear(81)结果不同，年份被设定为 0081或81两种情况。

**结果:**

Thu Mar 06 2014 10:57:47 GMT+0800

2014

Thu Mar 06 0081 10:57:47 GMT+0800

**注意:**

1.结果格式依次为：星期、月、日、年、时、分、秒、时区。(火狐浏览器)

2. 不同浏览器，时间格式有差异。

### 任务

补充右边编辑器第8行，使用getFullYear(）获得年份，并输出当前年份。

**?**不会了怎么办

使用getFullYear()获得年份

var myyear= mydate.getFullYear();

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>获得年份 </title>

<script type="text/javascript">

var mydate=new Date();

var myyear=mydate.getFullYear();

document.write("年份:"+myyear);

</script>

</head>

<body>

</body>

</html>

### 7-4返回星期方法

**getDay()**返回星期，返回的是0-6的数字，0 表示星期天。如果要返回相对应“星期”，通过数组完成，代码如下:

<script type="text/javascript">

var mydate=new Date();//定义日期对象

var weekday=["星期日","星期一","星期二","星期三","星期四","星期五","星期六"];

//定义数组对象,给每个数组项赋值

var mynum=mydate.getDay();//返回值存储在变量mynum中

document.write(mydate.getDay());//输出getDay()获取值

document.write("今天是："+ weekday[mynum]);//输出星期几

</script>

注意：以上代码是在2014年3月7日，星期五运行。

**结果:**

5

今天是：星期五

### 任务

补充右边编辑器第9行，使用getDay(）获得当前星期，并输出。

**?**不会了怎么办

使用getDay()获得星期

**正确代码:**

document.write("今天是：" + weekday[mydate.getDay()];

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>获得星期</title>

<script type="text/javascript">

var mydate=new Date();

var weekday=["星期日","星期一","星期二","星期三","星期四","星期五","星期六"];

document.write("今天是：" +weekday[mydate.getDay()] );

</script>

</head>

<body>

</body>

</html>

### 7-5返回/设置时间方法

**get/setTime()** 返回/设置时间，单位毫秒数，计算从 1970 年 1 月 1 日零时到日期对象所指的日期的毫秒数。

如果将目前日期对象的时间推迟1小时，代码如下:

<script type="text/javascript">

var mydate=new Date();

document.write("当前时间："+mydate+"<br>");

mydate.setTime(mydate.getTime() + 60 \* 60 \* 1000);

document.write("推迟一小时时间：" + mydate);

</script>

**结果:**

当前时间：Thu Mar 6 11:46:27 UTC+0800 2014

推迟一小时时间：Thu Mar 6 12:46:27 UTC+0800 2014

**注意:**1. 一小时 60 分，一分 60 秒，一秒 1000 毫秒

      2. 时间推迟 1 小时,就是: “x.setTime(x.getTime() + 60 \* 60 \* 1000);”

### 任务

补充右边编辑器第9行，使用getTime()和setTime()方法，将时间推后2小时。

**?**不会了怎么办

推后两个小时: 2\* 60 \* 60 \* 1000

mydate.getTime() + 2\* 60 \* 60 \* 1000

**正确代码:**

mydate.setTime(mydate.getTime() + 2\* 60 \* 60 \* 1000);

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>日期对象 </title>

<script type="text/javascript">

var mydate=new Date();

document.write("当前时间："+mydate+"<br>");

mydate.setTime(mydate.getTime() + 2\* 60 \* 60 \* 1000);

document.write("推迟二小时时间：" + mydate);

</script>

</head>

<body>

</body>

</html>

### 7-6 String 字符串对象

在之前的学习中已经使用字符串对象了，定义字符串的方法就是直接赋值。比如：

var mystr = "I love JavaScript!"

定义mystr字符串后，我们就可以访问它的属性和方法。

**访问字符串对象的属性length:**

stringObject.length; 返回该字符串的长度。

var mystr="Hello World!";

var myl=mystr.length;

以上代码执行后，myl 的值将是：12

**访问字符串对象的方法：**

使用 String 对象的 toUpperCase() 方法来将字符串小写字母转换为大写：

var mystr="Hello world!";

var mynum=mystr.toUpperCase();

以上代码执行后，mynum 的值是：HELLO WORLD!

### 任务

1.补充右边编辑器第8行，使用toLowerCase()方法，将字符串所有大写字母都变成小写的字符串。

2.字符串还有很多常用方法，快快进入下节学习字符串其它方法。

**?**不会了怎么办

message.toLowerCase()

**正确代码:**

var mychar=message.toLowerCase();

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>string对象 </title>

<script type="text/javascript">

var message="I love JavaScript!";

var mychar=message.toLowerCase();

document.write("字符串为："+mychar+"<br>");

</script>

</head>

<body>

</body>

</html>

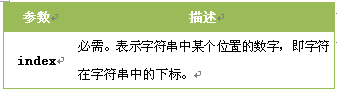
### 7-7返回指定位置的字符

charAt() 方法可返回指定位置的字符。返回的字符是长度为 1 的字符串。

**语法:**

stringObject.charAt(index)

**参数说明：**

[](http://img.mukewang.com/53251a310001cf1e03370092.jpg)  
**注意：**1.字符串中第一个字符的下标是 0。最后一个字符的下标为字符串长度减一（string.length-1）。

2.如果参数 index 不在 0 与 string.length-1 之间，该方法将返回一个空字符串。

**如:**在字符串 "I love JavaScript!" 中，返回位置2的字符：

<script type="text/javascript">

var mystr="I love JavaScript!"

document.write(mystr.charAt(2));

</script>

**注意：**一个空格也算一个字符。

以上代码的运行结果：

l

### 任务

补充右边编辑器第8行，使用charAt()方法，返回最后一个字符。

**?**不会了怎么办

mystr.charAt(mystr.length-1)

**正确代码:**

document.write(mystr.charAt(mystr.length-1);

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>string对象 </title>

<script type="text/javascript">

var mystr="I love JavaScript!"

document.write(mystr.charAt(mystr.length-1));

</script>

</head>

<body>

</body>

</html>

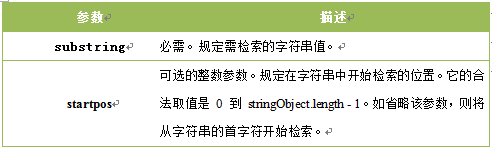
### 7-8返回指定的字符串首次出现的位置

indexOf() 方法可返回某个指定的字符串值在字符串中首次出现的位置。

**语法**

stringObject.indexOf(substring, startpos)

**参数说明：**

[](http://img.mukewang.com/53853d4200019feb04920149.jpg)  
**说明：**

1.该方法将从头到尾地检索字符串 stringObject，看它是否含有子串 substring。

2.可选参数，从stringObject的startpos位置开始查找substring，如果没有此参数将从stringObject的开始位置查找。

3.如果找到一个 substring，则返回 substring 的第一次出现的位置。stringObject 中的字符位置是从 0 开始的。

**注意：**1.indexOf() 方法区分大小写。

2.如果要检索的字符串值没有出现，则该方法返回 -1。

例如: 对 "I love JavaScript!" 字符串内进行不同的检索：

<script type="text/javascript">

var str="I love JavaScript!"

document.write(str.indexOf("I") + "<br />");

document.write(str.indexOf("v") + "<br />");

document.write(str.indexOf("v",8));

</script>

以上代码的输出：

0

4

9

### 任务

补充右边编辑器第8行，使用indexOf()方法，检索第二个字符ｏ出现的位置。

**?**不会了怎么办

mystr.indexof("o",6)

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>string对象 </title>

<script type="text/javascript">

var mystr="Hello World!"

document.write(mystr.indexOf("o",6));

</script>

</head>

<body>

</body>

</html>

### 7-9字符串分割split()

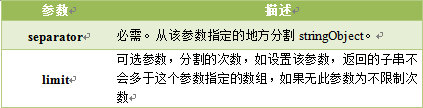
**知识讲解：**

split() 方法将字符串分割为字符串数组，并返回此数组。

**语法：**

stringObject.split(separator,limit)

**参数说明:**

**[](http://img.mukewang.com/532bee4800014c0404230108.jpg)**

**注意：**如果把空字符串 ("") 用作 separator，那么 stringObject 中的每个字符之间都会被分割。

**我们将按照不同的方式来分割字符串：**

使用指定符号分割字符串，代码如下:

var mystr = "www.imooc.com";

document.write(mystr.split(".")+"<br>");

document.write(mystr.split(".", 2)+"<br>");

**运行结果:**

www,imooc,com

www,imooc

将字符串分割为字符，代码如下：

document.write(mystr.split("")+"<br>");

document.write(mystr.split("", 5));

运行结果:

w,w,w,.,i,m,o,o,c,.,c,o,m

w,w,w,.,i

任务

运用split()方法，完成以下任务，并输出:

1. 补充右边编辑器第8行，以字符"-"分隔字符串对象mystr。

2. 补充右边编辑器第9行，将字符串对象mystr每个字符分割。

3. 补充右边编辑器第10行，将字符串对象mystr分割成字符，分割3次。

**?**不会了怎么办

**mystr.split("-")**

**mystr.split("")**

**mystr.split(" ",3)**

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

<script type="text/javascript">

var mystr="86-010-85468578";

document.write(mystr.split("-") + "<br />");

document.write(mystr.split("")+ "<br />");

document.write(mystr.split("",3));

</script>

</head>

<body>

</body>

</html>

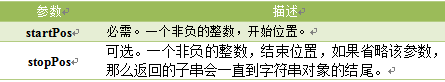
### 7-10提取字符串substring()

substring() 方法用于提取字符串中介于两个指定下标之间的字符。

**语法:**

stringObject.substring(starPos,stopPos)

**参数说明:**



**注意：**

1. 返回的内容是从 start开始(包含start位置的字符)到 stop-1 处的所有字符，其长度为 stop 减start。

2. 如果参数 start 与 stop 相等，那么该方法返回的就是一个空串（即长度为 0 的字符串）。

3. 如果 start 比 stop 大，那么该方法在提取子串之前会先交换这两个参数。

使用 substring() 从字符串中提取字符串，代码如下：

<script type="text/javascript">

  var mystr="I love JavaScript";

  document.write(mystr.substring(7));

  document.write(mystr.substring(2,6));

</script>

**运行结果:**

JavaScript

love

任务

运用substring()方法，完成子字符串的提取:  
1.补充右边编辑器第8行，从字符对象mystr中提取World!子字符串。

2. 补充右边编辑器第9行，从字符对象mystr中提取Hello子字符串。

**?**不会了怎么办

mystr.substring(6)

mystr.substring(0,5)

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>string对象</title>

<script type="text/javascript">

var mystr="Hello World!"

document.write(mystr.substring(6)+ "<br />");

document.write(mystr.substring(0,5));

</script>

</head>

<body>

</body>

</html>

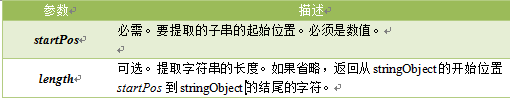
### 7-11提取指定数目的字符substr()

substr() 方法从字符串中提取从 startPos位置开始的指定数目的字符串。

**语法:**

stringObject.substr(startPos,length)

**参数说明:**

[](http://img.mukewang.com/532bf2e00001105305100098.jpg)

**注意：**如果参数startPos是负数，从字符串的尾部开始算起的位置。也就是说，-1 指字符串中最后一个字符，-2 指倒数第二个字符，以此类推。

如果startPos为负数且绝对值大于字符串长度，startPos为0。

使用 substr() 从字符串中提取一些字符，代码如下：

<script type="text/javascript">

var mystr="I love JavaScript!";

document.write(mystr.substr(7));

document.write(mystr.substr(2,4));

</script>

**运行结果：**

JavaScript!

love

### 任务

运用substr()方法，完成以下任务:

1. 补充右边编辑器第8行，提取字符world!。

2. 补充右边编辑器第9行，提取字符Hello。

**?**不会了怎么办

**1.字符串中第一个字符的位置为0。**

**2.字符串中空格也占一个位置。**

3.  mystr.substr(6)

     mystr.substr(0,5)

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

<script type="text/javascript">

var mystr="Hello World!";

document.write(mystr.substr(6) + "<br />");

document.write(mystr.substr(0,5));

</script>

</head>

<body>

</body>

</html>

### 7-12 Math对象

Math对象，提供对数据的数学计算。

使用 Math 的属性和方法，代码如下：

<script type="text/javascript">

var mypi=Math.PI;

var myabs=Math.abs(-15);

document.write(mypi);

document.write(myabs);

</script>

**运行结果:**

3.141592653589793

15

**注意：**Math 对象是一个固有的对象，无需创建它，直接把 Math 作为对象使用就可以调用其所有属性和方法。这是它与Date,String对象的区别。

Math 对象属性



Math 对象方法

### 7-13向上取整ceil()

ceil() 方法可对一个数进行向上取整。

**语法:**

Math.ceil(x)

**参数说明:**

[http://img.mukewang.com/532fe8e00001e4e605230063.jpg](http://img.mukewang.com/532fe8e00001e4e605230063.jpg)

