第3章 JavaScript基础

建议学时：2

第2章中学习了HTML语言，通过HTML，可以利用标签描述一个网页。但是，标签式的描述语言限制了网页一些在客户端进行运算的功能。本章将学习JavaScript语言，JavaScript嵌入HTML页面内，是一种运行在客户端、由浏览器进行编译运行的脚本语言，具有控制程序流程的功能。本章将学习其基本语法及基本对象。

3.1 JavaScript简介

JavaScript是一种网页脚本语言，虽然名字中含有Java，但它与Java语言是两种完全不同的语言。不过，JavaScript的语法与Java语言的语法非常类似。

JavaScript代码可以很容易地嵌入到HTML页面中。浏览器对JavaScript脚本程序进行编译、运行。

3.1.1 第一个JavaScript程序

JavaScript代码可以嵌入HTML中，如下是一个简单的例子：

firstPage.html

```
<html>
    <body>
        <script type="text/javascript">
            window.alert("第一个JavaScript程序");<!-- 跳出消息框 -->
        </script>
    </body>
</html>
```

保存为HTML页面后，使用浏览器打开，将会跳出如下消息框：

注意,JavaScript 代码块“<script type="text/javascript">JavaScript 代码</script>”,除了像上面一样写在<body></body>之间之外,还可以写入<head></head>之间,其效果相同。

“<script type="text/javascript">JavaScript 代码</script>”也可以写为:“<script language="javascript">JavaScript 代码</script>”。

JavaScript 与 Java 一样,对大小写是敏感的。

在 JavaScript 中,注释有三种写法:一种是 HTML 注释的写法,“<!--注释内容-->”,还有两种和 Java 语言相同,分别为“//单行注释”和“/*多行注释*/”。

除了可以将 JavaScript 代码嵌入 HTML 中之外,还可以专门将 JavaScript 代码写在单独的文件中:

code.js

```
window.alert("第一个 JavaScript 程序");
```

然后在另外的 HTML 页面中插入:

```
<script src="code.js" type="text/javascript"></script>
```

来导入该文件。

此外,HTML 代码中可以写多个 JavaScript 代码块。

3.1.2 JavaScript 语法

1. 变量定义

JavaScript 中的变量为弱变量类型,即变量的类型根据它被赋值的类型改变,定义一个变量使用的格式为:“var 变量名”,比如定义变量 arg,就可以使用 var arg;如果将一个字符串赋给它,它就是 String 类型,如果将一个数组赋给它,它的类型也就是数组类型。

var.html

```
<html>
    <body>
        <script type="text/javascript">
            var arg1,arg2,arg3;       <!--定义三个变量-->
            var arg4=5;               <!--定义一个整型(Integer)变量-->
            var arg5=10.0;            <!--定义一个浮点型(Float)变量-->
            var arg6="你好!";         <!--定义字符型(String)变量-->
            var arg7=true;            <!--定义一个布尔类型(Boolean)变量-->
            var arg8=new Array("王","李","赵","张");   <!--定义字符串数组-->
        </script>
    </body>
</html>
```

需要注意的是,JavaScript 中变量未声明就使用是不会报错的,但很容易出现不可预知的错误,所以建议所有变量先声明后使用。

另外,函数“Number(字符串)”可以将字符串转为数值;函数“String(数值)”可以将数值转换为字符串。

2. 函数定义

JavaScript 中定义一个函数的基本格式是：

```
function 函数名(参数列表){
    return 值;
}
```

也可以在使用中直接匿名定义：

```
var arg1 = function(参数列表){
    return 值;
}
```

看下面一段代码：

fun.html

```
<html>
    <body>
        <script type="text/javascript">
            var arg0 = "欢迎使用 JavaScript";
            print(arg0);
            function print(arg1){
                window.alert(arg1);
            }
        </script>
    </body>
</html>
```

图 3-1 页面运行效果

运行结果如图 3-1 所示。

实际上，JavaScript 的语法与 Java 的语法基本类似，所以不作详细讲述。以上介绍的几个知识点，都是 JavaScript 与 Java 有差别的语法，其他的常用语句与 Java 类似，比如 if 判断语句，在 JavaScript 是这样写的：

```
<html>
    <body>
        <script type="text/javascript">
            var score = 67;
            if(score >= 60){
                window.alert("及格");
            }else{
                window.alert("不及格");
            }
        </script>
    </body>
</html>
```

