

JavaScript Study Notes

1 准备

- 1-1 为什么学习 JavaScript

一、你知道，为什么 JavaScript 非常值得我们学习吗？

1. 所有主流浏览器都支持 JavaScript。
2. 目前，全世界大部分网页都使用 JavaScript。
3. 它可以让网页呈现各种动态效果。
4. 做为一个 Web 开发者，如果你想提供漂亮的网页、令用户满意的上网体验，JavaScript 是必不可少的工具。

二、易学性

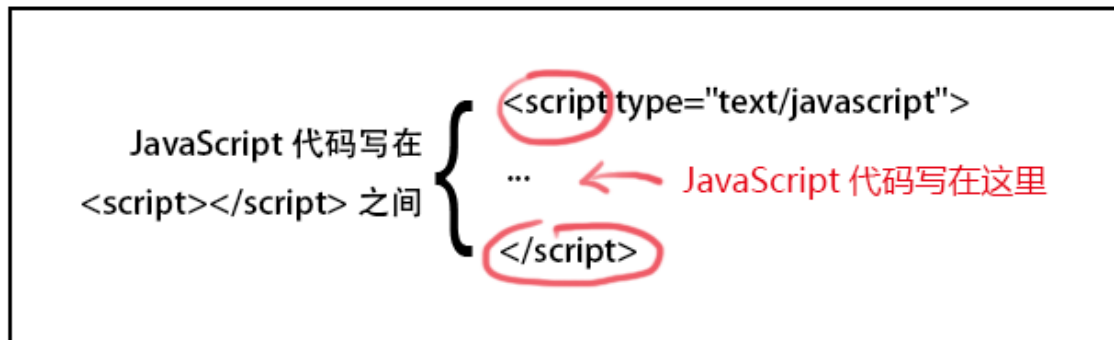
1. 学习环境无外不在，只要有文本编辑器，就能编写 JavaScript 程序。
2. 我们可以用简单命令，完成一些基本操作。

三、从哪开始学习呢？

学习 JavaScript 的起点就是处理网页，所以我们先学习基础语法和如何使用 DOM 进行简单操作。

- 1-2 如何插入 JavaScript

我们来看看如何写入 JS 代码？你只需一步操作，使用 `<script>` 标签在 HTML 网页中插入 JavaScript 代码。注意，`<script>` 标签要成对出现，并把 JavaScript 代码写在 `<script></script>` 之间。



`<script type="text/javascript">` 表示在 `<script></script>` 之间的是文本类型 (text), javascript 是为了告诉浏览器里面的文本是属于 JavaScript 语言。

- 1-3 引用 JavaScript 外部文件

通过前面知识学习，我们知道使用 `<script>` 标签在 HTML 文件中添加 JavaScript 代码，如图：

index.html

```
<html>
  <head>
    <script type="text/JavaScript">
      alert("JS代码");
    </script>
  </head>
  <body>
  </body>
</html>
```

在HTML中
插入JS代码

JavaScript 代码只能写在 HTML 文件中吗？当然不是，我们可以把 HTML 文件和 JS 代码分开，并单独创建一个 JavaScript 文件（简称 JS 文件），其文件后缀通常为 .js，然后将 JS 代码直接写在 JS 文件中。

