# JavaScript介绍

1. **JavaScript语法规则与JAVA很像，只不过javaScript是弱类型的语言，定义变量时都是用var变量名的方式命名的 , java是强类型的语言 ，必须用对应得类型声明变量，**
2. 相比HTML而言，JavaScript是动态脚本语言，把HTML静态页面转变成支持用户交互并响应相应时间的活页面。
3. Javascript和HTML关系密切，他们都是由浏览器来处理，HTML是通过浏览器的组网库和图形子系统来处理和展示，而Javascript程序是通过浏览器中内置Javascript解释执行的。
4. 浏览器处理WEB页面的一般过程为：浏览器请求一个WEB页面，WEB服务器则把整个页面的内容发送给浏览器，包括HTML和Javascript语句。之后浏览器加载WEB页面，加载的过程中，浏览器读取WEB页面的HTML语句，并逐行翻译，直到遇到Javascript代码中的开始标记时，Javascript解释器接管页面的处理过程并继续进行处理，到达Javascript代码中的结束标记时，Javascript解释器把将处理控制权交付出去，浏览器继续处理HTML内容。
5. 由浏览器处理的Javascript称作为客户端Javascript。虽然Javascript主要用于客户端脚本编程，但这并不是 Javascript的唯一用法，它也能够用户除WEB浏览器之外的其他环境，比如创建一种用于在服务器上编程的Javascript，则它可以看成是像ASP一样的服务端脚本语言。但是这里不讨论服务器端的，而把精力放在Javascript 运用最为广泛的客户端上。
6. Javascript不是JAVA。他们是由两家不同的公司开发出来的，Javascript是由Netscape公司开发，而JAVA则是 SunMicrosystem公司开发的；JAVA可以说是与WEB页面无关，而Javascript程序则嵌入在WEB页面中，并且必须在浏览器中运行；JAVA程序必须要经过编译器的编译(如J2SDK)后才可以运行，而Javascript 程序则由内置在浏览器中的Javascript引擎解释执行，不需要进行编译。
7. Javascript程序用于检测用户的活动，并对用户的操作做出相应的响应。通过Javascript程序，可以控制WEB页面的外观，也能够检测用户数据数据的有效性，而这些操作都不需要和服务器进行交互，这样就减轻了服务器的处理数据负担。
8. 总的来说Javascript能够完成以下的功能：

1）控制文档的外观和内容

2）控制浏览器的行为

3）与用户交互使用Cookies读写客户端状态

4）操作嵌入图象

# 二、编写与执行

1. 事件：事件是Javascript是让页面活起来的驱动力，所有的Javascript都是对某个事件做出的操作和反应。

|  |  |  |
| --- | --- | --- |
| **事件处理程序** | **HTML标记** | **触发时机** |
| onAbort | <img> | 图像加载过程给中断了时触发 |
| onBlur | <body>和窗体元素 | 窗体和窗体元素 窗体在失去键盘焦点时触发 |
| onChange | 窗体元素 | 用户修改了窗体元素的值后触发 |
| onClick | 所有元素 | 用户单击了类似按钮这样的窗体元素后触发 |
| onError | <img> | 加载图像过程中发生错误时触发 |
| onFocus | <body> | 窗体或窗体元素得到键盘焦点时触发 |
| onLoad | <body><img><object> | 文档、图像或对象完成加载时触发 |
| onMouseOut | 所有元素 | 鼠标指针离开对象时触发 |
| onMouseOver | 所有元素 | 鼠标指针移过对象时触发 |
| onSubmit | <form> | 用户提交窗体时触发 |
| onUnLoad | <body>,<frameset> | 卸载文档时触发 |

**例**:

<html>

<head><title>事件处理</title></head>

<body>

<form>

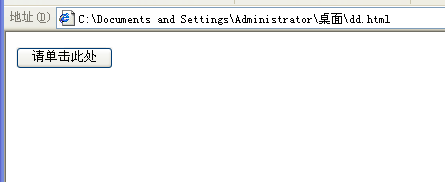
<input type="button" value="请单击此处" onClick="alert('单击了按钮，触发了 Click 事件')">

