1：可以将JavaScript代码放在html文件中任何位置，但是我们一般放在网页的head或者body部分。

2：Js代码必须写在<script type="text/javascript"> </script>之间 或者用<scriptsrc="script.js"></script>引入js文件。

3： 变量和变量赋值var mychar;

mychar="javascript";或者var mynum = 8;

函数定义：

function函数名()

{

函数代码;

}

函数定义，调用，以及html的调用。可以直接在html里调用。



function a()

{

函数代码;

return ?;(返回函数值)

}

var b = a();(将函数返回值保存在b中);

4：向网页输出语句: document.write("I");

Js输出空格问题：解决方法：1使用html标签&nbsp来解决：document.write("&nbsp;&nbsp;"+"1"+"&nbsp;&nbsp;&nbsp;&nbsp;"+"23"); 2使用css式样来解决“white-space:pre;”，该式样标识”空白会被浏览器保留”：document.write("<span style='white-space:pre;'>"+"  1     2   3 "+"</span>");:

5：js警告：alert(字符串或变量); var a=alert(“bbbbb”);点击确定则a=“true”。

6：js确认：confirm(str); 弹出含str的对话框，点确定返回true，点取消返回false；

<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>

6：js提问：**prompt(str1, str2)，**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**;** 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； 点击确定按钮，文本框中的内容将作为函数返回值，当点击取消按钮，将返回**null；**

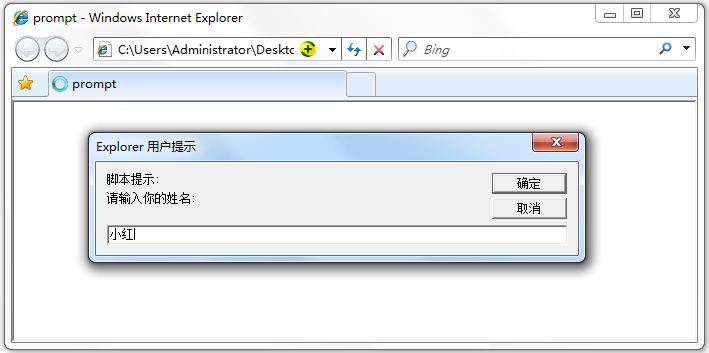
var myname=prompt("请输入你的姓名:");

if(myname!=null)

{ alert("你好"+myname); }

else

{ alert("你好 my friend."); }



7：js打开新窗户：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。



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

</script>

8：js关闭窗口：window.close(); //关闭本窗口

或者<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> 打开新窗口时就关闭了，故看不到新窗口。

9：DOM：文档对象模型DOM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）。

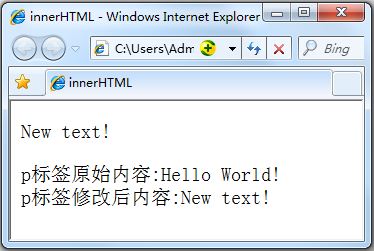
将HTML代码分解为DOM节点层次图:



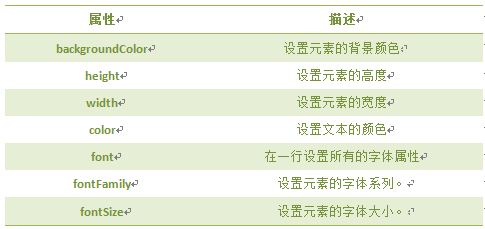
10：通过id获取元素：document.getElementById(“id”)

11：innerHTML 属性：用于获取或替换 HTML 元素的内容：**Object.innerHTML：**Object是获取的元素对象，如通过document.getElementById("ID")获取的元素。



结果

12：改变HTML样式：HTML DOM 允许 JavaScript 改变 HTML 元素的样式：Object.style.property=new style; Object是获取的元素对象，如通过document.getElementById("id")获取的元素。



<p id="pcon">Hello World!</p>

<script>

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

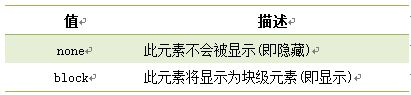
mychar.style.color="red";

mychar.style.fontSize="20";

mychar.style.backgroundColor ="blue";

</script>

13：display属性：显示和隐藏：Object.style.display = value；Object依旧为取到的对象；





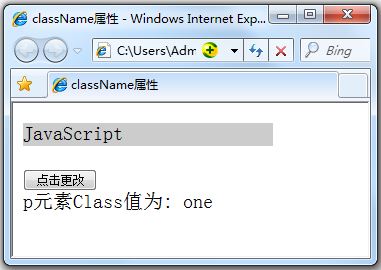
14：className 属性：设置或返回元素的class 属性：object.className = classname；