index.html

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

script.js

```
alert("JS代码");
```

JS代码
直接写入JS文件中

注意：在 JS 文件中，不需要 `<script>` 标签，直接编写 JavaScript 代码就可以了。

JS 文件不能直接运行，需嵌入到 HTML 文件中执行，我们需在 HTML 中添加如下代码，就可将 JS 文件嵌入 HTML 文件中。

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

在HTML中引入JS文件

index.html

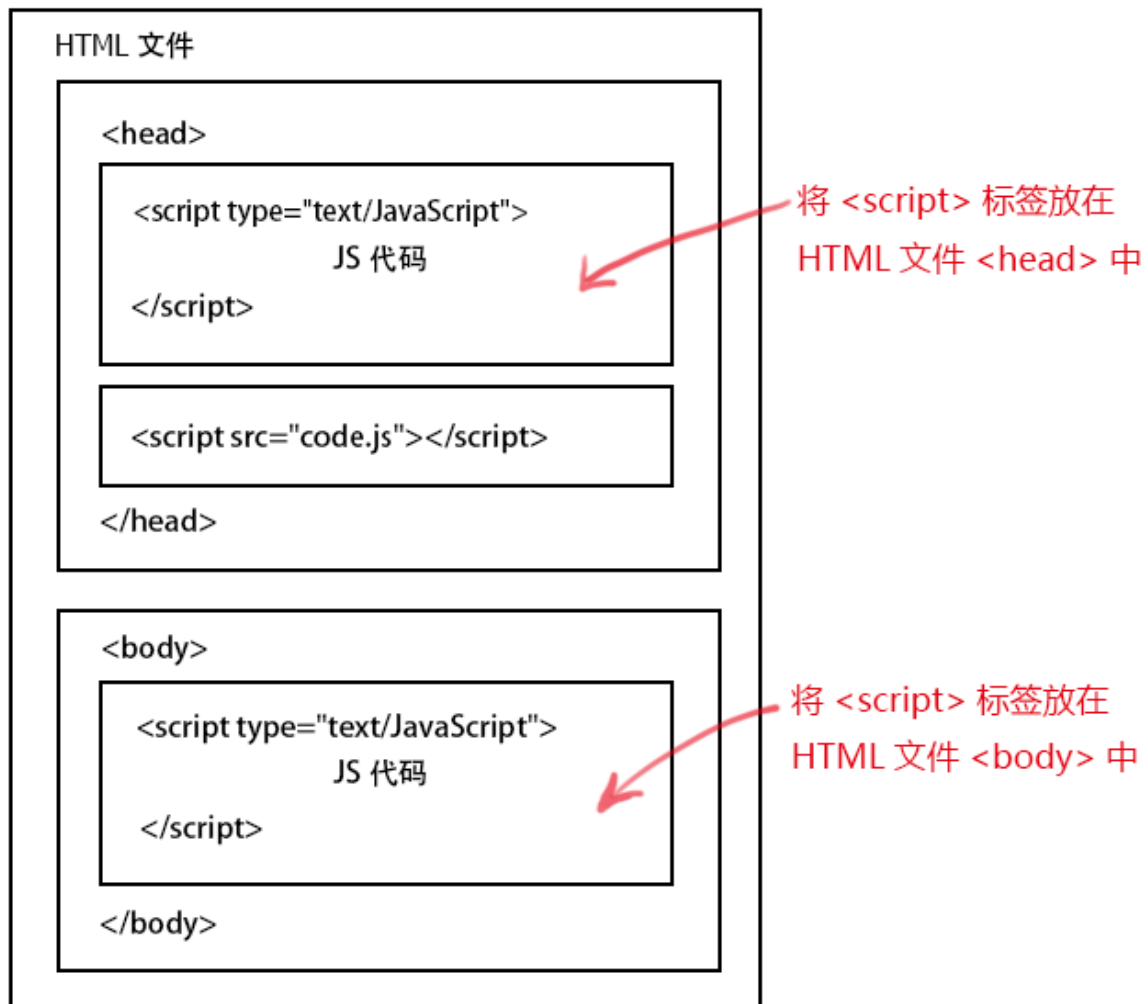
```
<html>
  <head>
    <script src="script.js"></script>
  </head>
  <body>
  </body>
</html>
```

script.js

```
alert("JS代码");
```

- 1-4 JavaScript 在页面中的位置

我们可以将 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。放在 `<head>` 部分 最常用的方式是在页面中 head 部分放置 `<script>` 元素，浏览器解析 head 部分就会执行这个代码，然后才解析页面的其余部分。放在 `<body>` 部分 JavaScript 代码在网页读取到该语句的时候就会执行。



注意：javascript 作为一种脚本语言可以放在 html 页面中任何位置，但是浏览器解释 html 时是按先后顺序的，所以前面的 script 就先被执行。比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。

- 1-5 JavaScript - 认识语句和符号

JavaScript 语句是发给浏览器的命令。这些命令的作用是告诉浏览器要做的事情。

每一句 JavaScript 代码格式： `语句;`

先来看看下面代码

```
<script type="text/javascript">  
  alert("hello!");  
</script>
```

例子中的 `alert("hello!");` 就是一个 JavaScript 语句。

一行的结束就被认定为语句的结束，通常在结尾加上一个分号 `;` 来表示语句的结束。

看看下面这段代码，有三条语句，每句结束后都有 `;`，按顺序执行语句。

```
<script type="text/javascript">
    document.write("I");
    document.write("love");
    document.write("JavaScript");
</script>
```

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。
- 1-6 JavaScript - 注释

注释的作用是提高代码的可读性，帮助自己和别人阅读和理解你所编写的 JavaScript 代码，注释的内容不会在网页中显示。注释可分为单行注释与多行注释两种。

我们为了方便阅读，注释内容一般放到需要解释语句的结尾处或周围。

单行注释，在注释内容前加符号 “//”。

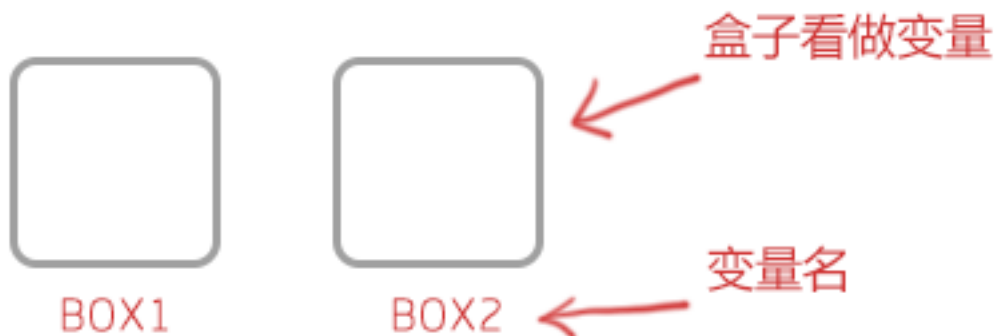
```
<script type="text/javascript">
    document.write(" 单行注释使用'//'"); // 我是注释，该语句功能在网页中输出内容
</script>
```

多行注释以 “/*” 开始，以 “*/” 结束。

```
<script type="text/javascript">
    document.write(" 多行注释使用/* 注释内容 */");
    /*
        多行注释
        养成书写注释的良好习惯
    */
</script>
```

- 1-7 JavaScript - 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，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）。



定义变量使用关键字 `var`，语法如下：

```
var 变量名
```

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

变量要先声明再赋值，如下：

```
var mychar;  
mychar="javascript";  
var mynum = 6;
```

变量可以重复赋值，如下：

```
var mychar;  
mychar="javascript";  
mychar="hello";
```

注意：

1. 在 JS 中区分大小写，如变量 `mychar` 与 `myChar` 是不一样的，表示是两个变量。
2. 变量虽然也可以不声明，直接使用，但不规范，需要先声明，后使用。