**注意：**它返回的是大于或等于x，并且与x最接近的整数。

我们将把 ceil() 方法运用到不同的数字上，代码如下：

<script type="text/javascript">

document.write(Math.ceil(0.8) + "<br />")

document.write(Math.ceil(6.3) + "<br />")

document.write(Math.ceil(5) + "<br />")

document.write(Math.ceil(3.5) + "<br />")

document.write(Math.ceil(-5.1) + "<br />")

document.write(Math.ceil(-5.9))

</script>

**运行结果：**

1

7

5

4

-5

-5

### 任务

在右边编辑器中补充代码，在script标签内，使用ceil（）方法，计算3.3、-0.1、-9.9、8.9值。

**?**不会了怎么办

ceil()向上取整计算，它返回的是大于或等于的值



以上方法不做全部讲解，只讲解部分方法。此节没有任务，快快进入下节学习。

### 任务

**?**不会了怎么办

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Math </title>

<script type="text/javascript">

document.write(Math.ceil(3.3)+"<br>");

document.write(Math.ceil(-0.1)+"<br>");

document.write(Math.ceil(-9.9)+"<br>");

document.write(Math.ceil(8.9)+"<br>");

</script>

</head>

<body>

</body>

</html>

### 7-14向下取整floor()

floor() 方法可对一个数进行向下取整。

**语法:**

Math.floor(x)

**参数说明：**

http://img.mukewang.com/532fea54000194c604380058.jpg

**注意：**返回的是小于或等于x，并且与x最接近的整数。

我们将在不同的数字上使用 floor() 方法，代码如下:

<script type="text/javascript">

document.write(Math.floor(0.8)+ "<br>")

document.write(Math.floor(6.3)+ "<br>")

document.write(Math.floor(5)+ "<br>")

document.write(Math.floor(3.5)+ "<br>")

document.write(Math.floor(-5.1)+ "<br>")

document.write(Math.floor(-5.9))

</script>

**运行结果：**

0

6

5

3

-6

-6

### 任务

补充右边编辑器在script标签内，使用floor（）方法，计算3.3、-0.1、-9.9、8.9值。

**?**不会了怎么办

floor()向下取整计算，它返回的是大于或等于的值

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Math </title>

<script type="text/javascript">

document.write(Math.floor(3.3)+"<br>");

document.write(Math.floor(-0.1)+"<br>");

document.write(Math.floor(-9.9)+"<br>");

document.write(Math.floor(8.9)+"<br>");

</script>

</head>

<body>

</body>

</html>

### 7-15四舍五入round()

round() 方法可把一个数字四舍五入为最接近的整数。

**语法:**

Math.round(x)

**参数说明：**

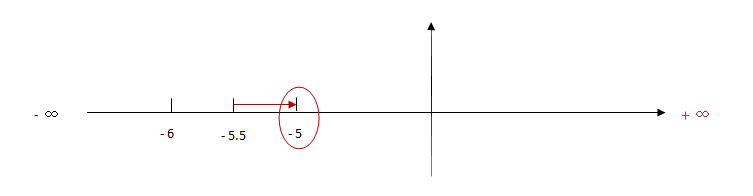
http://img.mukewang.com/532feb70000180ab03210059.jpg

**注意：**

1. 返回与 x 最接近的整数。

2. 对于 0.5，该方法将进行上舍入。(5.5 将舍入为 6)

3. 如果两个整数同等接近 x，则结果是接近 +∞的数字值 。(如 -5.5 将舍入为 -5; -5.52 将舍入为 -6),如下图:

[](http://img.mukewang.com/53fc4cc8000169a907530196.jpg)

把不同的数舍入为最接近的整数,代码如下：

<script type="text/javascript">

document.write(Math.round(1.6)+ "<br>");

document.write(Math.round(2.5)+ "<br>");

document.write(Math.round(0.49)+ "<br>");

document.write(Math.round(-6.4)+ "<br>");

document.write(Math.round(-6.6));

</script>

**运行结果：**

2

3

0

-6

-7

### 任务

补充右边编辑器代码，在script标签内，使用round()方法，计算3.3、-0.1、-9.9、8.9值。

**?**不会了怎么办

对数值进行四舍五入，对于 0.5，该方法将进行上舍入

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Math </title>

<script type="text/javascript">

document.write(Math.round(3.3)+ "<br>");

document.write(Math.round(-0.1)+ "<br>");

document.write(Math.round(-9.9)+ "<br>");

document.write(Math.round(8.9));

</script>

</head>

<body>

</body>

</html>

### 7-16随机数 random()

random() 方法可返回介于 0 ~ 1（大于或等于 0 但小于 1 )之间的一个随机数。

**语法：**

Math.random();

**注意：**返回一个大于或等于 0 但小于 1 的符号为正的数字值。

我们取得介于 0 到 1 之间的一个随机数，代码如下：

<script type="text/javascript">

document.write(Math.random());

</script>

**运行结果：**

0.190305486195328

**注意:**因为是随机数，所以每次运行结果不一样，但是0 ~ 1的数值。

获得0 ~ 10之间的随机数，代码如下:

<script type="text/javascript">

document.write((Math.random())\*10);

</script>

**运行结果：**

8.72153625893887

### 任务

补充右边编辑器代码，在script标签内，使用random ()方法和round()，计算不大于10的整数。

**?**不会了怎么办

使用random()随机取得0-10之间的数值。使用round()对数值进行四舍五入。

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Math </title>

<script type="text/javascript">

document.write(Math.random()\*10);

</script>

</head>

<body>

</body>

</html>

### 7-17Array 数组对象

数组对象是一个对象的集合，里边的对象可以是不同类型的。数组的每一个成员对象都有一个“下标”，用来表示它在数组中的位置，是从零开始的

**数组定义的方法：**

1. 定义了一个空数组:

var  数组名= new Array();

2. 定义时指定有n个空元素的数组：

var 数组名 =new Array(n);

3.定义数组的时候，直接初始化数据：

var  数组名 = [<元素1>, <元素2>, <元素3>...];

我们定义myArray数组，并赋值，**代码如下：**

var myArray = [2, 8, 6];

**说明：**定义了一个数组 myArray，里边的元素是：myArray[0] = 2; myArray[1] = 8; myArray[2] = 6。

**数组元素使用：**

数组名[下标] = 值;

**注意**: 数组的下标用方括号括起来，从0开始。

**数组属性：**

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。

**数组方法：**



以上方法不做全部讲解，只讲解部分方法。此节没有任务，快快进入下节学习。

### 任务

**?**不会了怎么办

### 7-18数组连接concat()

concat() 方法用于连接两个或多个数组。此方法返回一个新数组，不改变原来的数组。

**语法**

arrayObject.concat(array1,array2,...,arrayN)

**参数说明：**

****

**注意:**该方法不会改变现有的数组，而仅仅会返回被连接数组的一个副本。

我们创建一个数组，将把 concat() 中的参数连接到数组 myarr 中，代码如下：

<script type="text/javascript">

  var mya = new Array(3);

  mya[0] = "1";

  mya[1] = "2";

  mya[2] = "3";

  document.write(mya.concat(4,5)+"<br>");

  document.write(mya);

</script>

**运行结果：**

1,2,3,4,5

1,2,3

我们创建了三个数组，然后使用 concat() 把它们连接起来，代码如下：

<script type="text/javascript">

var mya1= new Array("hello!")

  var mya2= new Array("I","love");

  var mya3= new Array("JavaScript","!");

  var mya4=mya1.concat(mya2,mya3);

  document.write(mya4);

</script>

**运行结果：**

hello!,I,love,JavaScript,!

### 任务

补充右边编辑器第9行，使用concat()方法连接myarr1和myarr2两个数组并输出。

**?**不会了怎么办

### 使用concat()完成数组连接

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Array对象 </title>

<script type="text/javascript">

var myarr1= new Array("010")

var myarr2= new Array("-","84697581");

document.write(myarr1.concat(myarr2));

</script>

</head>

<body>

</body>

</html>

### 7-19指定分隔符连接数组元素join()

join()方法用于把数组中的所有元素放入一个字符串。元素是通过指定的分隔符进行分隔的。

**语法：**

arrayObject.join(分隔符)

**参数说明:**

http://img.mukewang.com/533297ff0001516905150059.jpg

注意：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。这个方法不影响数组原本的内容。 我们使用join（）方法，将数组的所有元素放入一个字符串中，代码如下：

<script type="text/javascript">

var myarr = new Array(3);

myarr[0] = "I";

myarr[1] = "love";

myarr[2] = "JavaScript";

document.write(myarr.join());

</script>

**运行结果：**

I,love,JavaScript

我们将使用分隔符来分隔数组中的元素，代码如下：

<script type="text/javascript">

var myarr = new Array(3)

myarr[0] = "I";

myarr[1] = "love";

myarr[2] = "JavaScript";

document.write(myarr.join("."));

</script>

**运行结果：**

I.love.JavaScript

### 任务

补充右边编辑器第10行，使用join()方法,用分隔符"-",将数组myarr3中元素连接，并输出。

**?**不会了怎么办

1. 使用join()完成数组分隔。  
2. document.write(myarr3.join("-"));

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Array对象 </title>

<script type="text/javascript">

var myarr1= new Array("86","010")

var myarr2= new Array("84697581");

var myarr3= myarr1.concat(myarr2);

document.write(myarr3.join("-"));

</script>

</head>

<body>

</body>

</html>

### 7-20颠倒数组元素顺序reverse()

reverse() 方法用于颠倒数组中元素的顺序。

**语法：**

arrayObject.reverse()

**注意：**该方法会改变原来的数组，而不会创建新的数组。

定义数组myarr并赋值，然后颠倒其元素的顺序：

<script type="text/javascript">

var arr = new Array(3)

arr[0] = "1"

arr[1] = "2"

arr[2] = "3"

document.write(myarr + "<br />")

document.write(myarr.reverse())

</script>

**运行结果：**

1,2,3

3,2,1

### 任务

补充右边编辑器第8行，使用reverse()方法,将数组myarr1顺序颠倒，并输出。

**?**不会了怎么办

### 使用reverse()颠倒数组顺序

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Array对象 </title>

<script type="text/javascript">

var myarr1= ["我","爱","你"];

document.write(myarr1.reverse());

</script>

</head>

<body>

</body>

</html>

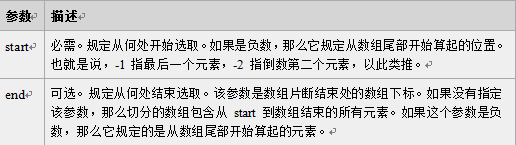
### 7-21选定元素slice()

slice() 方法可从已有的数组中返回选定的元素。

**语法**

arrayObject.slice(start,end)

**参数说明：**



1.返回一个新的数组，包含从 start 到 end （不包括该元素）的 arrayObject 中的元素。

2. 该方法并不会修改数组，而是返回一个子数组。

**注意：**

1. 可使用负值从数组的尾部选取元素。

2.如果 end 未被规定，那么 slice() 方法会选取从 start 到数组结尾的所有元素。

3. String.slice() 与 Array.slice() 相似。

我们将创建一个新数组，然后从其中选取的元素，代码如下：

<script type="text/javascript">

var myarr = new Array(1,2,3,4,5,6);

document.write(myarr + "<br>");

document.write(myarr.slice(2,4) + "<br>");

document.write(myarr);

</script>

**运行结果：**

1,2,3,4,5,6

3,4

1,2,3,4,5,6

### 任务

补充右边编辑器第8行，将选定数组元素"爱","你"，并输出。

**?**不会了怎么办

### 使用slice()选定数组元素

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Array对象 </title>

<script type="text/javascript">

var myarr1= ["我","爱","你"];

document.write(myarr1.slice(1,3));

</script>

</head>

<body>

</body>

</html>

### 7-22数组排序sort()

**sort()**方法使数组中的元素按照一定的顺序排列。

**语法:**

arrayObject.sort(方法函数)

**参数说明：**

http://img.mukewang.com/53329a2a000127f705170060.jpg

1.如果不指定<方法函数>，则按unicode码顺序排列。

2.如果指定<方法函数>，则按<方法函数>所指定的排序方法排序。

myArray.sort(sortMethod);

使用sort()将数组进行排序，代码如下：

<script type="text/javascript">

var myarr1 = new Array("Hello","John","love","JavaScript");

var myarr2 = new Array("80","16","50","6","100","1");

document.write(myarr1.sort()+"<br>");

document.write(myarr2.sort());

</script>

**运行结果：**

Hello,JavaScript,John,love

1,100,16,50,6,80

**注意:上面的代码没有按照数值的大小对数字进行排序。**

**如要实现这一点，就必须使用一个排序函数，代码如下：**

<script type="text/javascript">

function sortNum(a,b) {

return a - b;

//升序，如降序，把“a - b”该成“b - a”

}

var myarr = new Array("80","16","50","6","100","1");

document.write(myarr + "<br>");

document.write(myarr.sort(sortNum));

</script>

**运行结果：**

80,16,50,6,100,1

1,6,16,50,80,100

### 任务

1. 补充右边编辑器第8行，完成降序排列函数。

2. 补充右边编辑器第11行，使用sort()方法，进行数组降序列排列，并输出。

**?**不会了怎么办

1.排序函数，降序 return b - a;

