Javascript的基本组成是什么

ECMAscript + Bom +Dom

ECMAscript 就是一些语法，变量循环结构

Bom 浏览器对象模型

Dom 文档对象模型 （树状结构）

1如何添加就是脚本

<script type=”text/javascript”> </script>

2并不是完全的写在html中，可以将.js文件单独的拿出放在一个文件夹中

<script src=”script.js”> </script>

如进行页面显示初始化的js必须放在head里面，因为初始化都要求提前进行（如给页面body设置css等）；而如果是通过事件调用执行的function那么对位置没什么要求的。

**认识语句和字符**

**注意**:

1. “;”分号要在英文状态下输入，同样，JS中的代码和符号都要在英文状态下输入。

2. 虽然分号“;”也可以不写，但我们要养成编程的好习惯，记得在语句末尾写上分号

注释：单行注释，在注释内容前加符号 “//” 多行注释以"/\*"开始，以"\*/"结束。

**什么是变量**：

一般将盒子box’看做是变量 定义变量是 var 变量

**变量名可以任意取名，但要遵循命名规则**:

1.变量必须使用字母、下划线(\_)或者美元符($)开始。

2.然后可以使用任意多个英文字母、数字、下划线(\_)或者美元符($)组成。3.不能使用JavaScript关键词与JavaScript保留字。

变量必须先定义再赋值 一个变量可以重复赋值（js中是区分大小写的 也可以直接赋值但是是不规范的）

**判断语句**

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 }

**函数**

function 函数名()

{

函数代码;

}

注释：1. function定义函数的关键字。

2. "函数名"你为函数取的名字。

3. "函数代码"替换为完成特定功能的代码。

有参数的函数返回值

<script type="text/javascript">

function myfunction(txt)

{

alert(txt)

}

</script>

在head中<input type="button"

onclick="myfunction('Good Morning!')"

value="In the Morning"> ///直接通过函数名和参数传进来即可

**注意：** 对要返回的值用return来直接从函数中返回

**function prod(a,b)**

**{**

**x=a\*b**

**return x**

**}**

循环 还是用for

document.write("<h" + i + ">This is header " + i)

document.write("</h" + i + ">") 输出不同的标题内容注意它的大小随着变量的变化而变化

break 命令可以终止循环的运行，然后继续执行循环之后的代码（如果循环之后有代码的话）。

continue 命令会终止当前的循环，然后从**下一个值**继续运行。

**For ..in 用来数组的遍历**

var x

var mycars = new Array()

mycars[0] = "Saab"

mycars[1] = "Volvo"

mycars[2] = "BMW"

for (x in mycars) 语法 {

{ for（变量 in 对象）{执行语句}

document.write(mycars[x] + "<br />")

}

运算符 比如说 +

Text 1=what a

Text 2 =nice day text1 + text2 = what a nce day

**js对象**

字符串对象

**1长度 .length**

var txt="Hello World!"

document.write(txt.length

**2字符串添加样式**

document.write("<p>Big: " + txt.big() + "</p>")

document.write("<p>Small: " + txt.small() + "</p>")

**3index（）方法** ：如何使用 indexOf() 来定位字符串中某一个指定的字符首次出现的位置

var str="Hello world!"

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

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

document.write(str.indexOf("world"))

**4match（）方法** 使用 match() 来查找字符串中特定的字符，并且如果找到的话，则返回这个字符 （如果不存在所查的字符就返回null）

**5replace 替换**

var str="Visit Microsoft!"

document.write(str.replace(/Microsoft/,"W3Schools"))

其他的一些详见手册

**日期**

**1返回当前的日期 date（**）

document.write(Date()

**2 getTime（）**为了获取时间（从某一时间到另一时间的结果）

var minutes = 1000\*60

var hours = minutes\*60

var days = hours\*24

var years = days\*365

var d = new Date()

var t = d.getTime()

var y = t/years

document.write("It's been: " + y + " years since 1970/01/01!")