</form>

</body>

</html>

使用<form>标记和桑属性创建了一个窗体。Type属性指定了输入框的类型为按钮；value属性指定了这个按钮的值为"请单击此处"，这个值也是按钮的标题；当我们在浏览器窗口中单击这个按钮时，就会触发一个叫Click的Javascript事件，Click的意思是"单击"。使用onClick这个符号代表发生单击事件时执行的Javascript代码，也可以叫做事件处理程序。这里给onClick这个事件处理指定了一条语句，用户单击按钮后，就会执行这条语句。在这个示例中这条语句的功能是在一个消息框中显示"你刚刚单击了按钮，触发了Click事件！"。



1. 初识：客户端Javascript程序是嵌入在HTML文档中头标记<head>和</head>或体标记<body>和</body>之间。

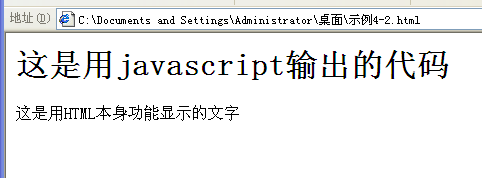
<script>标记作为程序的开始，</script>标记作为程序的结束。

注释：//，一行注释，而如果要有一次注释多行的话，那就可以用注释符/\*……..\*/。

**示例（在<body>标记中定义Javascript程序）**：这个例子的功能是在文档中输出一行文本，并

与HTML文档中定义的静态文本对比。

**例**：

<html>

<head><title>javascript编程示例</title></head>

<body>

<script language="javascript" type="text/javascript">

// javascript 代码使用的声明处

document.writeln("<h1>这是用 javascript 输出的代码</h1>");

//利用document对象的writeln方法往页面上输出"javascript输出的代码"这段文字

</script>

这是用HTML本身功能显示的文字

</body>

</html>

如果Javascript程序很长很复杂，或者想让某个Javascript程序重复使用，那就把这样的程序放到外部文件中，后缀名为.js，将外部文件名赋值给<script>标记的src属性，将这个js文件加载到WEB页面中。Javascript代码的外部文件可以是保存在本地，此时外部文件名可以使用全路径或者部分路径；同时外部文件也可以保存在另外一台机器上，此时赋值给src属性的文件名就必须是完整的 URL 名称，我们的 js 挂马技术就是用到这里的知识。

例（js文件名字为4-3.js，文件的的代码是：*document.writeln("<h1>这是用javascript引用外部js文件后输出的代码</h1>");*，而引用这个js文件的代码为：*<script language= "javascript" src= " C:\Documents and Settings\Administrator\桌面\4-3.js"></script>*）：

**例**：

<html>

<head><title>引用外部JS文件</title>

<script language= "javas cript" src= " C:\Documents a nd S ettings\Administrator\桌面\4-3.js"></script>

</head>

<body>

<script language="javascript" type="text/javascript">

document.writeln("这是不引用外部JS文件输出的javascript代码");

</script>

</body>

</html>

# 三、基本语法规则

1. 区分大小写
2. ; 结束符
3. + 连接符
4. 格式：

<script>

一条或多条Javascript语句

</script>

1. 属性：使用属性修饰<script>标记
   1. language属性指定程序代码使用哪一种脚本语言，常用的值为JavaScript、VBScript、Jscript。设置language 属性的一般格式为：*<script langrage="JavaScript">*
   2. type属性用于同时指定脚本语言和Internet内容类型。使用type属性指定脚本语言为JavaScript的常用语句格式为：*<script ty pe="text/javascript"></script>*
   3. 当要执行的脚本代码存放在某个外部文件中时，使用src属性指定该文件，文件的扩展名通常为.js，这样的文件称作 JavaScript 脚本文件。在文件名称中，可以包含路径或者 URL。

*<script langrage="JavaScript" type="text/javascript" src= "haha.js"></script>*

*<script langrage="JavaScript" type="text/javascript" src= http://xxxx.com/haha.js></script>*

JavaScript脚本文件保存Javascript代码，其中不需要编写<script>、</script>标记，也不需要任何HTML标记。使用src的属性的行为与直接把代码嵌套在<script>、</script>标记之间的行为是完全相同，这就是js挂马技术的原理，利用它就可以挂马。