- 1-8 JavaScript - 判断语句

`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 }
```

假设我们通过年龄来判断是否为成年人，如年龄大于等于 18 岁，是成年人，否则不是成年人。代码表示如下：

```
<script type="text/javascript">  
  var myage = 18;  
  if(myage>=18) //myage>=18 是判断条件  
  { document.write(" 你是成年人。");}  
  else //否则年龄小于 18  
  { document.write(" 未满 18 岁，你不是成年人。");}  
</script>
```

- 1-9 JavaScript - 函数

函数是完成某个特定功能的一组语句。如没有函数，完成任务可能需要五行、十行、甚至更多的代码。这时我们就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，就省重复输入大量代码的麻烦。

如何定义一个函数呢？基本语法如下：

```
function 函数名()  
{  
    函数代码;  
}
```

说明：

1. function 定义函数的关键字。
2. “函数名”是为函数取的名字。
3. “函数代码”替换为完成特定功能的代码。

我们来编写一个实现两数相加的简单函数，并给函数起个有意义的名字：“add2”，代码如下：

```
function add2(){  
    var sum = 3 + 2;  
    alert(sum);  
}
```

函数调用：

函数定义好后，是不能自动执行的，所以需调用它，只需直接在需要的位置写函数就 ok 了，代码如下：

```
<!DOCTYPE HTML>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />  
        <title>函数</title>  
        <script type="text/javascript">  
            function add2() {  
                sum = 5 + 6;  
                alert(sum);  
            }  
            add2(); // 调用函数，直接写函数名  
        </script>  
    </head>  
    <body>  
        <form>  
            <input type="button" value=" 点击" onclick="add2()" />  
            <!-- 单击按钮后，调用函数，onclick 为点击事件 -->  
        </form>  
    </body>  
</html>
```

2 常用互动方法

- 2-1 JavaScript - 输出内容

`document.write()` 可用于直接向 HTML 输出流写内容。简单的说就是直接在网页中输出内容。

第一种：输出内容用” “括起，直接输出” ”号内的内容

```
<script type="text/javascript">  
    document.write("I love JavaScript! "); //内容用""括起来，""里的内容直接输出。  
</script>
```

第二种：通过变量，输出内容

```
<script type="text/javascript">  
    var mystr="hello world!";  
    document.write(mystr); //直接写变量名，输出变量存储的内容。  
</script>
```

第三种：输出多项内容，内容之间用 + 号连接

```
<script type="text/javascript">
    var mystr="hello";
    document.write(mystr+"I love JavaScript"); //多项内容之间用 + 号连接
</script>
```

第四种：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>
```

输出空格：

- 2-2 JavaScript - 警告 (alert 消息对话框)

我们在访问网站的时候，有时会突然弹出一个窗口，上面写着一段提示信息文字。如果你不点击“确定”，就不能对网页做任何操作，这个小窗口就是使用 alert 实现的。

语法：

```
alert(字符串或变量);
```

看下面的代码：

```
<script type="text/javascript">
    var mynum = 30;
    alert("hello!");
    alert(mynum);
</script>
```

注：alert 弹出消息对话框（包含一个确定按钮）。

结果：按顺序弹出消息框

注意：

1. 在点击对话框“确定”按钮前，不能进行任何其它操作。
2. 消息对话框通常可以用于调试程序。
3. alert 输出内容，可以是字符串或变量，与 document.write 相似。

- 2-3 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，如：“你对吗？”等。弹出对话框（包括一个确定按钮和一个取消按钮）。