**3对以上的简化fetfullyear（）**

var d = new Date()

d.setFullYear(1992,10,3)

document.write(d)

**4 toUTCstring（）**  实现日期的转换

var d = new Date()

document.write (d.toUTCString()

**5 getday（）用来**；显示星期

var d=new Date()

var weekday=new Array(7)

weekday[0]="Sunday"

weekday[1]="Monday"

weekday[2]="Tuesday"

weekday[3]="Wednesday"

weekday[4]="Thursday"

weekday[5]="Friday"

weekday[6]="Saturday"

document.write("Today it is " + weekday[d.getDay()])

（结果便会自动显示星期几）

**实例显示一个钟表（5:59:12）**

function startTime()

{

var today=new Date()

var h=today.getHours()

var m=today.getMinutes()

var s=today.getSeconds()

// add a zero in front of numbers<10

m=checkTime(m)

s=checkTime(s)

document.getElementById('txt').innerHTML=h+":"+m+":"+s

t=setTimeout('startTime()',500)

}

function checkTime(i)

{

if (i<10)

{i="0" + i}

return i

}

（其他的详见参考手册）

**关于数组**

**1创建数组 var a= new arry[] 初始化进行赋值即可**

**2输出数组的内容（for循环）还有就是**

for (x in mycars)

{

document.write(mycars[x] + "<br />")

}

**3合并两个数组 concat()**

document.write(arr.concat(arr2)（初始化放入就可）

**4用数组的元素组成一个字符join()**

var arr = new Array(3)

arr[0] = "Jani"

arr[1] = "Hege"

arr[2] = "Stale"

document.write(arr.join() + "<br />")

document.write(arr.join("."))

（br或者。只是最终的输出结果不同）

**5文字数组（数组内容是文字）进行排序sort()**

document.write(arr.sort()

**js boolean逻辑对象**

用来检验是否为布尔常量

Var b1 =new Boolean（0）；

document.write("0 is boolean "+ b1 +"<br />")

**js math（）算数对象**

round（）是四舍五入

random（）随机取数

min（）

max（） 每次都是在输出的内容添加所需要的算术或者是其他的功能（document.write(需要的功能)）

J**avaScript HTML DOM 对象**

除了内置的 JavaScript 对象以外，你还可以使用 JavaScript 访问并处理所有的 HTML DOM 对象

检查浏览器的版本（version）

var browser=navigator.appName

var b\_version=navigator.appVersion

var version=parseFloat(b\_version)

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

document.write("<br />")

document.write("Browser version: "+ version)

**注意**

Navigator 对象

JavaScript Navigator 对象包含了有关访问者浏览器的所有信息。接下来我们学习 Navigator 对象的两个属性。

appName

保存浏览器类型

appVersion

存有浏览器的版本信息（其他信息中的一项）

**Javascript cookie**

用来识别用户

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

<html>

<head>

<script type="text/javascript">

function getCookie(c\_name)

{

if (document.cookie.length>0)

{

c\_start=document.cookie.indexOf(c\_name + "=")

if (c\_start!=-1)

{

c\_start=c\_start + c\_name.length+1

c\_end=document.cookie.indexOf(";",c\_start)

if (c\_end==-1) c\_end=document.cookie.length

return unescape(document.cookie.substring(c\_start,c\_end))

}

}

return ""

}

**////getcookie是首先创建一个用cookie用来存储时间，名字或者其他**

function setCookie(c\_name,value,expiredays)

{

var exdate=new Date()

exdate.setDate(exdate.getDate()+expiredays)

document.cookie=c\_name+ "=" +escape(value)+

((expiredays==null) ? "" : ";expires="+exdate.toGMTString())

}

**///检查是否已经创建一个cookie**

function checkCookie()

{

username=getCookie('username')

if (username!=null && username!="")

{alert('Welcome again '+username+'!')}

else

{

username=prompt('Please enter your name:',"")

if (username!=null && username!="")

{

setCookie('username',username,365)

}

}

**}////显示存储好就是欢迎词的界面如若不然就是注册表**

</script>

</head>

<body onLoad="checkCookie()">

</body>

</html>

**Javascript表单验证**

function validate\_required(field,alerttxt)

{

with (field)

{

if (value==null||value=="")

{alert(alerttxt);return false}

else {return true}

}

}

///是否已填写表单中的必填（或必选）项目。假如必填或必选项为空，那么警告框会弹出，并且函数的返回值为 false，否则函数的返回值则为 true

**Email验证**

意思就是说，输入的数据必须包含 @ 符号和点号(.)。同时，@ 不可以是邮件地址的首字符，并且 @ 之后需有至少一个点号

function validate\_email(field,alerttxt)