2. document.write(myarr.sort(sortNum));

**代码：**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Array对象 </title>

<script type="text/javascript">

function sortNum(a,b) {

return b-a;

}

var myarr = new Array("80","16","50","6","100","1");

document.write(myarr.sort(sortNum));

</script>

</head>

<body>

</body>

</html>

### 7-23编程练习

某班的成绩出来了，现在老师要把班级的成绩打印出来。

**效果图:**

2014年5月9日 星期六--班级总分为:81

**格式要求：**

1、显示打印的日期。 格式为类似“2014年03月21日 星期三” 的当前的时间。

2、计算出该班级的平均分（保留整数）。

**同学成绩数据如下：**

"小明:87; 小花:81; 小红:97; 小天:76;小张:74;小小:94;小西:90;小伍:76;小迪:64;小曼:76"

### 任务

第一步：可通过javascript的日期对象来得到当前的日期。

提示:使用Date()日期对象，注意星期返回值为0-6，所以要转成文字"星期X"

第二步：一长窜的字符串不好弄，找规律后分割放到数组里更好操作哦。

第三步：分割字符串得到分数，然后求和取整。

提示：parseInt() 字符串类型转成整型。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>系好安全带,准备启航</title>

<script type="text/javascript">

//通过javascript的日期对象来得到当前的日期，并输出。

var date=new Date();

var weekday=["星期日","星期一","星期二","星期三","星期四","星期五","星期六"];

document.write(date.getFullYear()+"年"+date.getMonth()+"月"+date.getDate()+"日"+weekday[date.getDay()]+"<br />");

//成绩是一长窜的字符串不好处理，找规律后分割放到数组里更好操作哦

var sorceStr = "小明:87;小花:81;小红:97;小天:76;小张:74;小小:94;小西:90;小伍:76;小迪:64;小曼:76";

var sn=sorceStr.split(";");

//var s1=sn[1].split(":");

//document.write(s1[1]+" ");

var sum=0;

for (var i=0;i<sn.length;i++){

var s1=sn[i].split(":");

var s2=parseInt(s1[1]);

sum=s2+sum;

}

document.write(sum+" ");

//从数组中将成绩撮出来，然后求和取整，并输出。

</script>

</head>

<body>

</body>

</html>

## ****第8章 浏览器对象****

### 8-1 window对象

window对象是BOM的核心，window对象指当前的浏览器窗口。

**window对象方法:**

**[](http://img.mukewang.com/535483720001a54506670563.jpg)**

**注意:**在JavaScript基础篇中，已讲解了部分属性，window对象重点讲解计时器。

### 任务

在右边编辑器script标签内补充代码，弹出对话框"欢迎来到慕课网"。

定义一个函数，实现打开一个网页，宽为600，高为400。

当点击"点击我，打开新窗口"按钮时，在打开网页。

**?**不会了怎么办

如果忘记了，可以查看JavaScript基础篇。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>window对象</title>

<script type="text/javascript">

function dianji(){

alert("点一点我！");

window.open('http://www.baidu.com','\_blank','width=600px,height=400px');

}

</script>

</head>

<body>

<form>

<input type="button" value="点击我，打开新窗口" onclick="dianji()" />

</form>

</body>

</html>

### 8-2 JavaScript 计时器

在JavaScript中，我们可以在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。  
**计时器类型：**  
一次性计时器：仅在指定的延迟时间之后触发一次。  
间隔性触发计时器：每隔一定的时间间隔就触发一次。  
**计时器方法：**

[](http://img.mukewang.com/534b35410001d7bc04090143.jpg)

### 任务

本节没有任务，快快进入下节学习计时器。

**?**不会了怎么办

### 8-3计时器setInterval()

在执行时,从载入页面后每隔指定的时间执行代码。

**语法:**

setInterval(代码,交互时间);

**参数说明：**

1. 代码：要调用的函数或要执行的代码串。

2. 交互时间：周期性执行或调用表达式之间的时间间隔，以毫秒计（1s=1000ms）。

**返回值:**

一个可以传递给 clearInterval() 从而取消对"代码"的周期性执行的值。

**调用函数格式(**假设有一个clock()函数**):**

setInterval("clock()",1000)

或

setInterval(clock,1000)

我们设置一个计时器，每隔100毫秒调用clock()函数，并将时间显示出来，**代码如下:**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>计时器</title>

<script type="text/javascript">

var int=setInterval(clock, 100)

function clock(){

var time=new Date();

document.getElementById("clock").value = time;

}

</script>

</head>

<body>

<form>

<input type="text" id="clock" size="50" />

</form>

</body>

</html>

### 任务

补充右边编辑器第10行，获取时间，格式"时:分:秒"，并赋值给attime。

补充右边编辑器第13行，使用setInterval()计时器来显示动态时间。

**?**不会了怎么办

setInterval(clock,100)

attime=time.getHours()+":"+time.getMinutes()+":"+time.getSeconds();

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>定时器</title>

<script type="text/javascript">

var attime;

function clock(){

var time=new Date();

attime=time.getHours()+"时"+time.getMinutes()+"分"+time.getSeconds()+"秒";

document.getElementById("clock").value = attime;

}

var int=setInterval(clock,1000);

</script>

</head>

<body>

<form>

<input type="text" id="clock" size="50" />

</form>

</body>

</html>

### 8-4取消计时器clearInterval()

clearInterval() 方法可取消由 setInterval() 设置的交互时间。

**语法：**

clearInterval(id\_of\_setInterval)

**参数说明:**  
id\_of\_setInterval：由 setInterval() 返回的 ID 值。

每隔 100 毫秒调用 clock() 函数,并显示时间。当点击按钮时，停止时间,代码如下:

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>计时器</title>

<script type="text/javascript">

function clock(){

var time=new Date();

document.getElementById("clock").value = time;

}

// 每隔100毫秒调用clock函数，并将返回值赋值给i

var i=setInterval("clock()",100);

</script>

</head>

<body>

<form>

<input type="text" id="clock" size="50" />

<input type="button" value="Stop" onclick="clearInterval(i)" />

</form>

</body>

</html>

### 任务

补充右边编辑器第12行，使用setInterval()计时器，显示动态时间。

补充右边编辑器第18行，点击Stop按钮后，取消计时器。

**?**不会了怎么办

要取消setInterval()计时器，要获取它的返回值ID

使用clearInterval()取消，参数是setInterval()计时器的返回值ID

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>计时器</title>

<script type="text/javascript">

function clock(){

var time=new Date();

document.getElementById("clock").value = time;

}

var i = setInterval("clock()",100)

</script>

</head>

<body>

<form>

<input type="text" id="clock" size="50" />

<input type="button" value="Stop" onclick="clearInterval(i)" />

</form>

</body>

</html>

### 8-5计时器setTimeout()

setTimeout()计时器，在载入后延迟指定时间后,去执行一次表达式,仅执行一次。

**语法:**

setTimeout(代码,延迟时间);

**参数说明：**

1. 要调用的函数或要执行的代码串。  
2. 延时时间：在执行代码前需等待的时间，以毫秒为单位（1s=1000ms)。

当我们打开网页3秒后，在弹出一个提示框，**代码如下:**

<!DOCTYPE HTML>

<html>

<head>

<script type="text/javascript">

setTimeout("alert('Hello!')", 3000 );

</script>

</head>

<body>

</body>

</html>

当按钮start被点击时，setTimeout()调用函数，在5秒后弹出一个提示框。

<!DOCTYPE HTML>

<html>

<head>

<script type="text/javascript">

function tinfo(){

var t=setTimeout("alert('Hello!')",5000);

}

</script>

</head>

<body>

<form>

<input type="button" value="start" onClick="tinfo()">

</form>

</body>

</html>

要创建一个运行于无穷循环中的计数器，我们需要编写一个函数来调用其自身。在下面的代码，当按钮被点击后，输入域便从0开始计数。

<!DOCTYPE HTML>

<html>

<head>

<script type="text/javascript">

var num=0;

function numCount(){

document.getElementById('txt').value=num;

num=num+1;

setTimeout("numCount()",1000);

}

</script>

</head>

<body>

<form>

<input type="text" id="txt" />

<input type="button" value="Start" onClick="numCount()" />

</form>

</body>

</html>

### 任务

补充右边编辑器第12和14行，使用setTimeout()实现计数统计效果，并在文本框中显示数值。

**?**不会了怎么办

第12行，在函数内容使用setTimeout()，实现自身调用，创建一个循环计数器。

第14行，使用setTimeout()，在设定时间后执行startCount()函数。

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>计时器</title>

</head>

<script type="text/javascript">

var num=0;

function startCount() {

document.getElementById('count').value=num;

num=num+1;

setTimeout("startCount()",1000);

}

setTimeout("startCount()",1000);

</script>

</head>

<body>

<form>

<input type="text" id="count" />

</form>

</body>

</html>

### 8-6取消计时器clearTimeout()

setTimeout()和clearTimeout()一起使用，停止计时器。

**语法:**

clearTimeout(id\_of\_setTimeout)

**参数说明:**  
**id\_of\_setTimeout：**由 setTimeout() 返回的 ID 值。该值标识要取消的延迟执行代码块。

下面的例子和上节的无穷循环的例子相似。唯一不同是，现在我们添加了一个 "Stop" 按钮来停止这个计数器：

<!DOCTYPE HTML>

<html>

<head>

<script type="text/javascript">

var num=0,i;

function timedCount(){

    document.getElementById('txt').value=num;

    num=num+1;

    i=setTimeout(timedCount,1000);

  }

  setTimeout(timedCount,1000);

  function stopCount(){

  clearTimeout(i);

  }

</script>

</head>

<body>

<form>

<input type="text" id="txt">

<input type="button" value="Stop" onClick="stopCount()">

</form>

</body>

</html>

### 任务

补充右边编辑器第16行，完成取消计时器函数。

补充右边编辑器第23行，点击Start按钮后，开始计数。

补充右边编辑器第24行，点击Stop按钮后，停止计数。

**?**不会了怎么办

要取消setTimeout()计时器，要获取它的返回值ID

使用clearTimeout()取消，参数是setInterval()计时器的返回值ID

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>计时器</title>

</head>

<script type="text/javascript">

var num=0;

var i;

function startCount(){

document.getElementById('count').value=num;

num=num+1;

i=setTimeout("startCount()",1000);

}

function stopCount(){

clearTimeout(i);

}

</script>

</head>

<body>

<form>

<input type="text" id="count" />

<input type="button" value="Start" onclick="startCount()" />

<input type="button" value="Stop" onclick="stopCount()"/>

</form>

</body>

</html>

### 8-7 History 对象

history对象记录了用户曾经浏览过的页面(URL)，并可以实现浏览器前进与后退相似导航的功能。

**注意:**从窗口被打开的那一刻开始记录，每个浏览器窗口、每个标签页乃至每个框架，都有自己的history对象与特定的window对象关联。

**语法：**

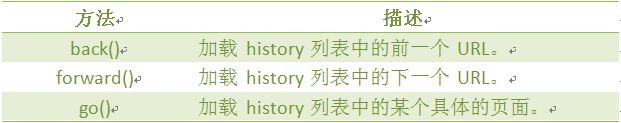
window.history.[属性|方法]

**注意：**window可以省略。

**History 对象属性**

[](http://img.mukewang.com/53548c030001759e05840068.jpg)

**History 对象方法**

**[](http://img.mukewang.com/53548c200001228206210123.jpg)**

使用length属性，当前窗口的浏览历史总长度，**代码如下：**

<script type="text/javascript">

var HL = window.history.length;

document.write(HL);

</script>

### 任务

补充右边编辑器第8行，获取浏览器历史列表中的URL数量。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>History对象</title>

</head>

<script type="text/javascript">

var HL =window.history.length;

document.write(HL);

</script>

<body>

</body>

</html>

### 8-8返回前一个浏览的页面

back()方法，加载 history 列表中的前一个 URL。

**语法：**

window.history.back();

比如，返回前一个浏览的页面，**代码如下：**

window.history.back();

**注意：等同于点击浏览器的倒退按钮。**

back()相当于go(-1),**代码如下:**

window.history.go(-1);

### 任务

补充右边代码编辑器第9行,完善GoBack()函数，实现返回前一个页面功能。

**注意:**本编辑器有局限性，可以把代码拷贝到本机上练习，或者在本节页面窗口多打开几个网页后(为了添加历史列表项)，在打开本节进行练习。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<script type="text/javascript">

function GoBack() {

window.history.back();

}

</script>

</head>

<body>

点击下面的锚点链接，添加历史列表项:

<br />

<a href="#target1">第一个锚点</a>

<a name="target1"></a>

<br />

<a href="#target2">第二个锚点</a>

<a name="target2"></a>

<br /><br />

使用下面按钮，实现返回前一个页面:

<form>

<input type="button" value="返回前一个页面" onclick="GoBack();" />

</form>

</body>

</html>

### 8-9返回下一个浏览的页面

forward()方法，加载 history 列表中的下一个 URL。

如果倒退之后，再想回到倒退之前浏览的页面，则可以使用forward()方法,**代码如下:**

window.history.forward();