语法：

```
confirm(str);
```

参数说明：

str：在消息对话框中要显示的文本
返回值：Boolean 值

返回值：

当用户点击“确定”按钮时，返回 true
当用户点击“取消”按钮时，返回 false

注：通过返回值可以判断用户点击了什么按钮

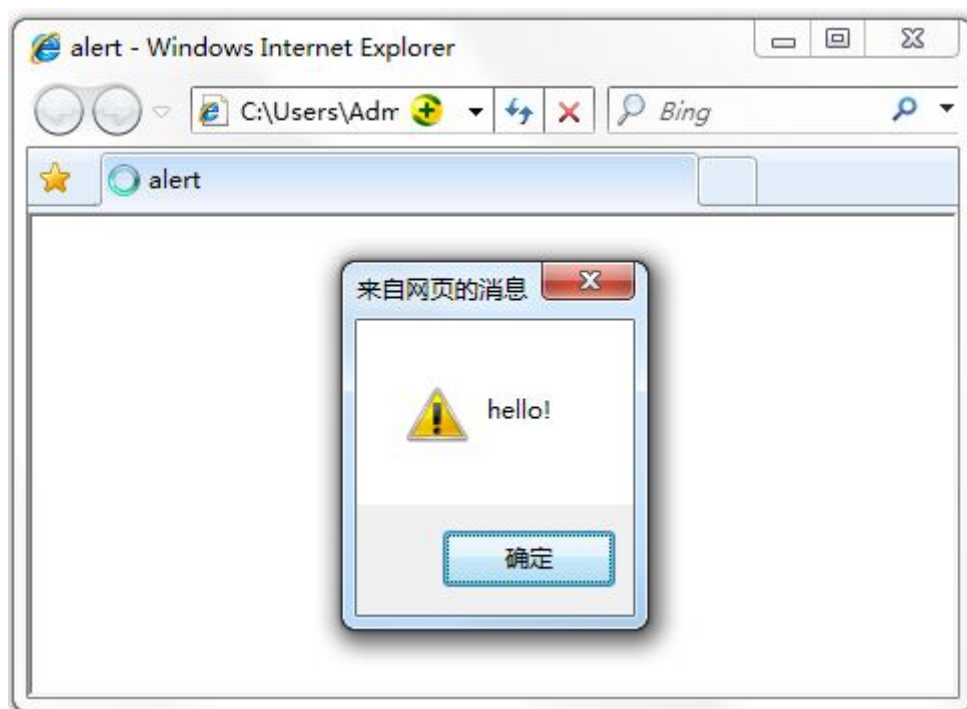


Figure 1: img

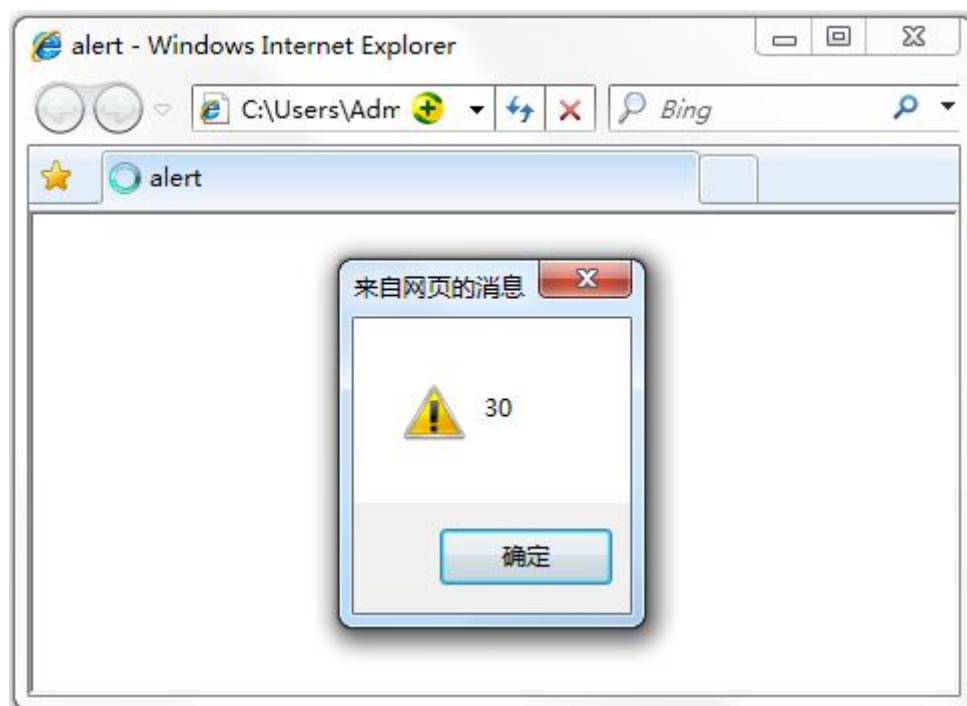
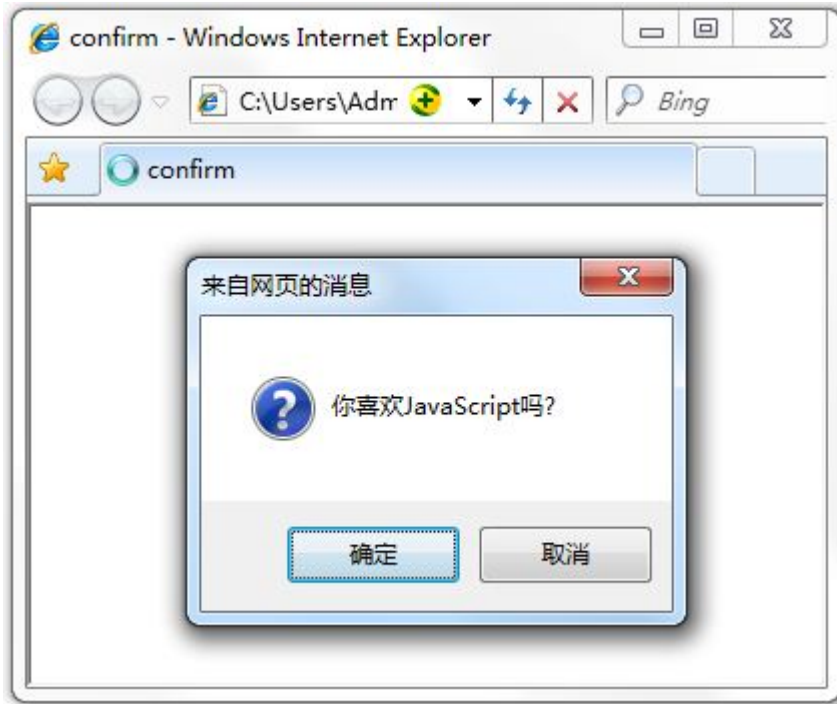


Figure 2: img

看下面的代码:

```
<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>
```

结果:



注: 消息对话框是排它的, 即用户在点击对话框按钮前, 不能进行任何其它操作。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>confirm</title>
    <script type="text/javascript">
      function rec(){
        var mymessage=confirm(" 你是女士吗? ");
        if(mymessage==true)
        {
          document.write(" 你是女士!");
        }
        else
        {
          document.write(" 你是男士!");
        }
      }
    </script>
  </head>
```