{

with (field)

{

apos=value.indexOf("@")

dotpos=value.lastIndexOf(".")

if (apos<1||dotpos-apos<2)

{alert(alerttxt);return false}

else {return true}

}

}

**Javascript 动画**

<script type="text/javascript">

function mouseOver()

{

document.b1.src ="b\_blue.gif"

}

function mouseOut()

{

document.b1.src ="b\_pink.gif"

}

</script>

在html中直接引用此函数

onmouseOver="mouseOver()"

onmouseOut="mouseOut()"

**javascript图像地图**

<img src="planets.gif" width="145" height="126"

alt="Planets" usemap="#planetmap" />

<area shape ="rect" coords ="0,0,82,126"///确定js所作用的对象（ps来去确定实际内容）

onMouseOver="writeText（内容）

<area shape ="circle" coords ="90,58,3"

onMouseOver="writeText（内容）

//当然这里的writeText是需要在<script>中提前构造的

**Javascript计时**

**简单的计时**

<script type="text/javascript">

function timedMsg()

{

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

}

</script>

升级版本的

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

var t2=setTimeout("document.getElementById('txt').value='4 seconds!'",4000)

var t3=setTimeout("document.getElementById('txt').value='6 seconds!'",6000)

（有两个文本框一个是按钮button。另一个是时间计时（2.4.6））

在一个无穷循环中的计数事件

<script type="text/javascript">

var c=0

var t

function timedCount()

{

document.getElementById('txt').value=c

c=c+1

t=setTimeout("timedCount()",1000)

}

</script>

带有停止按钮的（只需要在html中添加一个新的按钮，链接函数即可）

function stopCount()

{

clearTimeout(t)

}

JavaScript 计时事件

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

在 JavaScritp 中使用计时事件是很容易的，两个关键方法是:

setTimeout()

未来的某时执行代码

clearTimeout()

取消setTimeout()

**Js之输出内容**

document.write() 可用于直接向 HTML 输出流写内容

1输出内容直接用” 内容 ” document.write(“success”);

2通过变量输出内容 var num=6; document.write(num);(引用变量是不用“”)

3输出多项内容用+添加 var a=2; document.write(a+”love”);

4:输出HTML标签，并起作用，标签使用""括起来

<script type="text/javascript">

var mystr="hello";

document.write(mystr+"<br>");//输出hello后，输出一个换行符

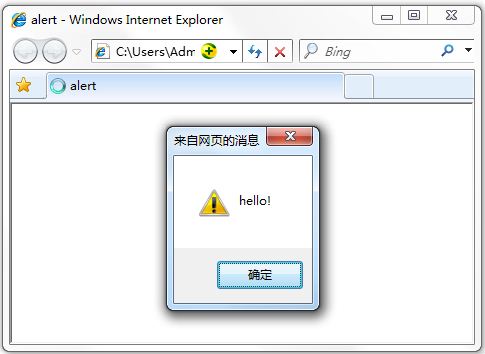
document.write("JavaScript");

</script>

**JavaScript消息框**

**JavaScript-警告（alert 消息对话框**）

alert("hello!");



**JavaScript-确认（confirm 消息对话框）**

confirm 消息对话框通常用于允许用户做选择的动作，

confirm(str)

str 消息对话框要显示的文本

<script type="text/javascript">

**var mymessage=confirm("你喜欢JavaScript吗?");**

if(mymessage==true)

{ document.write("很好,加油!"); }

else

{ document.write("JS功能强大，要学习噢!"); }

</script>

实例化

<script type="text/javascript">

function disp\_confirm()

{

var r=confirm("Press a button")

if (r==true)

{

document.write("You pressed OK!")

}

else

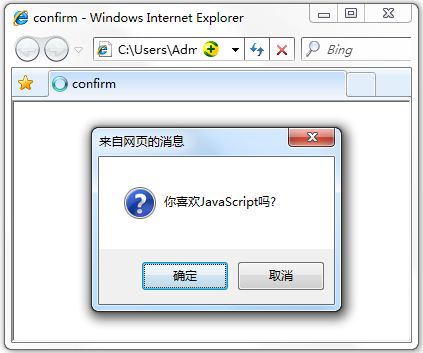
{

document.write("You pressed Cancel!")