**注意：等价点击前进按钮。**

forward()相当于go(1),**代码如下:**

window.history.go(1);

### 任务

补充右边代码编辑器第9行,完善GoForward()函数，实现返回下一个页面功能。

**注意:**本编辑器有局限性，可以把代码拷贝到本机上练习，或者在本节页面窗口多打开几个网页后(为了添加历史列表项)，在打开本节进行练习。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<script type="text/javascript">

function GoForward() {

window.history.forward();

}

</script>

</head>

<body>

点击下面的锚点链接，添加历史列表项:

<br />

<a href="#target1">第一个锚点</a>

<a name="target1"></a>

<br />

<a href="#target2">第二个锚点</a>

<a name="target2"></a>

<br /><br />

使用下面按钮，实现返回下一个页面:

<form>

<input type="button" value="返回下一个页面" onclick="GoForward()" />

</form>

</body>

</html>

### 8-10返回浏览历史中的其他页面

go()方法，根据当前所处的页面，加载 history 列表中的某个具体的页面。

**语法：**

window.history.go(number);

**参数：**

[](http://img.mukewang.com/5354947e00011a9a06490153.jpg)

浏览器中，返回当前页面之前浏览过的第二个历史页面，**代码如下：**

window.history.go(-2);

**注意：和在浏览器中单击两次后退按钮操作一样。**

同理，返回当前页面之后浏览过的第三个历史页面，**代码如下：**

window.history.go(3);

### 任务

补充右边代码编辑器第9行和第13行,使用go()方法，完善GoBack()和GoForward()函数，实现返回前或下一个页面功能。

**注意:**本编辑器有局限性，可以把代码拷贝到本机上练习，或者在本节页面窗口多打开几个网页后(为了添加历史列表项)，在打开本节进行练习。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<script type="text/javascript">

function GoBack() {

window.history.go(-1);

}

function GoForward() {

window.history.go(1);

}

</script>

</head>

<body>

点击下面的锚点链接，添加历史列表项:

<br />

<a href="#target1">第一个锚点</a>

<a name="target1"></a>

<br />

<a href="#target2">第二个锚点</a>

<a name="target2"></a>

<br /><br />

使用下面按钮，实现返回前或下一个页面:

<form>

<input type="button" value="返回前一个页面" onclick="GoBack();" />

<input type="button" value="返回下一个页面" onclick="GoForward();" />

</form>

</body>

</html>

### 8-11 Location对象

location用于获取或设置窗体的URL，并且可以用于解析URL。

**语法:**

location.[属性|方法]

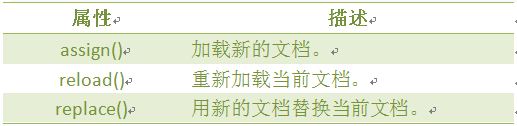
**location对象属性图示:**

[](http://img.mukewang.com/53605c5a0001b26909900216.jpg)

**location 对象属性：**

**[](http://img.mukewang.com/5354b1d00001c4ec06220271.jpg)**

**location 对象方法:**

**[](http://img.mukewang.com/5354b1eb00016a2405170126.jpg)**

### 任务

在右边编辑器script标签内，获取当前显示文档的URL,并输出。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>location</title>

</head>

<script type="text/javascript">

document.write(window.location.href+"<br>");

</script>

</head>

<body>

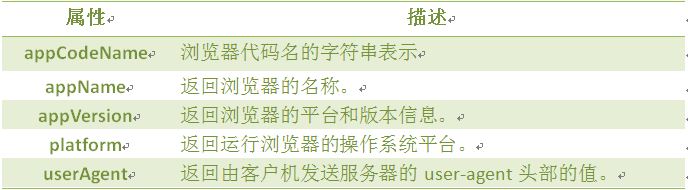
</body>

</html>

### 8-12 Navigator对象

Navigator 对象包含有关浏览器的信息，通常用于检测浏览器与操作系统的版本。

**对象属性:**

[](http://img.mukewang.com/5354cff70001428b06880190.jpg)

查看浏览器的名称和版本，**代码如下:**

<script type="text/javascript">

var browser=navigator.appName;

var b\_version=navigator.appVersion;

document.write("Browser name"+browser);

document.write("<br>");

document.write("Browser version"+b\_version);

</script>

### 任务

在右边编辑器script标签内，使用Navigator对象，查看浏览器相关信息。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<script type="text/javascript">

var browser=navigator.appName;

var b\_version=navigator.appVersion;

document.write("浏览器名称："+browser);

document.write("<br>");

document.write("浏览器版本信息："+b\_version);

</script>

<body>

</body>

</html>

### 8-13 userAgent

返回用户代理头的字符串表示(就是包括浏览器版本信息等的字符串)

**语法**

navigator.userAgent

几种浏览的user\_agent.，像360的兼容模式用的是IE、极速模式用的是chrom的内核。

[](http://img.mukewang.com/535a3a4a0001e03f06870189.jpg)

使用userAgent判断使用的是什么浏览器(假设使用的是IE8浏览器),**代码如下:**

function validB(){

var u\_agent = navigator.userAgent;

var B\_name="Failed to identify the browser";

if(u\_agent.indexOf("Firefox")>-1){

B\_name="Firefox";

}else if(u\_agent.indexOf("Chrome")>-1){

B\_name="Chrome";

}else if(u\_agent.indexOf("MSIE")>-1&&u\_agent.indexOf("Trident")>-1){

B\_name="IE(8-10)";

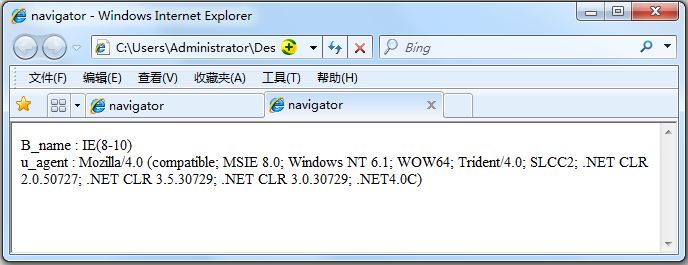
  }

document.write("B\_name:"+B\_name+"<br>");

  document.write("u\_agent:"+u\_agent+"<br>");

}

**运行结果:**

[](http://img.mukewang.com/535dea1e00017b0b06880265.jpg)

### 任务

补充右边编辑器第8行，使用userAgent获取浏览器信息，并保存在u\_agent中。

补充右边编辑器第24行，点击"查看浏览器"按钮，显示浏览器和浏览器相关信息。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>navigator</title>

<script type="text/javascript">

function validB(){

var u\_agent =navigator.userAgent;

var B\_name="不是想用的主流浏览器!";

if(u\_agent.indexOf("Firefox")>-1){

B\_name="Firefox";

}else if(u\_agent.indexOf("Chrome")>-1){

B\_name="Chrome";

}else if(u\_agent.indexOf("MSIE")>-1&&u\_agent.indexOf("Trident")>-1){

B\_name="IE(8-10)";

}

document.write("浏览器:"+B\_name+"<br>");

document.write("u\_agent:"+u\_agent+"<br>");

}

</script>

</head>

<body>

<form>

<input type="button" value="查看浏览器" onclick="validB()">

</form>

</body>

</html>

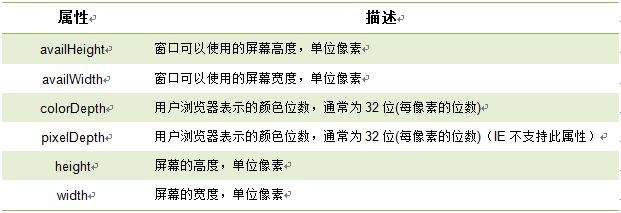
### 8-14 screen对象

screen对象用于获取用户的屏幕信息。

**语法：**

window.screen.属性

**对象属性:**

**[](http://img.mukewang.com/5354d2810001a47706210213.jpg)**

### 任务

此节没有任务，快快进入下一节学习。

**?**不会了怎么办

### 8-15屏幕分辨率的高和宽

window.screen 对象包含有关用户屏幕的信息。  
1. screen.height 返回屏幕分辨率的高  
2. screen.width 返回屏幕分辨率的宽  
**注意:**  
1.单位以像素计。  
2. window.screen 对象在编写时可以不使用 window 这个前缀。  
我们来获取屏幕的高和宽，代码如下:

<script type="text/javascript">

document.write( "屏幕宽度："+screen.width+"px<br />" );

document.write( "屏幕高度："+screen.height+"px<br />" );

</script>

### 任务

使用window.screen对象，完成下面任务,看看你的屏幕大小:  
补充右边编辑器第9行，输出屏幕宽度。  
补充右边编辑器第10行，输出屏幕高度。

**?**不会了怎么办

document.write( "屏幕宽度："+screen.width );

document.write( "屏幕高度："+screen.height);

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>屏幕分辨率的高和宽</title>

</head>

<body>

<script type="text/javascript">

document.write( "屏幕宽度："+screen.width+"px<br>");

document.write( "屏幕高度："+screen.height+"px");

</script>

</body>

</html>

### 8-16屏幕可用高和宽度

1. screen.availWidth 属性返回访问者屏幕的宽度，以像素计，减去界面特性，比如任务栏。

2. screen.availHeight 属性返回访问者屏幕的高度，以像素计，减去界面特性，比如任务栏。

**注意:**

不同系统的任务栏默认高度不一样，及任务栏的位置可在屏幕上下左右任何位置，所以有可能可用宽度和高度不一样。  
  
我们来获取屏幕的可用高和宽度，**代码如下：**

<script type="text/javascript">

document.write("可用宽度：" + screen.availWidth);

document.write("可用高度：" + screen.availHeight);

</script>

**注意:根据屏幕的不同显示值不同。**

### 任务

使用window.screen对象，完成下面任务,看看你的屏幕可用大小:  
补充右边编辑器第9行，输出屏幕可用宽度  
补充右边编辑器第10行，输出屏幕可用高度  
**注意:** 大家可以把任务栏的位置和大小改变下，看看获取的值有什么区别。

**?**不会了怎么办

document.write("可用宽度：" + screen.availWidth);

document.write("可用高度：" + screen.availHeight);

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>屏幕分辨率的高和宽</title>

</head>

<body>

<script type="text/javascript">

document.write("可用宽度："+screen.availWidth+"px<br>");

document.write("可用高度："+screen.availHeight+"px");

</script>

</body>

</html>

### 8-17编程练习

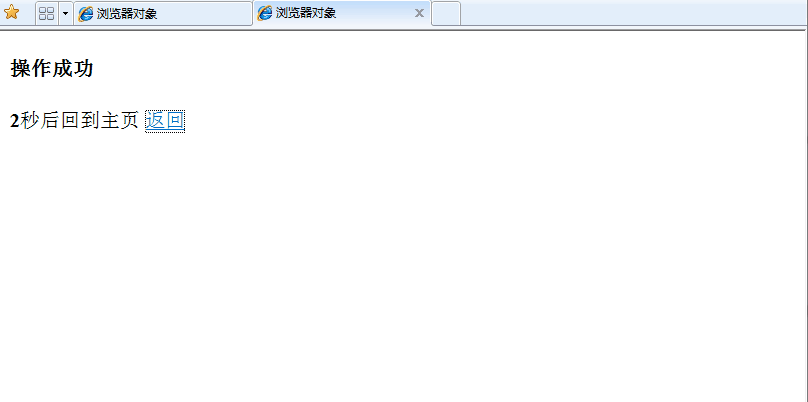
制作一个跳转提示页面：

**要求：**

1. 如果打开该页面后，如果不做任何操作则5秒后自动跳转到一个新的地址，如慕课网主页。

2. 如果点击“返回”按钮则返回前一个页面。

**效果:**

**[](http://img.mukewang.com/537d6fed0001572608080402.jpg)**

### 任务

第一步： 先编写好网页布局，如下:

[](http://img.mukewang.com/537d6eba0001779f03840115.jpg)

第二步： 获取显示秒数的元素，通过定时器来更改秒数。

第三步： 通过window的location和history对象来控制网页的跳转。

**?**不会了怎么办

**代码：**

<!DOCTYPE html>

<html>

<head>

<title>浏览器对象</title>

<meta http-equiv="Content-Type" content="text/html; charset=gkb"/>

</head>

<body>

<!--先编写好网页布局-->

<h1>操作成功</h1>

<p>&nbsp;<strong id="shu">5</strong>秒后回到主页&nbsp;<a href="">返回</a></p>

<script type="text/javascript">

//获取显示秒数的元素，通过定时器来更改秒数。

var s=document.getElementById("shu");

var i=5;

function clock(){

i--;

if(i==1){window.location.href="http://www.imooc.com/"}

s.innerHTML=i;

}

var timer=setInterval(clock,1000);