```
<body>
  <input name="button" type="button" onClick="rec()" value=" 点击我，弹出确认对话框" />
</body>
</html>
```

- 2-4 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。弹出消息对话框（包含一个确定按钮、取消按钮与一个文本输入框）。

语法：

```
prompt(str1, str2);
```

参数说明：

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

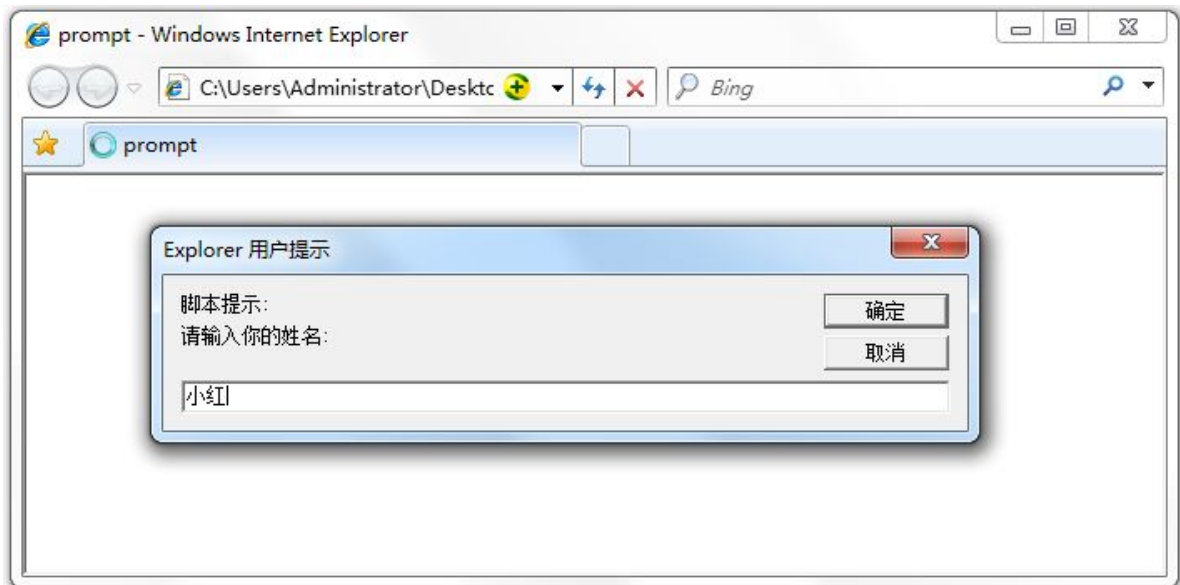
返回值：

1. 点击确定按钮，文本框中的内容将作为函数返回值
2. 点击取消按钮，将返回 null

看看下面代码：

```
var myname=prompt(" 请输入你的姓名:");
if(myname!=null)
{ alert(" 你好"+myname); }
else
{ alert(" 你好 my friend."); }
```

结果：



注：在用户点击对话框的按钮前，不能进行任何其它操作。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```

<title>prompt</title>
<script type="text/javascript">
function rec(){
    var score; //score 变量，用来存储用户输入的成绩值。
    score = prompt(" 你的成绩是: ");
    if(score>=90)
    {
        document.write(" 你很棒!");
    }
    else if(score>=75)
    {
        document.write(" 不错哟!");
    }
    else if(score>=60)
    {
        document.write(" 要加油!");
    }
    else
    {
        document.write(" 要努力了!");
    }
}
</script>
</head>
<body>
    <input name="button" type="button" onClick="rec()" value=" 点击我，对成绩做评价!" />
</body>
</html>

```

- 2-5 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。

语法：

```
window.open([URL], [窗口名称], [参数字符串])
```

参数说明：

1. 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。
2. 窗口名称：可选参数，被打开窗口的名称。
 - 该名称由字母、数字和下划线字符组成。
 - “_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
 - 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。
 - name 不能包含有空格。
3. 参数字符串：可选参数，设置窗口参数，各参数用逗号隔开。

参数表：

参数↴	值↴	说明↴
top↴	Number↴	窗口顶部离开屏幕顶部的像素数↴
left↴	Number↴	窗口左端离开屏幕左端的像素数↴
width↴	Number↴	窗口的宽度↴
height↴	Number↴	窗口的高度↴
menubar↴	yes,no↴	窗口有没有菜单↴
toolbar↴	yes,no↴	窗口有没有工具条↴
scrollbars↴	yes,no↴	窗口有没有滚动条↴
status↴	yes,no↴	窗口有没有状态栏↴

例如: 打开 <http://www.imooc.com> 网站, 大小为 300px * 200px (是指弹出的新窗口的大小, 而不是网页的大小), 无菜单, 无工具栏, 无状态栏, 有滚动条窗口:

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

注意: 运行结果考虑浏览器兼容问题。

- 2-6 JavaScript - 关闭窗口 (window.close)

close() 关闭窗口

用法:

```
window.close(); //关闭本窗口
```

或

```
<窗口对象>.close(); //关闭指定的窗口
```

例如: 关闭新建的窗口。

```
<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>
```

注意: 上面代码在打开新窗口的同时, 关闭该窗口, 看不到被打开的窗口。

- 2-7 编程练习

制作新按钮, “新窗口打开网站”, 点击打开新窗口。

```
<!DOCTYPE html>
<html>
<head>
    <title> new document </title>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
    <script type="text/javascript">
        function openWindow(){
            // 新窗口打开时弹出确认框, 是否打开
            var web = confirm(" 你是否要打开网站? ");
            if (web == true) {
```

```

        // 通过输入对话框，确定打开的网址，默认为 http: //www.imoooc.com/
        var url = prompt(" 请输入网址","http: //www.imoooc.com/");
        if (url != null) {
            //打开的窗口要求，宽 400 像素，高 500 像素，无菜单栏、无工具栏
            window.open('url','_blank','width=400,height=500');
        } else {
            alert(" 再见");
        }
    } else {
        alert(" 再见");
    }
}
</script>
</head>
<body>
    <input type="button" value=" 新窗口打开网站" onclick="openWindow()" />
</body>
</html>

```

3 DOM 操作

- 3-1 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）。

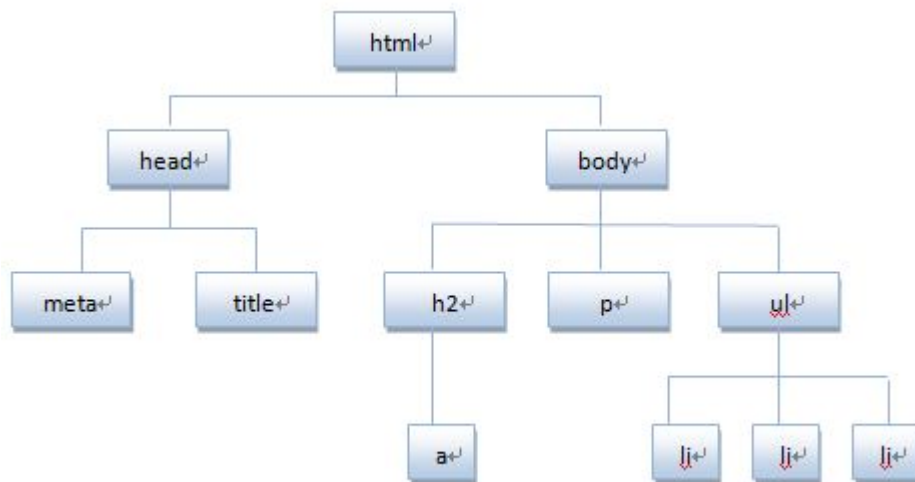
先来看看下面代码：

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
    <title>DOM</title>
  </head>
  <body>
    <h2><a href="http://www.imoooc.com">javascript DOM</a></h2>
    <p>对 HTML 元素进行操作，可添加、改变或移除 CSS 样式等</p>
    <ul>
      <li>JavaScript</li>
      <li>DOM</li>
      <li>CSS</li>
    </ul>
  </body>
</html>

```

将 HTML 代码分解为 DOM 节点层次图：

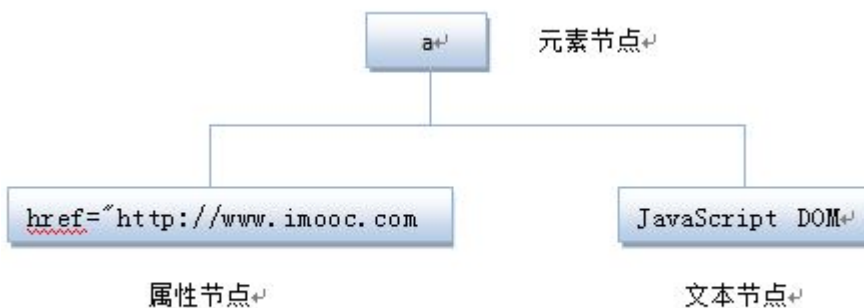


HTML 文档可以说由节点构成的集合，三种常见的 DOM 节点：

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 `...` 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"`。

看下面代码：

```
<a href="http://www.imooc.com">JavaScript DOM</a>
```



• 3-2 通过 ID 获取元素

学过 HTML/CSS 样式，都知道，网页由标签将信息组织起来，而标签的 id 属性值是唯一的，就像是每人有一个身份证号一样，只要通过身份证号就可以找到相对应的人。那么在网页中，我们通过 id 先找到标签，然后进行操作。

语法：

```
document.getElementById("id")
```

看看下面代码：

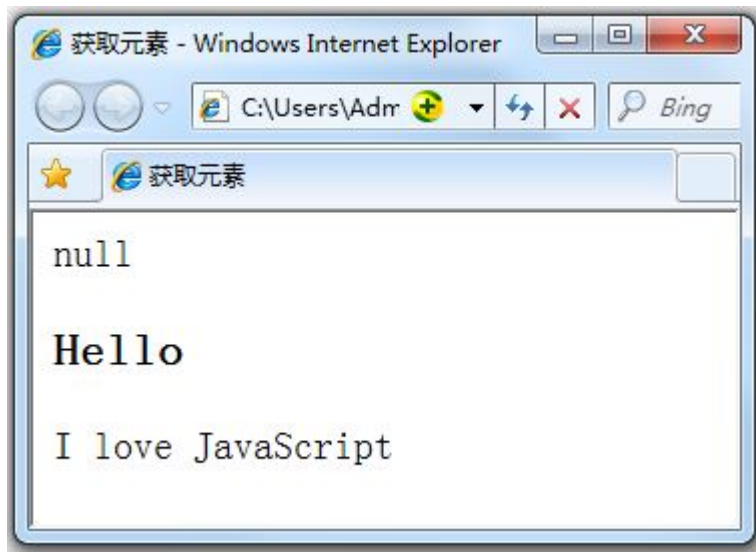
```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
```