# 四、动态生成页面内容

字符串拼接

Javascript规定，使用单引号或双引号把字符串文字括起来，这样来分辨出其他的Javascript语言元素。

“a”+”b”就相当于”ab”

write()和writeln()方法

document对象表示当前加载到浏览器中的HTML文档。

write()和 writeln()方法是 document 对象的两个方法，功能是把文本添加到浏览器中要显示文本的当前位置。write()和writeln()方法的语法格式为：

*document.write(value,…………)*

*document.writeln(value,………*)

从上面括号里的参数可以看出，这两个方法可以同时输出多个值。例如：

document.write("湖南省"，"长沙市") //在浏览器中添加输出"湖南省长沙市"

writeln()方法的功能与write()方法相同，他们的唯一区别是writeln()方法在输出文字的同时，还要输出一个换行符号，不过要使writeln()方法产生换行的效果还必须加上<pre>这个标记。当然还可以使用<br>标记也发生换行。

下面的示例演示了writeln()和write()方法的显示效果差异：

**例**：

<html>

<head><title> writeln()和write()及<pre>标记显示效果的差异</title></head>

<body>

<script language=JavaScript>

document.write("湖南省");

document.write("长沙市");

document.write("芙蓉区");

</script>

<br>===================================<br>

//<br>标记为换行的作用

<script language=JavaScript>

document.writeln("湖南省");

document.writeln("长沙市");

document.writeln("芙蓉区");

</script>

<br>==================================<br>

**<pre>**

<script language=JavaScript>

document.writeln("湖南省");

document.writeln("长沙市");

document.writeln("芙蓉区");

</script>

**</pre>**

</body>

</html>

# 五、与用户交互的基本方法

显示信息对话框的alert()方法

alert()方法生成的是一个独立的小窗口，称作对话框，用它来显示一条信息和一个按纽。虽然不同的浏览器中的alert()对话框的外观不完全相同，但是功能是一样的。alert()方法的语法格式为：*alert(expression)；*。expression是一个字符串类型的参数，并在对话框中显示该参数的值。

注意：alert()方法不处理参数字符串中包含的HTML标记，如果在字符串参数中包含 HTML 标记，那么alert()方法直接作为字符串而显示这些标记，但是可以在alert()方法的参数中使用转义字符。

跨站就是经常是弹出一个对话框，利用的知识就是alert()方法，通常写入的代码是*<script>alert("cmd")</script>*。

**例**：

<html>

<head><title>alert()方法的使用示例</title></head>

<body>

<script>alert("alert方法")</script>

<script>alert("alert\n方法")</script>

<script>alert("可以用来跨站")</script>

</body>

</html>

在第二个alert()方法中使用了\n换行符，所以在弹出的对话框中就会有两行。

显示确认选择对话框的confirm()方法

这种风格的对话框中显示一个问号和两个按钮：“确定”和“取消”。当点击“确定”按钮时，confirm()方法返回true，否则就返回false。confirm()方法只需要一个参数，用于向用户提出YES/NO风格问题的回答。Confirm()方法的语法格式为：*confirm(string)*。其中string是要在该对话框中显示的提示信息。

**例**：

<html>

<head><title>confirm()方法的使用示例</title></head>

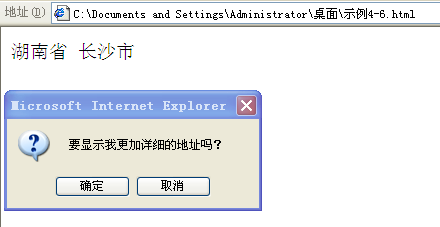
<body>

<script language="javascript">

document.writeln("湖南省 长沙市");

if(confirm("要显示我更加详细的地址吗？")==true){

document.writeln("芙蓉区 远大路");

alert("已经显示我的详细地址了");

}

else{

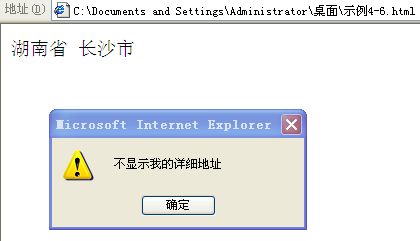
alert("不显示我的详细地址");