//通过window的location和history对象来控制网页的跳转。

</script>

</body>

</html>

## ****第9章 DOM对象，控制HTML元素****

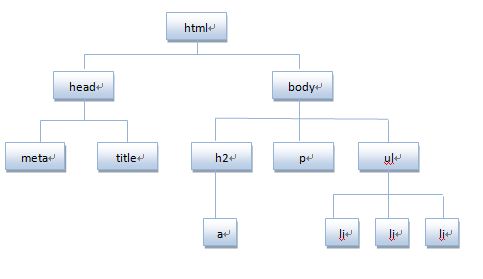
### 9-1认识DOM

文档对象模型DOM（Document Object Model）定义访问和处理HTML文档的标准方法。DOM 将HTML文档呈现为带有元素、属性和文本的树结构（节点树）。

**先来看看下面代码:**

[](http://img.mukewang.com/5375ca640001c67307860420.jpg)

**将HTML代码分解为DOM节点层次图:**

[](http://img.mukewang.com/5375ca7e0001dd8d04830279.jpg)

**HTML文档可以说由节点构成的集合，DOM节点有:**

**1. 元素节点：**上图中<html>、<body>、<p>等都是元素节点，即标签。

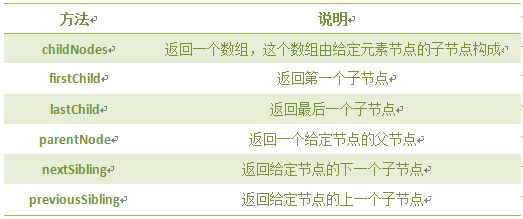
**2. 文本节点:**向用户展示的内容，如<li>...</li>中的JavaScript、DOM、CSS等文本。

**3. 属性节点:**元素属性，如<a>标签的链接属性href="http://www.imooc.com"。

**节点属性:**

[](http://img.mukewang.com/5375c953000117ee05240129.jpg)

**遍历节点树:**

[](http://img.mukewang.com/53f17a6400017d2905230219.jpg)

**以上图ul为例，它的父级节点body,它的子节点3个li,它的兄弟结点h2、P。**

**DOM操作:**

[](http://img.mukewang.com/538d29da000152db05360278.jpg)

**注意:**前两个是document方法。

### 任务

**在JavaScript入门篇中，对DOM进行了基础讲解（http://www.imooc.com/learn/36），我们通过任务来回顾下。**

在右边编辑器第11行补充代码，通过getElementById获取id为con的h2标签。

在右边编辑第12行补充代码，修改h2标签的样式，将颜色设为红色。

在右边编辑器第13行补充代码，修改h2标签的样式，将背景颜色设为灰色(#CCC)。

在右边编辑第14行补充代码，通过style.display实现隐藏。

**?**不会了怎么办

var mychar = document.getElementById("con");

mychar.style.color="red";

mychar.style.backgroundColor ="#ccc";

mychar.style.display="none";

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>style样式</title>

</head>

<body>

<h2 id="con">I love JavaScript</H2>

<p> JavaScript使网页显示动态效果并实现与用户交互功能。</p>

<script type="text/javascript">

var mychar = document.getElementById("con");

mychar.style.color="red";

mychar.style.backgroundColor ="#ccc";

mychar.style.display="none";

</script>

</body>

</html>

### 9-2 getElementsByName()方法

返回带有指定名称的节点对象的集合。

**语法：**

document.getElementsByName(name)

与getElementById() 方法不同的是，通过元素的 name 属性查询元素，而不是通过 id 属性。

**注意:**

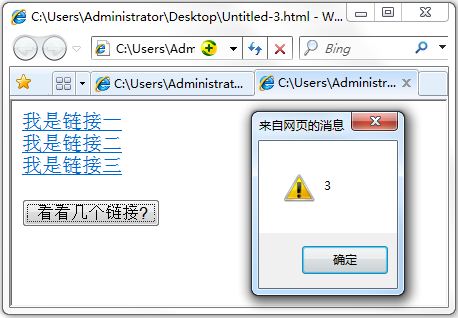
1. 因为文档中的 name 属性可能不唯一，所有 getElementsByName() 方法返回的是元素的数组，而不是一个元素。

2. 和数组类似也有length属性，可以和访问数组一样的方法来访问，从0开始。

**看看下面的代码:**

[](http://img.mukewang.com/5375d5ec00012bac06210322.jpg)

**运行结果:**

[](http://img.mukewang.com/53795f0b0001233404580318.jpg)

### 任务

试一试，通过name属性来查询元素:

补充第6行代码，获取 name 为 myt 的元素，同时输出获取的元素节点的长度。

**?**不会了怎么办

var mynode=document.getElementsByName("myt");

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<script type="text/javascript">

function getnum(){

var mynode=document.getElementsByName("myt");

alert(mynode.length);

}

</script>

</head>

<body>

<input name="myt" type="text" value="1">

<input name="myt" type="text" value="2">

<input name="myt" type="text" value="3">

<input name="myt" type="text" value="4">

<input name="myt" type="text" value="5">

<input name="myt" type="text" value="6">

<br />

<input type="button" onclick="getnum()" value="看看有几项？" />

</body>

</html>

### 9-3 getElementsByTagName()方法

返回带有指定标签名的节点对象的集合。返回元素的顺序是它们在文档中的顺序。

**语法:**

getElementsByTagName(Tagname)

**说明:**

1. Tagname是标签的名称，如p、a、img等标签名。

2. 和数组类似也有length属性，可以和访问数组一样的方法来访问，所以从0开始。

看看下面代码，通过getElementsByTagName()获取节点。

[](http://img.mukewang.com/53ec174a0001404206540436.jpg)

### 任务

**试一试，使用三种获取节点的方法，完成下面的任务:**

在第73行补充代码，通过ID获取标题H1。

在第78行补充代码，通过name获取值为sex的元素。

在第84行补充代码，通过标签名获取input元素。

**?**不会了怎么办

1. var myH=document.getElementById("myHead");

2. var myS=document.getElementsByName("sex");

3. var myI = document.getElementsByTagName("input");

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>JavaScript</title>

</head>

<body>

<form name="Input">

<table align="center" width="500px" height="50%" border="1">

<tr>

<td align="center" width="100px">

学号:

</td>

<td align="center" width="300px">

<input type="text" id=userid name="user" onblur="validate();">

<div id=usermsg></div>

</td>

</tr>

<tr>

<td align="center" width="100px">

姓名:

</td>

<td align="center">

<input type="text" name="name">

</td>

</tr>

<tr>

<td align="center" width="%45">

性别:

</td>

<td align="center">

<input type="radio" name="sex" value="男">

男

<input type="radio" name="sex" value="女">

女

</td>

</tr>

<tr>

<td align="center" width="30%">

年龄:

</td>

<td align="center" width="300px">

<input type="text" name="age">

</td>

</tr>

<tr>

<td align="center" width="100px">

地址:

</td>

<td align="center" width="300px">

<input type="text" name="addr">

</td>

</tr>

</table>

</form>

<h1 id="myHead" onclick="getValue()">

看看三种获取节点的方法?

</h1>

<p>

点击标题弹出它的值。

</p>

<input type="button" onclick="getElements()"

value="看看name为sex的节点有几个?" />

<Br>

<input type="button" onclick="getTagElements()"

value="看看标签名为input的节点有几个?" />

<script type="text/javascript">

function getValue()

{

var myH=document.getElementById("myHead");

alert(myH.innerHTML)

}

function getElements()

{

var myS=document.getElementsByName("sex");

alert(myS.length);

}

function getTagElements()

{

var myI=document.getElementsByTagName("input");

alert(myI.length);

}

</script>

</body>

</html>

### **9-4区别**getElementByID,getElementsByName,getElementsByTagName以人来举例说明，人有能标识身份的身份证，有姓名，有类别(大人、小孩、老人)等。

1. ID 是一个人的身份证号码，是唯一的。所以通过getElementsById获取的是指定的一个人。

2. Name 是他的名字，可以重复。所以通过getElementsByName获取名字相同的人集合。

3. TagName可看似某类，getElementBysTagName获取相同类的人集合。如获取小孩这类人，getElementsByTagName("小孩")。

**把上面的例子转换到HTML中，如下:**

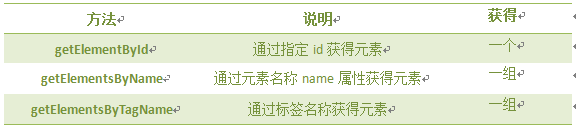
<input type="checkbox" name="hobby" id="hobby1"> 音乐

input标签就像人的类别。

name属性就像人的姓名。

id属性就像人的身份证。

**方法总结如下:**

[](http://img.mukewang.com/5405263300018bcf05760129.jpg)

**注意：**方法区分大小写

通过下面的例子(6个name="hobby"的复选项，两个按钮)来区分三种方法的不同:

<input type="checkbox" name="hobby" id="hobby1"> 音乐

<input type="checkbox" name="hobby" id="hobby2"> 登山

<input type="checkbox" name="hobby" id="hobby3"> 游泳

<input type="checkbox" name="hobby" id="hobby4"> 阅读

<input type="checkbox" name="hobby" id="hobby5"> 打球

<input type="checkbox" name="hobby" id="hobby6"> 跑步

<input type="button" value = "全选" id="button1">

<input type="button" value = "全不选" id="button1">

1. document.getElementsByTagName("input")，结果为获取所有标签为input的元素，共8个。

2. document.getElementsByName("hobby")，结果为获取属性name="hobby"的元素，共6个。

3. document.getElementById("hobby6")，结果为获取属性id="hobby6"的元素，只有一个，"跑步"这个复选项。

### 任务

1.在第27行处补充完整，实现当点击"全选"按钮时，将选中所有的复选项。

**提示:**document.getElementsByTagName("input")获取的是所有inupt标签，包括复选项和按钮，所以要判断是否是复选项，如是选中。

2.在第33行处补充完整，实现当点击"全不选"按钮时，将取消所有选中的复选项。

3.在第40行处补充完整，在文本框中输入输入1-6数值，当点击"确定"按钮时，根据输入的数值，通过id选中相应的复选项。

**?**不会了怎么办

<!DOCTYPE HTML>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=gb2312">  
        <title>无标题文档</title>  
    </head>  
      
    <body>  
        <form>  
          请选择你爱好:<br>  
          <input type="checkbox" name="hobby" id="hobby1">  音乐  
          <input type="checkbox" name="hobby" id="hobby2">  登山  
          <input type="checkbox" name="hobby" id="hobby3">  游泳  
          <input type="checkbox" name="hobby" id="hobby4">  阅读  
          <input type="checkbox" name="hobby" id="hobby5">  打球  
          <input type="checkbox" name="hobby" id="hobby6">  跑步 <br>  
          <input type="button" value = "全选" onclick = "checkall();">  
          <input type="button" value = "全不选" onclick = "clearall();">  
          <p>请输入您要选择爱好的序号，序号为1-6:</p>  
          <input id="wb" name="wb" type="text" >  
          <input name="ok" type="button" value="确定" onclick = "checkone();">  
        </form>  
        <script type="text/javascript">  
        function checkall(){  
            var hobby = document.getElementsByTagName("input");  
            for(i = 0;i < hobby.length;i++){  
                    if(hobby[i].type == "checkbox")  
                    {  
                        hobby[i].checked = true;  
                    }  
                }  
        }  
        function clearall(){  
            var hobby = document.getElementsByName("hobby");  
            for(i = 0;i < hobby.length;i++){  
                hobby[i].checked = false;  
            }  
        }  
          
        function checkone(){  
            var j=document.getElementById("wb").value;  
            var hobby = document.getElementById("hobby"+j);  
            hobby.checked = true;  
        }  
          
        </script>  
    </body>  
</html>

### 9-5 getAttribute()方法

通过元素节点的属性名称获取属性的值。

**语法：**

elementNode.getAttribute(name)

**说明:**

1. elementNode：使用getElementById()、getElementsByTagName()等方法，获取到的元素节点。

2. name：要想查询的元素节点的属性名字

看看下面的代码，获取h1标签的属性值：

[](http://img.mukewang.com/538d242700019ec809330432.jpg)

**运行结果:**

h1标签的ID ：alink  
h1标签的title ：getAttribute()获取标签的属值

### 任务

**试一试，使用getAttribute()方法，完成下面的任务:**

在第20行补充代码，使用getAttribute()方法，获取LI标签的title值。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>getAttribute()</title>

</head>

<body>

<p id="intro">课程列表</p>

<ul>

<li title="第1个li">HTML</li>

<li>CSS</li>

<li title="第3个li">JavaScript</li>

<li title="第4个li">Jquery</li>

<li>Html5</li>

</ul>

<p>以下为获取的不为空的li标签title值:</p>

<script type="text/javascript">

var con=document.getElementsByTagName("li");

for (var i=0; i< con.length;i++){

var text=con[i].getAttribute("title");

if(text!=null)

{

document.write(text+"<br>");

}

}

</script>

</body>

</html>

笔记：

1. elementNode.getAttribute(name)  
   1. elementNode：使用getElementById()、getElementsByTagName()等方法，获取到的元素节点。  
   2. name：要想查询的元素节点的属性名字
2. con = document.getElementsByTagName 返回的是数组  
   可以用 con.item(i ) 或者 con[i] 找到该标签的对象，用 con[i].getAttribute(" id/title/class ") 返回标签内的属性
3. getAttribute()方法  
   通过元素节点的属性名称获取属性的值。  
   语法：  
   elementNode.getAttribute(name)  
   说明:  
   1. elementNode：使用getElementById()、getElementsByTagName()等方法，获取到的元素节点。  
   2. name：要想查询的元素节点的属性名字
4. 注意for循环