```

<title>获取元素</title>
<script type="text/javascript">
    var mye = document.getElementById("con"); // 获取元素存储在变量 mye 中
    document.write(mye); // 输出变量 mye
</script>
</head>
<body>
    <h3>Hello</h3>
    <p id="con">I love JavaScript</p>
</body>
</html>

```

结果: null 或 [object HTMLParagraphElement]



注: 获取的元素是一个对象, 如想对元素进行操作, 我们要通过它的属性或方法。

- 3-3 innerHTML 属性

innerHTML 属性用于获取或替换 HTML 元素的内容。

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。

** 我们通过 id="con" 获取

元素, 并将元素的内容输出和改变元素内容, 代码如下:**

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
    <title>innerHTML</title>
</head>
<body>

```

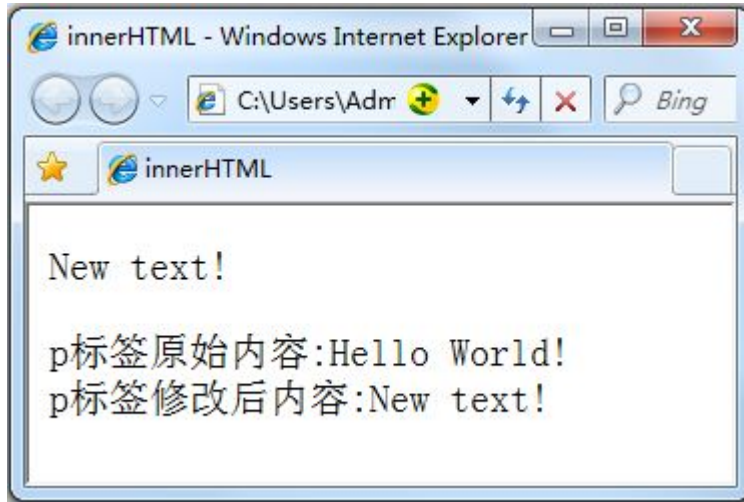


```

<p id="con">Hello World!</p>
<script type="text/javascript">
    var mycon = document.getElementById("con"); // 获取元素存储在变量 mye 中
    document.write("p 标签原始内容: " + mycon.innerHTML + "<br />"); // 输入元素内容
    mycon.innerHTML = "New text!";
    document.write("p 标签修改后的内容: " + mycon.innerHTML);
</script>
</body>
</html>

```

结果:



```

<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>innerHTML</title>
  </head>
  <body>
    <h2 id="con">javascript</H2>
    <p> JavaScript 是一种基于对象、事件驱动的简单脚本语言，嵌入在 HTML 文档中，由浏览器负责解释和执行，在网页上产生动态的
    <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>
  </body>
</html>

```

Hello World!

JavaScript是一种基于对象、事件驱动的简单脚本语言，嵌入在HTML文档中，由浏览器负责解释和执行，在网页上产生动态的显示效果并实现与用户交互功能。

原标题:javascript

修改后的标题:Hello World!

- 3-4 改变 HTML 样式

HTML DOM 允许 JavaScript 改变 HTML 元素的样式。如何改变 HTML 元素的样式呢？

语法：

```
Object.style.property=new style;
```

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

基本属性表 (property)：

属性↗	描述↗
backgroundColor↗	设置元素的背景颜色↗
height↗	设置元素的高度↗
width↗	设置元素的宽度↗
color↗	设置文本的颜色↗
font↗	在一行设置所有的字体属性↗
fontFamily↗	设置元素的字体系列。↗
fontSize↗	设置元素的字体大小。↗

注意：该表只是一小部分 CSS 样式属性，其它样式也可以通过该方法设置和修改。

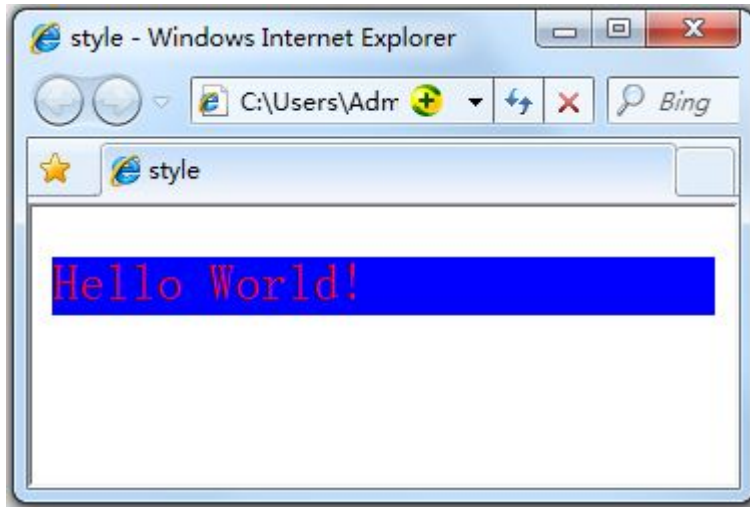
看看下面的代码：

改变

元素的样式，将颜色改为红色，字号改为 20，背景颜色改为蓝：

```
<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>
```

结果：



- 3-5 显示和隐藏 (display 属性)