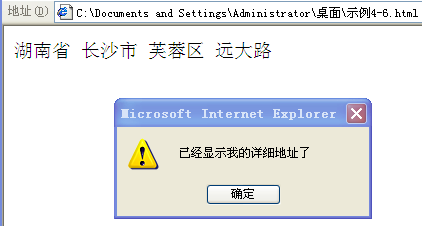
}

</script>

</body>

</html>

首先输出字符串"湖南省 长沙市"并弹出一个选择确认对话框，如果选择确定，那么confirm()方法返回true，和上面我们if…..else语句中的判断条件相符合，那么就会继续输出字符串"芙蓉区 远大路"，并弹出信息对话框，提示"已经显示我的详细地址了"，如果选择的是取消，那么就弹出信息对话框，提示"不显示我的详细地址"



显示提示对话框的 prompt()方法

使用prompt()方法显示提示对话框，在这个对话框中，可以显示一条由参数指定的消息，并等待用户在文本框中输入一段文字，同时这个对话框还显示了两个按钮：“确定”和“取消”。用户点击“确定”按钮后，返回用户在对话框的文本框中输入的文字，否则返回null，即返回空。

prompt()方法使用两个参数：第一个参数是一个字符串，指定在对话框中显示的消息；第二个参数也是一个字符串，指定在文本框中显示的初始值，如果省略了这个参数那么就会在文本框中显示字符串"undefined"。



prompt()方法的两种可用语法格式为：

*prompt(message);*

*prompt(message,defaultText);*

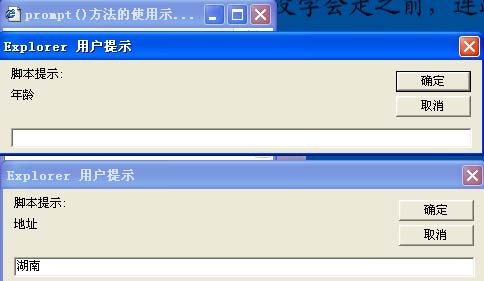
其中message是指定对话框中显示的消息，defaultText指定文本框中显示的初始内容。

**例**：

<head><title>prompt()方法的使用示例</title></head>

<body>

<script language="javascript">

prompt("年龄","")

prompt("地址","湖南")

</script>

</body>

</html>

# 六、变量和函数的声明与使用

变量

变量代表的是系统里一个存储区域。变量用于存储数据，变量一般有名称、类型及变量的值这三方面的特性。

变量的名称是以字符或下划线开头，后面跟一个或多个字母或数字，而且变量名称区分大小写。

**变量的声明：**

1. 用关键字var显式声明变量。
2. 使用赋值语句隐式地声明变量。在变量声明的同时可以给这个变量赋值，即变量的初始化。

变量的声明语句的一般格式为：

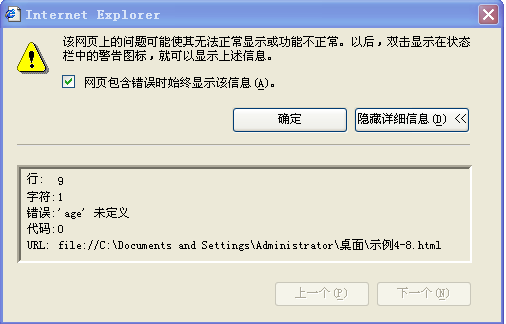
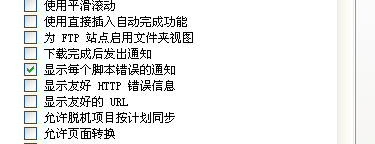
*var variable\_name=value; // 声明变量的同时初始化变量*

*var variable\_name=value; // 声明变量，但是没有初始化*

*variable\_name=value: // 在为变量赋值的同时声明变量*

var是关键字，variable\_name是变量名称，value是赋给变量的值，变量声明用逗号分割。

**例**：

<html>

<head>

<title>变量声明及用法</title>

<pre>

<script language="JavaScript">

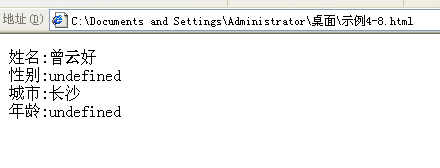
var name="LGY"; //声明并初始化变量

var sex; //声明但未初始化

city="长沙"; //赋值并初始化变量

age; //错误的声明方式

document.writeln("姓名:"+name);

document.writeln("性别:"+sex);

document.writeln("城市:"+city);

document.writeln("年龄:"+age);