### 9-6 setAttribute()方法

setAttribute() 方法增加一个指定名称和值的新属性，或者把一个现有的属性设定为指定的值。

**语法：**

elementNode.setAttribute(name,value)

**说明：**

1.name: 要设置的属性名。

2.value: 要设置的属性值。

**注意：**

1.把指定的属性设置为指定的值。如果不存在具有指定名称的属性，该方法将创建一个新属性。

2.类似于getAttribute()方法，setAttribute()方法只能通过元素节点对象调用的函数。

任务

**试一试，使用getAttribute()和setAttribute()方法，完成下面的任务:**

在第21行补充代码，使用getAttribute()方法获取元素属性值，保存在变量text。

在第25行补充代码，使用setAttribute()方法设置title属性值。

**?**不会了怎么办

1. var text=Lists[i].getAttribute("title");

2. Lists[i].setAttribute("title","WEB前端技术");

代码：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<p id="intro">我的课程</p>

<ul>

<li title="JS">JavaScript</li>

<li title="JQ">JQuery</li>

<li title="">HTML/CSS</li>

<li title="JAVA">JAVA</li>

<li title="">PHP</li>

</ul>

<h1>以下为li列表title的值,当title为空时，新设置值为"WEB前端技术":</h1>

<script type="text/javascript">

var Lists=document.getElementsByTagName("li");

for (var i=0; i<Lists.length;i++)

{

var text=Lists[i].getAttribute("title");

document.write(text +"<br>");

if(text=="")

{

Lists[i].setAttribute("title","WEB前端技术");

document.write(Lists[i].getAttribute("title")+"<br>");

}

}

</script>

</body>

</html>

### 9-7节点属性

在文档对象模型 (DOM) 中，每个节点都是一个对象。DOM 节点有三个重要的属性 ：

1. nodeName : 节点的名称

2. nodeValue ：节点的值

3. nodeType ：节点的类型

**一、nodeName 属性:**节点的名称，是只读的。

1. 元素节点的 nodeName 与标签名相同  
2. 属性节点的 nodeName 是属性的名称  
3. 文本节点的 nodeName 永远是 #text  
4. 文档节点的 nodeName 永远是 #document

**二、nodeValue 属性：**节点的值

1. 元素节点的 nodeValue 是 undefined 或 null  
2. 文本节点的 nodeValue 是文本自身  
3. 属性节点的 nodeValue 是属性的值

**三、nodeType 属性:**节点的类型，是只读的。以下常用的几种结点类型:

**元素类型    节点类型**  
  元素          1  
  属性          2  
  文本          3  
  注释          8  
  文档          9

任务

试一试，在<script>的标签内容，获取所有LI标签，并输出相应节点的名称、节点的值、节点的类型。

**?**不会了怎么办

1. 获取li标签使用getElementsByTagName()。

2. 输出所有li元素节点的属性，使用for和 nodeName 、 nodeValue 、nodeType属性。

代码：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>节点属性</title>

</head>

<body>

<ul>

<li>javascript</li>

<li>HTML/CSS</li>

<li>jQuery</li>

</ul>

<script type="text/javascript">

var list=document.getElementsByTagName("li");

for(var i=0; i<list.length; i++)

{

document.write(list[i].nodeName+"<br>");

document.write(list[i].nodeValue+"<br>");

document.write(list[i].nodeType+"<br>");

}

</script>

</body>

</html>

### 9-8访问子结点childNodes

访问选定元素节点下的所有子节点的列表，返回的值可以看作是一个数组，他具有length属性。

**语法：**

elementNode.childNodes

**注意：**

如果选定的节点没有子节点，则该属性返回不包含节点的 NodeList。

**我们来看看下面的代码:**

[](http://img.mukewang.com/538405fa00010e6c05630357.jpg)

**运行结果:**

**IE:**

UL子节点个数:3

节点类型:1

**其它浏览器:**

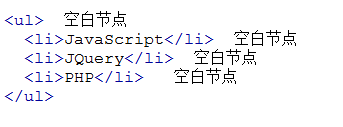
UL子节点个数:7

节点类型:3

**注意:**

1. IE全系列、firefox、chrome、opera、safari兼容问题

2. 节点之间的空白符，在firefox、chrome、opera、safari浏览器是文本节点，所以IE是3，其它浏览器是7，如下图所示:

[](http://img.mukewang.com/538d2b8a000163e303430127.jpg)

**如果把代码改成这样:**

<ul><li>javascript</li><li>jQuery</li><li>PHP</li></ul>

**运行结果:（IE和其它浏览器结果是一样的）**

UL子节点个数:3

节点类型:1

### 任务

**试一试，在script标签内，获取子节点，并输出相应属性。**

1. 获取第一个DIV的子节点。

2. 使用for遍历每个节点。

3. 输出相应节点的属性。

注意: 输出内容可能和想的内容不同，想想为什么。

**?**不会了怎么办

1. 使用getElementsByTagName与childNodes

2. 因为获取的孩子节点是一个数组，所以使用for循环。

3. 输出节点属性，nodeName、nodeType、nodeValue。

var x=document.getElementsByTagName("div")[0].childNodes;

for(i=0;i<x.length;i++){

document.write("节点类型:"+x[i].nodeName+x[i].nodeType+x[i].nodeValue+"<br>");}

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<div>

javascript

<p>javascript</p>

<div>jQuery</div>

<h5>PHP</h5>

</div>

<script type="text/javascript">

var x=document.getElementsByTagName("div")[0].childNodes;

for(var i=0; i<x.length; i++){

document.write("节点类型:"+x[i].nodeName+" "+x[i].nodeType+" "+x[i].nodeValue+"<br />");}

</script>

</body>

</html>

### 9-9访问子结点的第一和最后项

一、firstChild 属性返回‘childNodes’数组的第一个子节点。如果选定的节点没有子节点，则该属性返回 NULL。

**语法：**

node.firstChild

**说明：**与elementNode.childNodes[0]是同样的效果。

二、 lastChild 属性返回‘childNodes’数组的最后一个子节点。如果选定的节点没有子节点，则该属性返回 NULL。

**语法：**

node.lastChild

**说明：**与elementNode.childNodes[elementNode.childNodes.length-1]是同样的效果。

**注意:**上一节中，我们知道Internet Explorer 会忽略节点之间生成的空白文本节点，而其它浏览器不会。我们可以通过检测节点类型，过滤子节点。 (以后章节讲解)

### 任务

**试一试，找到指定元素(div)的第一个和最后一个子节点。**

1. 第15行输出指定元素(div)的第一个子节点。

2. 第16行输出指定元素(div)的最后一个子节点。

3. 也许结果和你想象的不一样，试着调整代码，输出正确代码。

**?**不会了怎么办

1. document.write(x.firstChild.nodeName+"<br>");

2. document.write(x.lastChild.nodeName+"<br>");

3. 将 HTML 调整成 <div><p>javascript</p><div>jQuery</div><h5>PHP</h5></div>

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<div id="con"><p>javascript</p><div>jQuery</div><h5>PHP</h5></div>

<script type="text/javascript">

var x=document.getElementById("con");

document.write(x.firstChild.nodeName+"<br />");

document.write(x.lastChild.nodeName+"<br ");

</script>

</body>

</html>

### 9-10访问父节点parentNode

获取指定节点的父节点

**语法：**

elementNode.parentNode

**注意:父节点只能有一个。**

看看下面的例子,获取 P 节点的父节点，代码如下:

<div id="text">

<p id="con"> parentNode 获取指点节点的父节点</p>

</div>

<script type="text/javascript">

  var mynode= document.getElementById("con");

  document.write(mynode.**parentNode**.nodeName);

</script>

运行结果:

parentNode 获取指点节点的父节点

DIV

**访问祖节点:**

elementNode.parentNode.parentNode

看看下面的代码:

<div id="text">

  <p>

    parentNode

    <div id="con"> 获取指点节点的父节点</div>

  </p>

</div>

<script type="text/javascript">

  var mynode= document.getElementById("con");

  document.write(mynode.parentNode.nodeName);

</script>

运行结果:

parentNode

获取指点节点的父节点

DIV

注意: 浏览器兼容问题，chrome、firefox等浏览器标签之间的空白也算是一个文本节点。

### 任务

**试一试，通过获取的mylist节点，使用访问父节点parentNode，将"HTML/CSS"课程内容输出。**

补充第30行代码，将"HTML/CSS"课程内容输出。

**?**不会了怎么办

document.write(mylist.parentNode.parentNode.parentNode.lastChild.innerHTML);

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<ul id="con">

<li id="lesson1">javascript

<ul>

<li id="tcon"> 基础语法</li>

<li>流程控制语句</li>

<li>函数</li>

<li>事件</li>

<li>DOM</li>

</ul>

</li>

<li id="lesson2">das</li>

<li id="lesson3">dadf</li>

<li id="lesson4">HTML/CSS

<ul>

<li>文字</li>

<li>段落</li>

<li>表单</li>

<li>表格</li>

</ul>

</li></ul>

<script type="text/javascript">

var mylist = document.getElementById("tcon");

document.write(mylist.parentNode.parentNode.parentNode.lastChild.innerHTML);

</script>

</body>

</html>

**注：**浏览器兼容问题，chrome、firefox等浏览器标签之间的空白也算是一个文本节点。这句话在这几节内容里，对程序的理解有很重要的作用

### 9-11访问兄弟节点

1. nextSibling 属性可返回某个节点之后紧跟的节点（处于同一树层级中）。

**语法：**

nodeObject.nextSibling

**说明：**如果无此节点，则该属性返回 null。

2. previousSibling 属性可返回某个节点之前紧跟的节点（处于同一树层级中）。

**语法：**

nodeObject.previousSibling

**说明：**如果无此节点，则该属性返回 null。

注意: 两个属性获取的是节点。Internet Explorer 会忽略节点间生成的空白文本节点（例如，换行符号），而其它浏览器不会忽略。

**解决问题方法:**

判断节点nodeType是否为1, 如是为元素节点，跳过。

[](http://img.mukewang.com/5386e3c80001c25607010856.jpg)

**运行结果:**

LI = javascript

nextsibling: LI = jquery

任务

试一试，编写get\_previousSibling 函数，获得指定节点的上一个节点。

**?**不会了怎么办

function get\_previousSibling(n)  
{  
var x=n.previousSibling;  
while (x.nodeType!=1)  
  {  
  x=x.previousSibling;  
  }  
return x;  
}

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>nextSibling</title>

</head>

<body>

<ul id="u1">

<li id="a">javascript</li>

<li id="b">jquery</li>

<li id="c">html</li>

</ul>

<ul id="u2">

<li id="d">css3</li>

<li id="e">php</li>

<li id="f">java</li>

</ul>

<script type="text/javascript">

function get\_nextSibling(n)

{

var x=n.nextSibling;

while (x.nodeType!=1)

{

x=x.nextSibling;

}

return x;

}

function get\_preSibling(n)

{

var x=n.previousSibling;

while(x.nodeType!=1){

x=x.previousSibling;

}

return x;

}

var x=document.getElementsByTagName("li")[0];

document.write(x.nodeName);

document.write(" = ");

document.write(x.innerHTML);

var y=get\_nextSibling(x);

document.write("<br />nextsibling: ");

document.write(y.nodeName);

document.write(" = ");

document.write(y.innerHTML+"<br/><br/>");

var x=document.getElementsByTagName("li")[2];

document.write(x.nodeName);

document.write(" = ");

document.write(x.innerHTML);

document.write("<br />previousSibling: ");

document.write(y.nodeName);

document.write(" = ");

document.write(y.innerHTML);

</script>

</body>

</html>

### 9-12创建元素节点createElement

createElement()方法可创建元素节点。此方法可返回一个 Element 对象。

**语法：**

document.createElement(tagName)

**参数:**

tagName：字符串值，这个字符串用来指明创建元素的类型。

**注意：**要与appendChild() 或 insertBefore()方法联合使用，将元素显示在页面中。

我们来创建一个按钮，代码如下：

<script type="text/javascript">

var body = document.body;

var input = document.createElement("input");

input.type = "button";

input.value = "创建一个按钮";

body.appendChild(input);

</script>

效果：在HTML文档中，创建一个按钮。

我们也可以使用setAttribute来设置属性，代码如下：

<script type="text/javascript">

var body= document.body;

var btn = document.createElement("input");

btn.setAttribute("type", "text");

btn.setAttribute("name", "q");

btn.setAttribute("value", "使用setAttribute");

btn.setAttribute("onclick", "javascript:alert('This is a text!');");

body.appendChild(btn);