}

}

</script>



**JavaScript-提问（prompt 消息对话框）**

prompt弹出消息对话框,通常用于询问一些需要与用户交互的信**息。**

prompt(str1, str2)（str1: 要显示在消息对话框中的文本，不可修改

str2：文本框中的内容，可以修改）

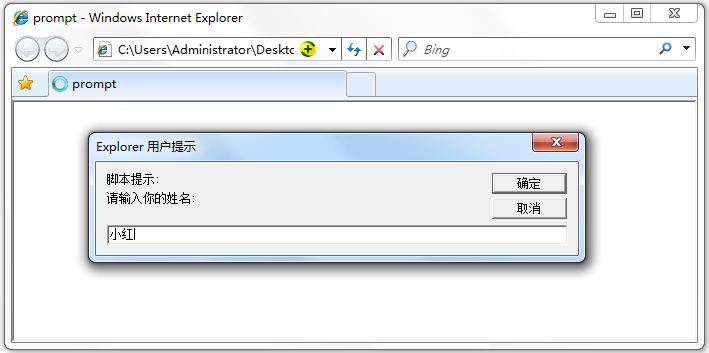
var myname=prompt("请输入你的姓名:");

if(myname!=null)

{ alert("你好"+myname); }

else

{ alert("你好 my friend."); }



**JavaScript-打开新窗口（window.open**）

open() 方法可以查找一个已经存在或者新建的浏览器窗口

window.open([URL], [窗口名称], [参数字符串])

URL：可选参数，在窗口中要显示网页的网址或路径。如果省略这个参数，或者它的值是空字符串，那么窗口就不显示任何文档。

窗口名称：可选参数，被打开窗口的名称。

1.该名称由字母、数字和下划线字符组成。

2."\_top"、"\_blank"、"\_self"具有特殊意义的名称。

\_blank：在新窗口显示目标网页

\_self：在当前窗口显示目标网页

\_top：框架网页中在上部窗口中显示目标网页

3.相同 name 的窗口只能创建一个，要想创建多个窗口则 name 不能相同。

4.name 不能包含有空格。

参数字符串：可选参数，设置窗口参数，各参数用逗号隔开。

eg：<script type="text/javascript"> window.open('http://www.imooc.com','\_blank','width=300,height=200,menubar=no,toolbar=no, status=no,scrollbars=yes')