</script>

</pre>

</head>

<body></body>

</html>

函数

函数是完成特定任务，并返回一个值的一组代码。

**参数：**

形式参数：它是在定义函数的时候为函数赋予的参数，代表了参数的位置和类型，所以系统操作系统并不会给形式参数分配实际的储藏空间，而是在调用函数时由实际参数代表形式参数参与函数的进行。

实际参数：它是在调用函数的时候传递给函数的参数，通常在调用函数前就已经给这些参数分配了内存，并且包含了实际的数据。所以在函数的执行过程中，实际参数参与了函数的运行。而在函数中定义的形式参数只是表明了调用函数时实际传递给函数的参数类型而已。

1. 定义

**定义函数时**，先写上关键字function、函数的名称，然后是一对小括号，括号内可以有一个或多个参数，也可以一个参数也没有。参数的作用是用于得到调用程序时传递给函数的信息。最后是用大括号括起来的一条或多条语句。

定义函数的**语法格式**有两种，分别为：

*function func\_name() {statement;}*

*function func\_name(para1,para2,.....paran) {statements;}*

function是定义函数的关键字；fun-name是函数的名称；para1,para2,…..paran是函数的形式参数；statements是函数完成函数功能的一条或者多条语句。

**例**：

Function hello() { document.write("欢迎大家学习脚本黑客技术");}

1. 调用

写上函数名称、小括号以及小括号中的实际参数（没有参数时可以不写）。例如调用上面的函数：

*hello()；*

调用函数的语法格式也有两种，分别为：

*func\_name()*

*func\_name(arg1,arg2,……argn)*

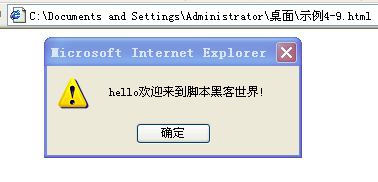
func-name是函数名称，与定义函数时使用的名称相同；arg1,arg2,……argn是传递给函数的实际参数，这里的实际参数的类型、个数、意义以及次序要与定义函数时的形式参数相同。

**Javascript调用函数的方法：**

**a）在Javascript代码中调用函数：**

函数在一段Javascript中定义，而在另一个Javascript程序中调用函数。一般要遵循先定义后调用的原则。

**例**：

<html>

<head>

<title>函数定义以及函数调用</title>

<script language=JavaScript>

function hello(){

var name="欢迎来到脚本黑客世界";

alert("hello"+name+"!");

}

</script>

</head>

<body>

<script language=JavaScript>

hello();

</script>

</body>

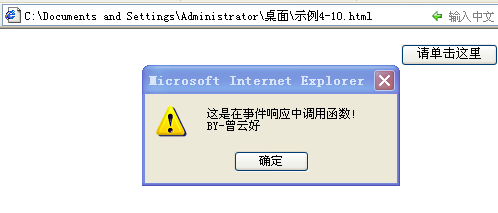
</html>

**b) 在事件响应中调用函数：**

通过编写程序对事件作出反应，这种反应称作响应事件。

<html>

<head>

<title>在事件响应中调用函数</title>

<script language=JavaScript>

function haha(){

alert("这是在事件响应中调用函数!\nBY-LGY");

}

</script>

</head>

<body><center>

<form>

<input type="button" value="请单击这里" onClick="haha();">

</form>

</body>

</html>

**c) 通过链接调用函数**

在<a>标记中，在关键字javascript:后面写上要调用函数的名称，用引号把这些内容引起来，然后赋值hre属性，例如：*<a href= "javascript:haha()">请单击这里</a>*。

**例**：

<html>

<head>

<title>通过链接调用函数</title>

<script language=JavaScript>

function xixi(){

alert("这是通过链接调用函数!\nBY-EDEGOD");

}

</script>

</head>

<body><center>

<form>

<a href= "javascript:xixi()">请单击这里</a></form>

</body>

</html>

1. 返回值

return返回值，return只能出现在函数中，程序在执行函数的过程中，遇到了 return 语句，那么就不再执行 return 语句后面的语句，而将值返回给调用函数程序。