</script>

效果：在HTML文档中，创建一个文本框，使用setAttribute设置属性值。 当点击这个文本框时，会弹出对话框“This is a text!”。

任务

试一试，实现在HTML文档中创建一个链接，并设置相应属性。

1. 在右边编辑器补充代码，完善createa(url,text)创建链接函数，参数1为链接地址，参数2为链接文本。函数中添加链接地址、文本、文字颜色属性。

2. 调用createa函数，链接地址 http://www.imooc.com，文本为:慕课网

**?**不会了怎么办

var body= document.body;  
//创建链接  
function createa(url,text)  
{  
    var a = document.createElement("a");  
    a.href = url;  
    a.innerHTML = text;  
    a.style.color = "red";  
    body.appendChild(a);  
}  
// 调用函数创建链接  
createa("http://www.imooc.com/","慕课网");

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<script type="text/javascript">

var main = document.body;

//创建链接

function createa(url,text)

{

var a=document.createElement('a');

a.innerHTML=text;//注意innerHTML的转换

a.setAttribute("href",url);

a.setAttribute("style","color:red;font-size:24px;");

main.appendChild(a);

}

// 调用函数创建链接

createa("http://www.imooc.com","慕课网");

</script>

</body>

</html>

### 9-13创建文本节点createTextNode

createTextNode() 方法创建新的文本节点，返回新创建的 Text 节点。

**语法：**

document.createTextNode(data)

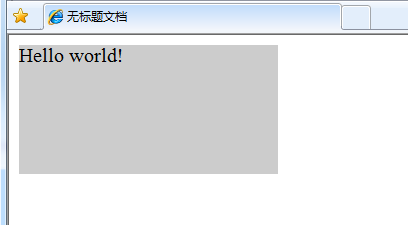
**参数：**

data : 字符串值，可规定此节点的文本。

我们来创建一个<div>元素并向其中添加一条消息，代码如下：

[](http://img.mukewang.com/53951c200001d32d07130554.jpg)

**运行结果:**

[](http://img.mukewang.com/53951c720001996d04080225.jpg)

任务

在代码编辑器<script>标签中，创建一个P标签，设置className属性，使用createTextNode创建文本节点"I love JavaScript!"。

**?**不会了怎么办

<script type="text/javascript">

   var element = document.createElement("p");  
   element.className = "message";  
   var textNode = document.createTextNode("I love JavaScript!");  
   element.appendChild(textNode);  
   document.body.appendChild(element);  
          
</script>

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

<style type="text/css">

.message{

width:200px;

height:100px;

background-color:#CCC;}

</style>

</head>

<body>

<script type="text/javascript">

var p=document.createElement("p");

p.className="message";

var textNode=document.createTextNode("I love javescript!");

p.appendChild(textNode);

document.body.appendChild(p);

</script>

</body>

</html>

### 9-14插入节点appendChild()

在指定节点的最后一个子节点列表之后添加一个新的子节点。

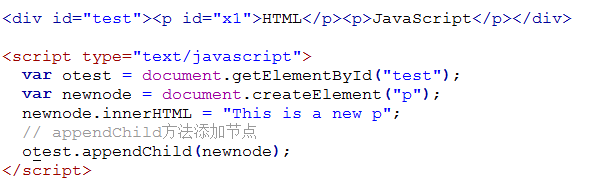
**语法:**

appendChild(newnode)

**参数:**

newnode：指定追加的节点。

我们来看看，div标签内创建一个新的 P 标签，代码如下:

[](http://img.mukewang.com/5398fd020001ad4905890193.jpg)

**运行结果:**

HTML

JavaScript

This is a new p

任务

**试一试，完成为UL标签添加一个新项PHP。**

1.在代码编辑器中，script标签内，为ul添加一个li。

2.设置li内容为PHP。

**?**不会了怎么办

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<ul id="test">

<li>JavaScript</li>

<li>HTML</li>

</ul>

<script type="text/javascript">

var otest = document.getElementById("test");

var newnode = document.createElement("li");

var newtext= document.createTextNode("PHP");

newnode.appendChild(newtext);

otest.appendChild(newnode);

</script>

</body>

</html>

### 9-15插入节点insertBefore()

insertBefore() 方法可在已有的子节点前插入一个新的子节点。

**语法:**

insertBefore(newnode,node);

**参数:**

newnode:要插入的新节点。

node:可选，指定此节点前插入节点。

我们来看看下面代码，和 appendChild()效果一样。

[](http://img.mukewang.com/539531db0001f10206870406.jpg)

**运行结果:**

JavaScript

HTML

This is a new p

我们在来看看下面代码，在指定节点前插入节点。

[](http://img.mukewang.com/5395318100010c6806960431.jpg)

**运行结果:**

This is a new p

JavaScript

HTML

**注意:** otest.insertBefore(newnode,node); 也可以改为:  otest.insertBefore(newnode,otest.childNodes[0]);

任务

试一试，在script 标签内补充代码，实现创建一个新li标签，内容为"php",并将新创建的li插入到内容为HTML的标签前。

**?**不会了怎么办

<body>

<ul id="test"><li>JavaScript</li><li>HTML</li></ul>

<script type="text/javascript">

  var otest = document.getElementById("test");    
  var newnode = document.createElement("li");  
  newnode.innerHTML = "PHP" ;    
 otest.insertBefore(newnode,otest.childNodes[1]);   
            
</script>

</body>

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<ul id="test"><li>JavaScript</li><li>HTML</li></ul>

<script type="text/javascript">

var otest = document.getElementById("test");

var newnode = document.createElement("li");

newnode.innerHTML = "PHP";

otest.insertBefore(newnode,otest.childNodes[1]);

</script>

</body>

</html>

### 9-16删除节点removeChild()

removeChild() 方法从子节点列表中删除某个节点。如删除成功，此方法可返回被删除的节点，如失败，则返回 NULL。

**语法:**

nodeObject.removeChild(node)

**参数:**

node ：必需，指定需要删除的节点。

我们来看看下面代码，删除子点。

[](http://img.mukewang.com/5399744d000153a306060342.jpg)

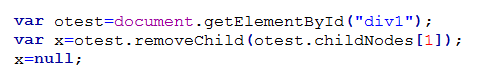
**运行结果:**

HTML

删除节点的内容: javascript

**注意:** 把删除的子节点赋值给 x，这个子节点不在DOM树中，但是还存在内存中，可通过 x 操作。

如果要完全删除对象，给 x 赋 null 值，代码如下:

[](http://img.mukewang.com/539975a800017c8e04790082.jpg)

### 任务

试一试，定义clearText()函数，完成节点内容的删除。

1. 删除该节点的内容，先要获取子节点。

2. 然后使用 for 遍历每个子节点。

3. 使用removeChild()删除节点。

**?**不会了怎么办

function clearText() {  
  var content=document.getElementById("content");  
  for(var i=content.childNodes.length-1;i>=0;i--){  
     var childNode = content.childNodes[i];  
     content.removeChild(childNode);  
  }  
}

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<div id="content">

<h1>html</h1>

<h1>php</h1>

<h1>javascript</h1>

<h1>jquery</h1>

<h1>java</h1>

</div>

<script type="text/javascript">

function clearText() {

var content=document.getElementById("content");

// 在此完成该函数

var j=content.childNodes.length;

for(var i=0;i<j;i++){

var x=content.removeChild(content.childNodes[0]);

}

}

</script>

<button onclick="clearText()">清除节点内容</button>

</body>

</html>

### 9-17替换元素节点replaceChild()

replaceChild 实现子节点(对象)的替换。返回被替换对象的引用。

**语法：**

node.replaceChild (newnode,oldnew )

**参数：**

newnode : 必需，用于替换 oldnew 的对象。   
oldnew : 必需，被 newnode 替换的对象。

我们来看看下面的代码:

[](http://img.mukewang.com/539557d70001c3ee07190429.jpg)

效果: 将文档中的 Java 改为 JavaScript。

**注意:**

1. 当 oldnode 被替换时，所有与之相关的属性内容都将被移除。

2. newnode 必须先被建立。

### 任务

试一试，补充函数 replaceMessage() 代码，实现将 b 标签替换成 i 标签。

**?**不会了怎么办

function replaceMessage(){  
           var oldnode=document.getElementById("oldnode");  
           var oldHTML= oldnode.innerHTML;             
           var newnode=document.createElement("i");           
           oldnode.parentNode.replaceChild(newnode,oldnode);  
           newnode.innerHTML=oldHTML;  
             
}

代码：<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>无标题文档</title>

</head>

<body>

<div><b id="oldnode">JavaScript</b>是一个很常用的技术，为网页添加动态效果。</div>

<a href="javascript:replaceMessage()"> 将加粗改为斜体</a>

<script type="text/javascript">

function replaceMessage(){

var oldnode = document.getElementById('oldnode');

var oldHTML = oldnode.innerHTML;

var newnode = document.createElement('i');

oldnode.parentNode.replaceChild(newnode,oldnode);

newnode.innerHTML=oldHTML;

}

</script>

</body>

</html>

### 9-18浏览器窗口可视区域大小

获得浏览器窗口的尺寸（浏览器的视口，不包括工具栏和滚动条）的方法:

一、对于IE9+、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

二、对于 Internet Explorer 8、7、6、5：

•  document.documentElement.clientHeight表示HTML文档所在窗口的当前高度。

•  document.documentElement.clientWidth表示HTML文档所在窗口的当前宽度。

或者

Document对象的body属性对应HTML文档的<body>标签

•  document.body.clientHeight

•  document.body.clientWidth

在不同浏览器都实用的 JavaScript 方案：

var w= document.documentElement.clientWidth

|| document.body.clientWidth;

var h= document.documentElement.clientHeight

|| document.body.clientHeight;

### 任务

在script标签内，补充右边编辑器代码，获取浏览器当前窗口大小。

**?**不会了怎么办

**浏览器当前窗口大小的方法，要注意浏览器兼容问题。**

代码：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

</head>

<body>

<script type="text/javascript">

var h=document.documentElement.clientHeight||document.body.clientHeight;

var w=document.documentElement.clientWidth||document.body.clientWidth;

document.write("浏览器的高度："+h+"px,宽度："+w+"px");

</script>

</body>

</html>

### 9-19网页尺寸scrollHeight

scrollHeight和scrollWidth，获取网页内容高度和宽度(不包括滚动条)。

**一、针对IE、Opera:**

scrollHeight 是网页内容实际高度，可以小于 clientHeight。

**二、针对NS、FF:**

scrollHeight 是网页内容高度，不过最小值是 clientHeight。也就是说网页内容实际高度小于 clientHeight 时，scrollHeight 返回 clientHeight 。

**三、浏览器兼容性**

var w=document.documentElement.scrollWidth

|| document.body.scrollWidth;

var h=document.documentElement.scrollHeight

|| document.body.scrollHeight;

**注意:区分大小写**

scrollHeight和scrollWidth还可获取Dom元素中内容实际占用的高度和宽度。

### 任务

**在script标签内，补充右边编辑器代码，获取网页的实际宽度和高度。**

**?**不会了怎么办

**使用scrollHeight、scrollWidth获取网页的大小时，要注意浏览器兼容问题。**

**代码：**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

</head>

<body>

<script type="text/javascript">

var w=document.body.scrollWidth || document.documentElement.scrollWidth;

var h=document.body.scrollHeight || document.documentElement.scrollHeight;

document.write(w+"px"+"<br />"+h+"px");

</script>

</body>

</html>

### 9-20网页尺寸offsetHeight

offsetHeight和offsetWidth，获取网页内容高度和宽度(包括滚动条等边线，会随窗口的显示大小改变)。

**一、值**

offsetHeight = clientHeight + 滚动条 + 边框。

**二、浏览器兼容性**

var w= document.documentElement.offsetWidth

|| document.body.offsetWidth;

var h= document.documentElement.offsetHeight

|| document.body.offsetHeight;

### 任务

在script标签内，补充右边编辑器代码，获取网页的宽度和高度,并输出。

**?**不会了怎么办

使用offsetHeight、offsetWidth获取网页的大小。

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

</head>

<body>

<script type="text/javascript">

var w= document.documentElement.offsetWidth

|| document.body.offsetWidth;

var h= document.documentElement.offsetHeight

|| document.body.offsetHeight;

document.write("宽:"+w+"px<br />"+"高:"+h+"px");

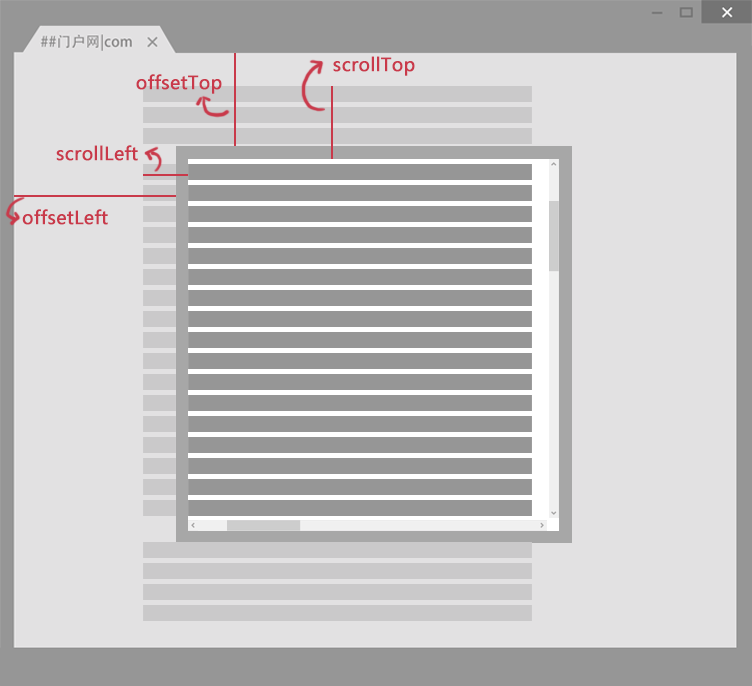
</script>

</body>

</html>

### 9-21网页卷去的距离

**我们先来看看下面的图：**

[](http://img.mukewang.com/5347b2b10001e1a307520686.jpg)