</script>（注意最后运行结果是否与浏览器兼容的问题）



（其中的最后一项参数指标）

**JavaScript-关闭窗口（window.close）**

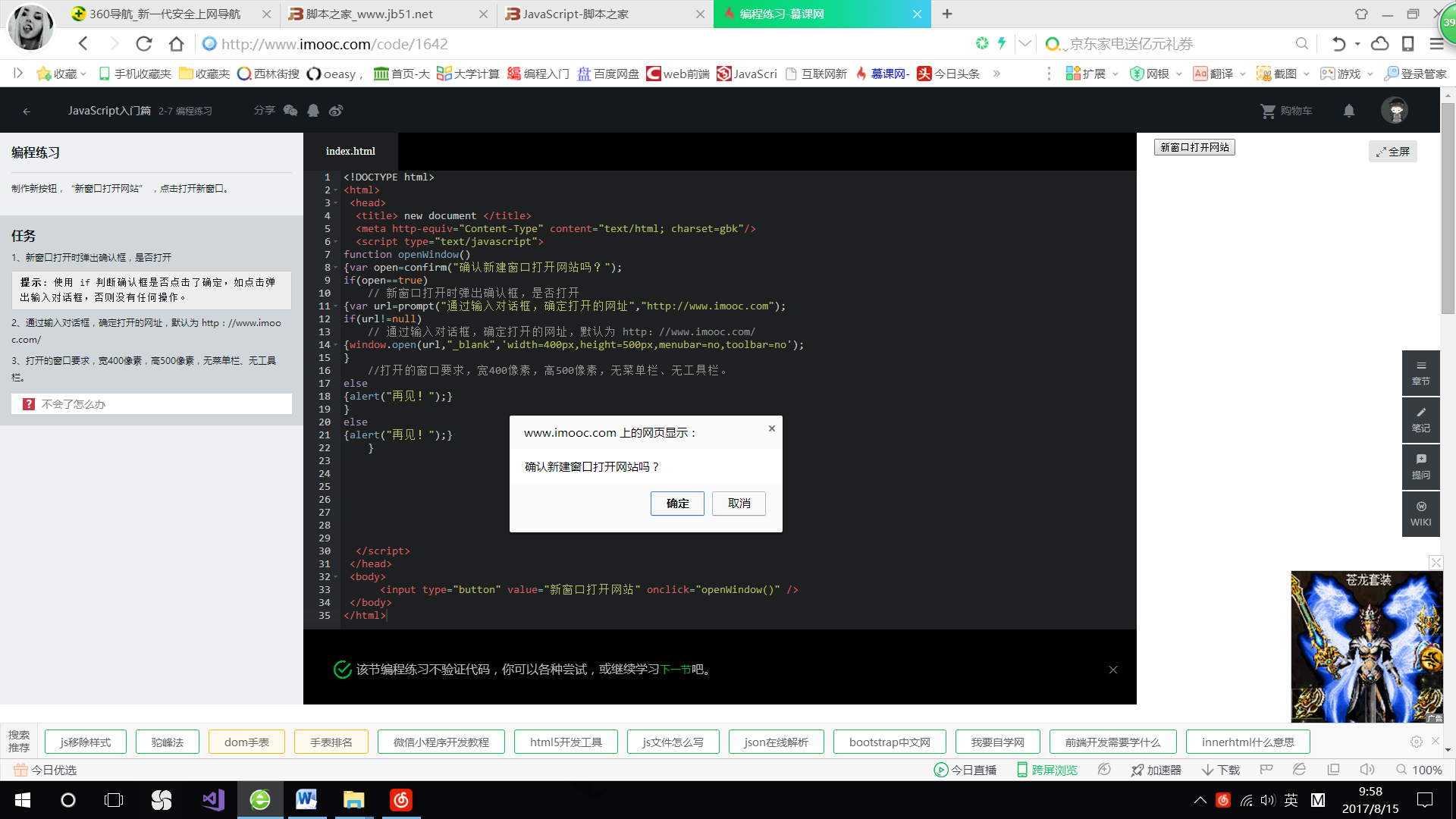
window.close(); //关闭本窗口 （<窗口对象>.close(); //关闭指定的窗口）

<script type="text/javascript">

var mywin=window.open('http://www.imooc.com'); //将新打的窗口对象，存储在变量mywin中

mywin.close();

</script>



**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

HTML文档可以说由节点构成的集合，三种常见的DOM节点:

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"。

**通过id获取元素**

document.getElementById(“id”)

eg: <body>

<p id="con">JavaScript</p>

<script type="text/javascript">

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

document.write("结果:"+mychar); //输出获取的P标签。

</script>

</body>

**innerHTML属性（**用于获取或替换HTML**）**

语法：object.innerHTML

注意:

1.Object是获取的元素对象，如通过document.getElementById("ID")获取的元素。

2.注意书写，innerHTML区分大小写。

<script type="text/javascript">

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

document.write("原标题:"+mychar.innerHTML+"<br>"); //输出原h2标签内容

mychar.innerHTML="Hello world!";

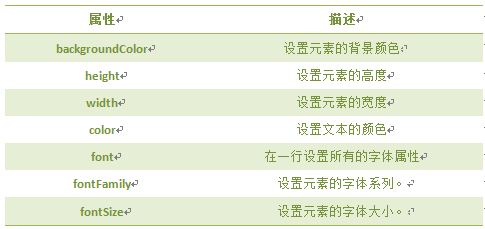
document.write("修改后的标题:"+mychar.innerHTML); //输出修改后h2标签内容

</script>

**改变HTML样式**

语法：Object.style.property=new style;

基本属性表（Property）



Eg：<p id="pcon">Hello World!</p>

<script>

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

mychar.style.color="red";

mychar.style.fontSize="20";

mychar.style.backgroundColor ="blue";

</script> （改变样式）

**显示和隐藏的效果**

网页中经常会看到隐藏和显示的效果，就是通过display的属性来设置的

Eg：object.style.display =vaule

注意:Object是获取的元素对象，如通过document.getElementById("id")获取的元素。

Value的取值 block（为块级元素即显示） none（不显示）

控制类名（className属性）

Object.className=classname

作用:

1.获取元素的class 属性

2. 为网页内的某个元素指定一个css样式来更改该元素的外观