如果在函数中没有return 语句，那么Javascript 会隐含地在函数末尾添加一条返回undefined值的return语句，因此所有的函数都是有返回值的，只不过在没有显示使用return语句的函数中，系统默认添加了一条返回undefined值的return语句，所以其实所有的函数都有返回值的。

return语句有两种形式的**语法格式**，分别为：

*return;*

*return expression;*

expression是一个表达式，它的值作为函数的返回值。例如下面的函数返回两个数差值：*function maxnum (a,b){ return (a-b);}*。

**例**：

<html>

<head>

<title>return的用法</title>

<script language=JavaScript>

function haha(a){

var b=a\*365;

return b;

}

</script>

</head>

<body>

<script language="JavaScript">

var a=prompt("请输入您的岁数！","");

var tody=haha(a);

alert("你出生了"+tody+"天");

</script>

</body>

</html>

在此例中先用prompt方法输入一个岁数，然后调用函数haha计算，最后通过alert返回当前一共出生了多少天，比如我输入我的岁数20，最后弹出了我出生的天数为7300。

# 七、 对象

数据和处理数据的方法结合在一起就构成了对象。

对象包括属性和方法两个特点：①属性是用于描述对象的特征 ②方法表示对象能够干什么

创建和使用自定义对象

构造函数创建对象：如Object()、Array()、Date()等，构造函数是一种用于创建对象实例的特殊方法，它具备了创建对象并初始化对象的功能，正是由于这种构造对象的特性，才把这种类型的特殊函数称为构造函数。

语法格式为：*var myNewObject = new Object(para1,para2………);*。

示例：

var cat = new Object();

var MyStudent = new Array("曾云好", "土豆");

对象的属性是描述对象的特征，它实际上是对象的成员变量。使用点运算符(.)将属性与所属对象关联起来，即在对象名后面写上点运算符，然后写上对象的属性名，例如一台电脑的颜色为红，那么就可以写成Computer.color=Red;。

如果对象的属性也是一个对象，那么在创建属性的时候就要用new关键字和Object()构造函数创建一个新的对象，并把该新对象的引用赋值给作为属性使用的子对象名称。

**例**：

<html>

<head>

<title>创建和使用对象的属性</title></head>

<body>

<script language=JavaScript>

var vehicle=new Object();

vehicle.car=new Object();

vehicle.car.name="奔驰";

vehicle.car.speed="300公里/小时";

vehicle.car.color="red";

document.write("小汽车的名字："+vehicle.car.name);

document.write("<br>奔驰的速度为："+vehicle.car.speed);

document.write("<br>奔驰的颜色："+vehicle.car.color);

</script>

</body>

</html>

上面定义的是一个交通工具的对象vehicle。而小汽车car是交通工具对象的一个属性。但是同时小汽车car也是一个对象，所以在使用car对象的时候就要用声明对象一样声明这个属性。而同时小汽车car还是有自己的属性，如颜色、速度等等。

除了可以使用Object()等构造函数来创建自定义对象之外，还可以通过函数来创建对象。这个时候的函数被称为对象的模板。在定义函数之后，可以使用new关键字和函数名称一起创建新的对象，并把它赋值给变量。这个时候定义的函数实际上就是一个构造函数。在定义函数时，使用this关键字来代表函数将来所创建的对象。

**例**：

<html>

<head>

<title>创建和使用对象的属性</title></head>

<script language="javascript">

function phone(name,phoneno,site){

this.name=name;

this.phoneno=phoneno;

this.site=site;

}

</script>

<body>

<script language="javascript">

var myphone=new phone("曾云好","0731—123456789","湖南省长沙市");

document.write

("<br>"+myphone.name+"<br>"+myphone.phoneno+"<br>"+myphone.site);

</script>

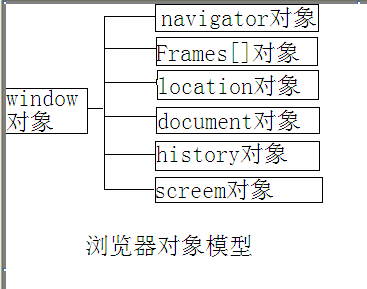
</body>

</html>

先定义了一个phone函数，而且在函数里的属性都是使用的this关键字。而在后面使用关键字new和函数名来创建一个使用这个函数的对象，并把结果输出。

浏览器对象模型

浏览器创建的对象我们通常称作文档(Document)对象但是它只是浏览器使用众多对象中的一部分。浏览器操作的各种对象组合起来我们叫浏览器对象模型(Browser ObjectModel)，简称BOM。