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



点击“点击修改”就会修改显示的式样。

15：创建数组：var myarray= new Array(8);如果数组为空则输出：undefined。

二维数组 myarr[0][1] ,0 表示表的行，1表示表的列。定义：var arr=[[1,2,3],[4,5,6]];

16：事件：JavaScript 创建动态页面。事件是可以被 JavaScript 侦测到的行为



17:鼠标单击（onclick）常与按钮一起使用。

18：鼠标经过事件(onmouseover)：<input name="password" type="password" onmouseover="message()">当鼠标经过password时弹出消息框并调用message函数。

19：鼠标移开事件（onmouseout）；当鼠标移开时弹出消息框并调用message函数。

20：光标聚焦事件（onfocus）：<input name="password" type="password" onfocus ="message()">：当光标移至文本框时，即焦点在文本框内触发该事件。

21：失焦事件（onblur）：当光标移开当前聚焦对象时触发；

22：内容选中事件（onselect）：当文本框或文本域中的文字被选中时触发,同时调用message：<textarea name="summary" cols="60" rows="5" onselect =”message”>请写入个人简介</textarea>

23：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。

24：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。

<body onload="message()">

欢迎学习JavaScript。

</body>

25：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。

<script type="text/javascript">

window.onunload = onunload\_message;

function onunload\_message(){

alert("您确定离开该网页吗？");

}

</script>

26： Date对象中处理时间和日期的常用方法：



自定义初始值：var d = new Date(2012, 10, 1);或var d = new Date('Oct 1, 2012');

getDay返回星期，0到6；0为星期日。

27：charAt（）方法返回指定位置的字符，如果越界则返回空字符串。

28：indexOf（）方法返回某个指定的字符串在字符串中首次出现的位置：stringObject.indexOf(substring, startpos)；substring为指定的字符串，startpos开始的位置，如果没有则从头开始。如果要检索的字符串没有出现则返回-1。

29：window对象：

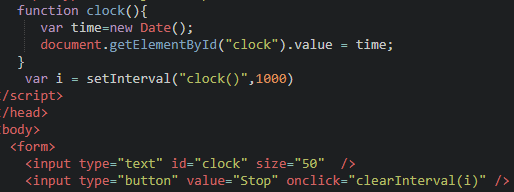


30：js计时器：

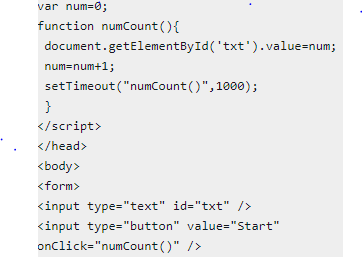


31：setInterval(代码,交互时间); 在执行时,从载入页面后每隔指定的时间执行代码。毫秒级

setInterval("clock()",1000)或setInterval(clock,1000)；

32: clearInterval() 方法可取消由 setInterval() 设置的交互时间。clearInterval(id),id为setInterval（）返回的id值。代码：

33：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。setTimeout(代码,延迟时间);毫秒为单位。

当按钮被点击后，输入域便从0开始计数。

34：clearTimeout()和setTimeout()一起使用，停止计时器。clearTimeout(id)，id：由 setTimeout() 返回的 ID 值。该值标识要取消的延迟执行代码块。



点击Start按钮后，开始计数, 点击Stop按钮后，停止计数.

35: 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。window.history.[属性|方法],

Window可以省略。属性：length，返回历史列表中的URL数量。

方法：back（）加载history列表中的上一个URL，语法window.history.back()，相当于go(-1);

forward（）加载history列表中的下一个URL, 相当于go(1)；go（）加载history列表中的某个具体的页面。

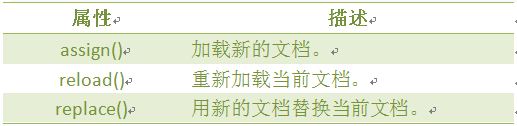
36: location用于获取或设置窗体的URL，并且可以用于解析URL。语法：location.[属性|方法]，location对象属性图示:



对象属性：

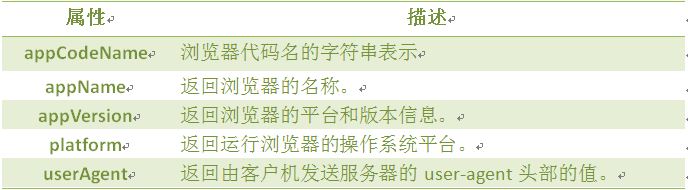


对象方法：

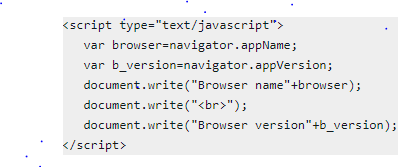


显示当前文档URL：document.write(window.location.href);

**3**7：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。



查看浏览器的名称和版本：



screen对象用于获取用户的屏幕信息，语法：window.screen.属性