网页中经常会看到显示和隐藏的效果，可通过 display 属性来设置。

语法：

```
Object.style.display = value
```

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

value 取值：

值	描述
none	此元素不会被显示(即隐藏)
block	此元素将显示为块级元素(即显示)

看看下面代码：

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>display</title>
    <script type="text/javascript">
      function hidetext() {
        document.getElementById("con").style.display="none";
      }
      function showtext() {
        document.getElementById("con").style.display="block";
      }
    </script>
  </head>
  <body>
    <h1>Javascript</h1>
    <p id="con">作为一个 Web 开发师来说，如果你想要提供漂亮的网页、令用户满意的上网体验，JavaScript 是必不可少的工具。
    <form>
      <input type="button" onclick="hidetext()" value=" 不显示段落内容" />
      <input type="button" onclick="showtext()" value=" 显示段落内容" />
```

```
</form>
</body>
</html>
```

JavaScript

做为一个Web开发师来说，如果你想提供漂亮的网页、令用户满意的上网体验，JavaScript是必不可少的工具。

隐藏内容

显示内容

- 3-6 控制类名 (className 属性)

className 属性设置或返回元素的 class 属性。

语法：

```
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

** 看看下面代码，获得

元素的 class 属性和改变 className: **

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <title>className 属性</title>
    <style>
      body{font-size:16px;}
      .one{
        border:1px solid #eee;
        width:230px;
        height:50px;
        background:#ccc;
        color:red;
      }
      .two{
        border:1px solid #ccc;
        width:230px;
        height:50px;
        background:#9CF;
        color:blue;
      }
    </style>
  </head>

  <body>
```

```

<p id="p1" > JavaScript 使网页显示动态效果并实现与用户交互功能。</p>
<input type="button" value=" 添加样式" onclick="add()"/>
<p id="p2" class="one">JavaScript 使网页显示动态效果并实现与用户交互功能。</p>
<input type="button" value=" 更改外观" onclick="modify()"/>

<script type="text/javascript">
    function add(){
        var p1 = document.getElementById("p1");
        p1.className="one";
    }
    function modify(){
        var p2 = document.getElementById("p2");
        p2.className="two";
    }
</script>
</body>
</html>

```

JavaScript使网页显示动态效果并实现与用户交互功能。

添加样式

JavaScript使网页显示动态效果并实现与用户交互功能。

更改外观

JavaScript使网页显示动态效果并实现与用户交互功能。

添加样式

JavaScript使网页显示动态效果并实现与用户交互功能。

更改外观

4 编程挑战

• 4-1 编程挑战

请编写“改变颜色”、“改变宽高”、“隐藏内容”、“显示内容”、“取消设置”的函数，点击相应按钮执行相应操作，点击“取消设置”按钮后，提示是否取消设置，如是执行操作，否则不做操作。

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" Content="text/html; charset=utf-8" />
    <title>javascript</title>
    <style type="text/css">
      body{font-size:12px;}
      #txt{
        height:400px;
        width:600px;
        border:#333 solid 1px;
        padding:5px;
      }
      p{
        line-height:18px;
        text-indent:2em;
      }
    </style>
  </head>

  <body>
    <h2 id="con">JavaScript 课程</h2>
    <div id="txt">
      <h5>JavaScript 为网页添加动态效果并实现与用户交互的功能。</h5>
      <p>1. JavaScript 入门篇，让不懂 JS 的你，快速了解 JS。</p>
      <p>2. JavaScript 进阶篇，让你掌握 JS 的基础语法、函数、数组、事件、内置对象、BOM 浏览器、DOM 操作。</p>
      <p>3. 学完以上两门基础课后，在深入学习 JavaScript 的变量作用域、事件、对象、运动、cookie、正则表达式、ajax</p>
    </div>
    <form>
      <!--当点击相应按钮，执行相应操作，为按钮添加相应事件-->
      <input type="button" value=" 改变颜色 " onclick="changeColor()">
      <input type="button" value=" 改变宽高 " onclick="changeSize()">
      <input type="button" value=" 隐藏内容 " onclick="hideText()">
      <input type="button" value=" 显示内容 " onclick="showText()">
      <input type="button" value=" 取消设置 " onclick="reSet()">
    </form>
    <script type="text/javascript">
      var mycon = document.getElementById("con");
      var mytxt = document.getElementById("txt");

      //定义" 改变颜色 " 的函数
      function changeColor() {
        mycon.style.color="red";
        mycon.style.backgroundColor="#ccc";
        mytxt.style.color="red";
        mytxt.style.backgroundColor="#ccc";
      }
    </script>
  </body>
</html>
```

```

        //定义" 改变宽高 " 的函数
function changeSize() {
    //mycon.style.width="300px";
    //mycon.style.height="300px";
    mytxt.style.width="300px";
    mytxt.style.height="300px";
}

        //定义" 隐藏内容 " 的函数
function hideText() {
    mycon.style.display="none";
    mytxt.style.display="none";
}

        //定义" 显示内容 " 的函数
function showText() {
    mycon.style.display="block";
    mytxt.style.display="block";
}

        //定义" 取消设置 " 的函数
function reSet() { // reset 是关键词
    var tmp = confirm(" 是否确定重置? ");
    if (tmp == true) {
        //mycon.style="none";
        //mytxt.style="none";
        mycon.removeAttribute('style');
        mytxt.removeAttribute('style');
    } else {
        alert("Wise Choice!");
    }
}
    </script>
</body>
</html>

```