从图中可以看出window对象是顶层对象它包含了相关显示文档的窗口信息，例如窗口中包含的文档、框架、历史、位置等。

① navigator对象包含了浏览器的信息，例如浏览器的名称、版本、运行平台、是否支持cookie等

② Frames数组用于引用一组窗口对象

③ location对象存储了当前页面的URL地址

④ document对象即为文档对象，是浏览器对象模型中最重要的对象，也是我们在编程的时候使用最多的对象

⑤ history对象记录了当前会话中用户访问了哪些WEB页面

⑥ screen对象提供了当前计算机的显示特性，例如屏幕的宽度、高度、颜色深度等等。

**navigator对象（重点，其他略过）**

通过Javascript的内置对象navigator可以知道当前使用的浏览器是那种类型、哪种版本、是否支持Java、当前浏览器都有哪些插件(Plug-ins)等信息，从而让程序能够对特殊的浏览器做出相应的处理。

下表为常用的navigator对象的属性：

|  |  |
| --- | --- |
| 属性 | 意义 |
| appCodeName | 浏览器的代码名称，只读属性。Netscape和IE的该属性都是"Mozilla" |
| appName | 浏览器的实际名称，只读属性。 |
| cookieEnable | 说明浏览器是否支持cookies，只读属性。 |
| Platform | 给出运行浏览器的操作系统或硬件类型。 |
| systemLanguage | 给出操作系统的缺省语言，只读属性。 |
| userAgent | 给出浏览器在HTTP请求中使用的用户代理首部的值。 |

navigator对象的方法，常用的有javaEnable()，用来判断当前浏览器是否对Java支持。

**例**：

<html>

<head>

<title>检测浏览器及其特性</title></head>

<body>

<h3>检测浏览器的名称、版本号、HTTP用户代理首部的值<h3>

<script language="javascript">

document.write("<b>浏览器名称：</b>"+navigator.appName+"<br>");

document.write("<b>浏览器代码名称：</b>"+navigator.appCodeName+"<br>");

document.write("<b>浏览器是否打开cookie：</b>"+navigator.cookieEnable+"<br>");

document.write("<b>运行平台：</b>"+navigator.platform+"<br>");

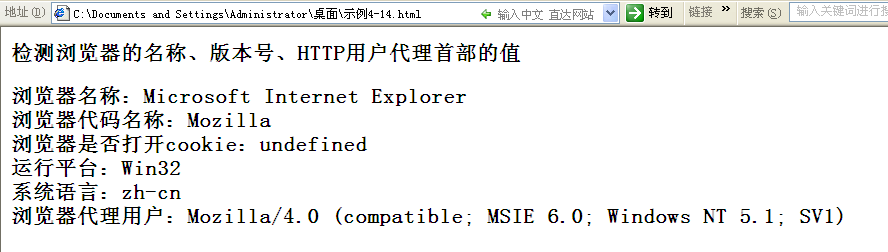
document.write("<b>系统语言：</b>"+navigator.systemLanguage+"<br>");

document.write("<b>浏览器代理用户：</b>"+navigator.userAgent+"<br>");

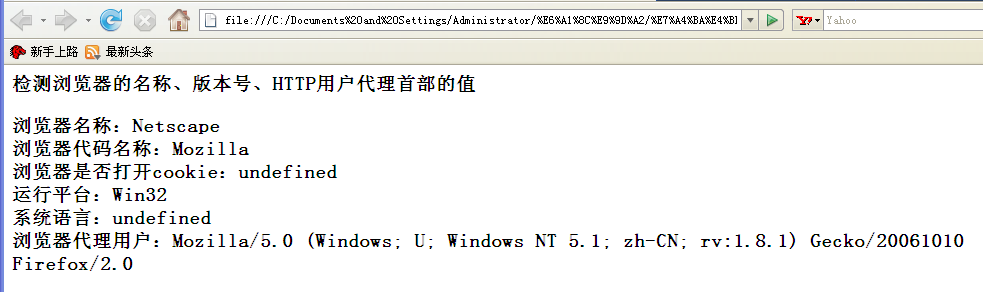
</script>

</body>

</html>



IE浏览器



火狐浏览器

窗口是Javascript浏览器对象模型中的顶层对象，称作窗口对象。它代表了打开的浏览器窗口。窗口对象是其他所有对象的基础，它也有自己的属性和方法，用于控制窗口特征和行为。Javascript允许省略窗口对象的名称，即省略window这个名称，如正规的写法应该是window.alert("hello 曾云好！")，但允许使用alert("hello 曾云好！")写法。