**scrollLeft:**设置或获取位于给定对象左边界与窗口中目前可见内容的最左端之间的距离 ，即左边灰色的内容。

**scrollTop:**设置或获取位于对象最顶端与窗口中可见内容的最顶端之间的距离 ，即上边灰色的内容。

**offsetLeft:**获取指定对象相对于版面或由 offsetParent 属性指定的父坐标的计算左侧位置 。

**offsetTop:**获取指定对象相对于版面或由 offsetTop 属性指定的父坐标的计算顶端位置 。

**注意:区分大小写**

### 任务

在右边编辑器中，调整横竖滚动条后，点击按钮后，查看offsetTop、offsetLeft、scrollTop、scrollLeft值的变化。

**?**不会了怎么办

注意这四个方法的区别，请仔细学习知识讲解内容。

**代码：**

<!DOCTYPE HTML>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<script type="text/javascript">

function req(){

var div = document.getElementById("div1");

document.getElementById("li1").innerHTML = (div.offsetTop)+"px";//div1距离屏幕顶部的距离

document.getElementById("li2").innerHTML = (div.offsetLeft)+"px";//div1距离屏幕左部的距离

document.getElementById("li3").innerHTML = (div.scrollTop)+"px";//div1纵向滚动条滚动的距离

document.getElementById("li4").innerHTML = (div.scrollLeft)+"px";//div1横向滚动条滚动的距离

}

</script>

</head>

<body style="border:10px solid #ccc;padding:0px 0px;margin:5px 10px">

<div style="width:60%;border-right:1px dashed red;float:left;">

<div style="float:left;">

<div id="div1" style="border:5px red solid;height:300px;width:200px;overflow:auto">

<div style="height:500px;width:400px">请调整横竖滚动条后，点击按钮后查看offsetTop、offsetLeft、scrollTop、scrollLeft值。</div>

</div>

<input type="button" value="点击我!" onclick="req()" style="border: 1px solid purple;height: 25px;"/>

</div>

</div>

<div style="width:30%;float:left;">

<ul style="list-style-type: none; line-height:30px;">结果：

<li>offsetTop : <span id="li1"></span></li>

<li>offsetLeft : <span id="li2"></span></li>

<li> scrollTop : <span id="li3"></span></li>

<li>scrollLeft : <span id="li4"></span></li>

</ul>

</div>

<div style="clear:both;"></div>

</body>

</html>

### 9-22编程练习

制作一个表格，显示班级的学生信息。

**要求：**

1. 鼠标移到不同行上时背景色改为色值为 #f2f2f2，移开鼠标时则恢复为原背景色 #fff

2. 点击添加按钮，能动态在最后添加一行

3. 点击删除按钮，则删除当前行

### 任务

第一步： 首先，我们创建删除函数，并在删除按钮上添加点击事件；

提示: 使用removeChild()。

第二步： 编写一个函数，供添加按钮调用，动态在表格的最后一行添加子节点；

提示: 使用createElement()、innerHTML、appendChild()。

第三步： 更改鼠标移动改变背景则可以通过给每行绑定鼠标移上事件和鼠标移除事件来改变所在行背景色。

提示:

1. 获取表格的行，getElementsByTagName 。

2. 使用for进行循环，为每行添加事件及背景颜色设置。

**?**不会了怎么办

**代码：**

<!DOCTYPE html>

<html>

<head>

<title> new document </title>

<meta http-equiv="Content-Type" content="text/html; charset=gbk"/>

<script type="text/javascript">

window.onload = function(){

var tbody = document.getElementById('table').lastChild;

trs = tbody.getElementsByTagName('tr');

for(var i =1;i<trs.length;i++){

trs[i].onmouseover = function(){

this.style.backgroundColor ="#f2f2f2";

}

trs[i].onmouseout = function(){

this.style.backgroundColor ="#fff";

}

}

}

function addOne(obj){

var tbody = document.getElementById('table').lastChild;

var tr = document.createElement('tr');

var td = document.createElement("td");

td.innerHTML = "<input type='text'/>";

tr.appendChild(td);

td = document.createElement("td");

td.innerHTML = "<input type='text'/>";

tr.appendChild(td);

td = document.createElement("td");

td.innerHTML = "<a href='javascript:;' onclick='deleteRow(this)'>删除</a>";

tr.appendChild(td);

tbody.appendChild(tr);

}

function deleteRow(obj){

var tbody = document.getElementById('table').lastChild;

var tr = obj.parentNode.parentNode;

tbody.removeChild(tr);

}

</script>

</head>

<body>

<table border="1" width="50%" id="table">

<tr>

<th>学号</th>

<th>姓名</th>

<th>操作</th>

</tr>

<tr>

<td>xh001</td>

<td>王小明</td>

<td><a href="javascript:;" onclick="deleteRow(this)">删除</a></td>

</tr>

<tr>

<td>xh002</td>

<td>刘小芳</td>

<td><a href="javascript:;" onclick="deleteRow(this)">删除</a></td>

</tr>

</table>

<input type="button" value="添加一行" onclick="addOne()" />

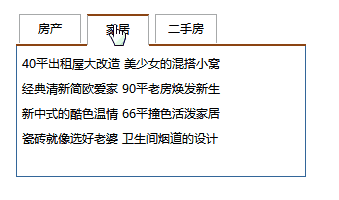
</body>

</html>

### 10-1编程挑战

现在利用之前我们学过的JavaScript知识，实现选项卡切换的效果。

**效果图:**



**文字素材:**

房产：

    275万购昌平邻铁三居 总价20万买一居  
    200万内购五环三居 140万安家东三环  
    北京首现零首付楼盘 53万购东5环50平  
    京楼盘直降5000 中信府 公园楼王现房

家居:

     40平出租屋大改造 美少女的混搭小窝  
     经典清新简欧爱家 90平老房焕发新生  
     新中式的酷色温情 66平撞色活泼家居  
     瓷砖就像选好老婆 卫生间烟道的设计

二手房：

     通州豪华3居260万 二环稀缺2居250w甩  
     西3环通透2居290万 130万2居限量抢购  
     黄城根小学学区仅260万 121平70万抛!  
     独家别墅280万 苏州桥2居优惠价248万  
 

### 任务

大家先思考和分析实现思路，然后在动手实现

一、HTML页面布局

提示:

选项卡标题使用ul..li

选项卡内容使用div

二、CSS样式制作

提示:

整个选项卡的样式设置

选项卡标题的样式设置

选项卡内容的样式设置

一开始只显示一个选项卡内容，其它选项卡内容隐藏。

三、JS实现选项卡切换

提示:

获取选项卡标题和选项卡内容

选项卡内容多个，需要循环遍历来操作，得知哪个选项卡和哪个选项内容匹配

通过改变DOM的css类名称,当前点击的选项卡显示，其它隐藏。

**?**不会了怎么办

**代码：**

<!DOCTYPE html>

<html>

<head lang="en">

<meta charset="UTF-8">

<title>实践题 - 选项卡</title>

<style type="text/css">

\*{padding:0px;margin: 0px;font:12px normal "microsoft yahei";}

#tabs {width:290px;padding:5px;height:150px;margin:20px;}

#tabs ul{list-style:none;display: block;height:30px;line-height:30px;border-bottom:2px saddlebrown solid;}

#tabs ul li{background:#fff;cursor:pointer;float:left;list-style:none;height:28px;line-height:28px;margin:0px 3px;border:1px solid #aaaaaa;border-bottom:none;display:inline-block;width:60px;text-align: center;}

#tabs ul li.on{border-top:2px solid saddlebrown;border-bottom: 2px solid #fff;}

#tabs div{height:120px;line-height: 25px;border:1px solid #336699;border-top:none;padding:5px;}

.hide{display: none;}

</style>

<script type="text/javascript">

window.onload = function(){

var oTab = document.getElementById("tabs");

var oUl = oTab.getElementsByTagName("ul")[0];

var oLis = oUl.getElementsByTagName("li");

var oDivs= oTab.getElementsByTagName("div");

for(var i= 0,len = oLis.length;i<len;i++){

oLis[i].index = i;

oLis[i].onclick = function() {

for(var n= 0;n<len;n++){

oLis[n].className = "";

oDivs[n].className = "hide";

}

this.className = "on";

oDivs[this.index].className = "";

}

};

}

</script>

</head>

<body>

<div id="tabs">

<ul>

<li class="on">房产</li>

<li>家居</li>

<li>二手房</li>

</ul>

<div>

275万购昌平邻铁三居 总价20万买一居<br>

200万内购五环三居 140万安家东三环<br>

北京首现零首付楼盘 53万购东5环50平<br>

京楼盘直降5000 中信府 公园楼王现房<br>

</div>

<div class="hide">

40平出租屋大改造 美少女的混搭小窝<br>

经典清新简欧爱家 90平老房焕发新生<br>

新中式的酷色温情 66平撞色活泼家居<br>

瓷砖就像选好老婆 卫生间烟道的设计<br>

</div>

<div class="hide">

通州豪华3居260万 二环稀缺2居250w甩<br>

西3环通透2居290万 130万2居限量抢购<br>

黄城根小学学区仅260万 121平70万抛!<br>

独家别墅280万 苏州桥2居优惠价248万<br>

</div>

</div>

</body>

</html>

## 要点：

### JS 中“===”与 “==”比较的区别

1. ==用于一般比较，==在比较的时候可以转换数据类型。

2. ===用于严格比较，===严格比较，只要类型不匹配就返回flase。

**看一个简单的例子:**

给定 x=5

== 等于 x==8 为 false

=== 全等（值和类型） x===5 为 true；x==="5" 为 false

**举例说明：**

"1" == true

类型不同，"=="将先做类型转换，把true转换为1，即为 "1" == 1；

此时，类型仍不同，继续进行类型转换，把"1"转换为1，即为 1 == 1；

此时，"==" 左右两边的类型都为数值型，比较成功！

如果比较："1" === true 左侧为字符型，右侧为bool布尔型，左右两侧类型不同，结果为false；

如果比较："1" === 1 左侧为字符型，右侧为int数值型，左右两侧类型不同，结果为false；

如果比较：1 === 1 左侧为int数值型，右侧为int数值型，左右两侧类型相同，数值大小也相同，结果为true；

如果比较：1 === 2 左侧为int数值型，右侧为int数值型，左右两侧类型相同，但数值大小不同，结果为false；

**简而言之就是 "==" 只要求值相等; "===" 要求值和类型都相等**

### ++a与a++的区别

a++是运算后自增，而++a是先自增在运算（即++a 先自加再取值，a++先取值在自加)

**相同点：**

a++和++a中， i的值自增1，也就是a=a+1;

**不同点：**

a++ ： 把（a++）作为一个整体表达式，a 的值虽然自增1，但是整个表达式的值是取 a自增  之前  的值；

++a : 也把（++a）作为一个整体表达式，a 的值也自增1，但是整个表达式的值是取 a 自增  之后  的值。

**看实例:**

m=a++ 相当于 m=a,a=a+1

m=++a 相当于 a=a+1,m=a 这种自加运算和自减运算,都相当于两步骤

**总结:** a++ 是先把 a 赋值,然后自己加一, ++a,是先自己加一,然后再把 a 赋值。

### JS中substr和substring的用法和区别

substr 和 substring都是JS 截取字符串函数,两者用法很相近,下面是两者的语法很示例:

**一、substr 方法**

返回一个从指定位置开始的指定长度的子字符串。  
string.substr(start [, length ])

**注意:** length可选项。如 length 为 0 或负数，将返回一个空字符串。如果没有指定该参数，则子字符串到 string 的最后。

**二、substring 方法**

返回位于 String 对象中指定位置的子字符串。  
string.substring(start, end)

**注意:**

substring 方法将返回一个包含从 start 到最后（不包含 end ）的子字符串的字符串。

**三、示例代码**

var str = "I love JS!";// 有一个str字符串，如想获取JS子字符串，用两种方法如何实现。  
str.substr(7, 2); // 获取子字符串。

str.substring(7, 9); // 获取子字符串。

**结果:  JS**

**区别:第二参数，substr第二个参数是获取子字符串的长度，substring第二个参数是获取子字符串的结束位置。**

**四、注意事项**

substr和substring两个函数截取带有空格的字符串后的长度是每个空格算一个字符长度。例如：

    var a = "I am imooc!";

    a.substring(0, 5).length的值是5，而不是4，但alert(a.substring(0, 5));的值却是I am ，这样在做alert("I am" == a.substring(0, 5));的时候就是false了，alert("I am" == a.substring(0, 4));才是true。