**窗口的属性**

|  |  |
| --- | --- |
| 属性 | 意义 |
| closed | 指明窗口是否已经被关闭，true表示关闭，false表示未关闭，只读属性。 |
| document | 只读，为当前窗口或框架包含的Document对象的引用。 |
| name | 窗口的名称，窗口名称可以通过window.open()方法指定。 |
| screenX,screenY | 整数型只读值，指定窗口左上角在屏幕上的X，Y坐标。 |

**窗口的方法**

|  |  |
| --- | --- |
| 方法 | 意义 |
| alert(message) | 在对话框中显示消息message，无返回值。 |
| blur() | 从窗口中移去键盘输入点，无返回值。 |
| Close() | 关闭窗口，无返回值。 |
| moveBy(dx,dy) | 将窗口从当前位置移动到指定的距离。 |
| moveTo(x,y) | 将窗口移动到指定的位置 |
| Open(url,name,features) | 在name参数指定的窗口中显示url指定的文档。 |
| setInterval(code,interval,args…) | 每隔interval指定的毫秒数就执行一次code字符串指定的代码。 |
| setTimeout(code,delay) | 在延迟delay参数指定的毫秒数后指定code字符串指定的代码。 |

其中alert()和open()是最频繁的。

**open()方法：**

除了可以在浏览器中通过执行"文件"——>"新建"——>"窗口"菜单命令打开一个新窗口外,也可以通过代码来打开一个新窗口,此时,使用到window对象的open方法。

语法格式为: *var window\_object=window.open("url",windowname,[features]);*

例如：*var winobj=window.open("http://www.google.com","google");*

**例**：

<html>

<head>

<title>open方法的简单应用</title>

<script language="JavaScript">

function openNewwindow(){

var winobj=window.open("http://www.google.com","google");

}

</script>

</head>

<body>

<h3>open方法的简单应用:点击即可打开google</h3>

<a href="javascript:openNewwindow()">点击即可打开google</a>

</body>

</html>

这里用到了前面的函数调用方式，同时使用了链接触发函数，点击后即可打开google。

**Document对象（重点，其他略过）**

利用文档对象模型，使用Javascript操作页面的过程通常称为动态HTML，也称为DHTML或Dynamic HTML。文档对象模型不是Javascript的一部分，而是独立于编程语言和操作平台的一个编程接口，VBscript同样可以在IE浏览器中通过DOM操作页面的内容。

document对象是window的一个属性。document对象表示浏览器中显示的HTML文档，它是客户端Javascript编程最重要的对象之一。Document对象主要对应于文档主体部分——<body></body>标记之间的内容。

语法格式：

*window.document*

*docuemt*

**常用属性**：

|  |  |
| --- | --- |
| 属性 | 意义 |
| bgColor | 指定文档的背景颜色 |
| cookie | 允许读写HTTP的cookie，字符串类型 |
| fgcolor | 指定文档的前景颜色，字符串类型 |
| title | 文档标题，对应于文档中的<title>标记内容 |
| URL | 指定当前文档的URL，只读属性。 |

其实窃取cookie就是利用的document对象的cookie属性。我们只要在存在漏洞的网站输入<script>alert(document.cookie)</script>就可以窃取到cookie。

**document对象的方法**：

|  |  |
| --- | --- |
| 方法 | 意义 |
| open() | 删除现有文档的内容，打开一个新的文档输出流。 |
| close() | 关闭open方法打开的文档输出流 |
| focus() | 指定文档得到焦点 |
| write(value,….) | 在当前的文档中插入的字符串或在open()打开的文档末尾添加字符串 |
| writeln(value,….) | 与Write()方法先同，只是在输出字符串的末尾添加回车换行字符 |