又如 for 循环：

```
<html>
    <body>
```

```
        <script type="text/javascript">
            for(var i=1;i<10;i++){
                window.alert(i);
            }
        </script>
    </body>
</html>
```

以上的写法与 Java 是一样的。

下面用循环举一个实际的例子。编写一个恶意程序，用户打开，会不断跳出消息框。代码如下：

恶意网页.html

```
<html>
    <body>
        <script language="javascript">
            str = new Array("你受骗了","你真的受骗了","真笨啊");
            while(true){
                for(i=0;i<str.length;i++){
                    window.alert(str[i]);
                }
            }
        </script>
    </body>
</html>
```

该代码运用了 JavaScript 中的循环，使得消息框怎么单击都不会结束，而且无法关掉浏览器，只能通过关闭进程结束。读者可以进行实验。

3.2 JavaScript 内置对象

除了在代码里面进行简单的编程之外，我们还可以通过 JavaScript 提供的内置对象来对网页进行操作，内置对象由浏览器提供，可以直接使用，不用事先定义。例如，在 3.1.1 小节中的 window.alert("第一个 JavaScript 程序")，其中 window 就是一个内置对象。

使用最多的内置对象有 4 个，并且之后的学习也将主要围绕着 4 个对象展开。

(1) window：负责操作浏览器窗口，负责窗口状态，开闭等。

(2) document：负责操作浏览器载入的文档(HTML 文件)。它从属于 window。

(3) history：可以代替后退(前进)按钮访问历史记录，从属于 window。

(4) location：访问地址栏，也是从属于 window。

注意，如果一个对象从属于另一个对象，使用时用“.”隔开，如：“window.document.×××”，但是，如果从属于 window 对象，window 可以省略。如：“window.document.×××”可以写为“document.×××”。

3.2.1 window 对象

window 对象的作用有如下几个。

1. 出现提示框

window 对象可以跳出提示框。主要有如下功能。

window.alert("内容")：出现消息框。

window.confirm("内容")：出现确认框。

window.prompt("内容")：出现输入框。

下面代码的功能为显示一些提示框：

window1.html

```
<html>
    <body>
        <script type="text/javascript">
            //1:消息框
            window.alert("消息框");
            //2:确认框,根据 result 的值 true 或者 false 来判断
            result = window.confirm("您确认提交吗?");
            //3: 输入框,str 为输入的值,如点取消,str 的值为 null
            str = window.prompt("请您输入一个字符串","");
        </script>
    </body>
</html>
```

用浏览器打开该文件，将会依次出现以下提示框，如图 3-2 所示。

图 3-2　提示框运行效果

浏览器跳出提示框后，载入页面将会停滞，直到用户做出操作动作。其中消息框在实际运用中最为广泛，确认框其次，输入框则较为少见。

2. 打开、关闭窗口

window 对象还用于控制窗口状态和开闭。窗口打开主要使用 window 的 open 函数。看下面的一段代码：

window2.html

```
<html>
    <body>
        <script type="text/javascript">
            window.status = "出现新窗口";
            //打开新窗口
            newWindow = window.open("window1.html","new1",
                    "width=300,height=300,top=500,left=500");
            //可以通过返回值来控制新窗口
```

```
            //newWindow.close(); //关闭窗口
        </script>
    </body>
</html>
```

本例中，首先让本窗口状态栏变为字符串“出现新窗口”，然后打开一个新窗口“window1.html”，命名为 new1，指定宽度、高度和其位置。运行的效果如图 3-3 所示。

源程序中，“newWindow.close();”表示关闭 newWindow。

window 的 status 属性显示在浏览器的左下角状态栏中，如图 3-4 所示。

图 3-3 打开窗口运行效果

图 3-4 状态栏

综上所述，“window.open()”在网页制作中使用非常广泛，参数有三个，第一个是新窗口的地址，第二个是新窗口的名称，第三个是新窗口的状态，其中新窗口可设置的状态的属性如下。

toolbar：是否有工具栏，可选 1 和 0。

location：是否有地址栏，可选 1 和 0。

status：是否有状态栏，可选 1 和 0。

menubar：是否有菜单条，可选 1 和 0。

scrollbars：是否有滚动条，可选 1 和 0。

resizable：是否可改变大小，可选 1 和 0。

width，height：窗口的宽度和高度，用像素表示。

left，top：窗口左上角相对于桌面左上角的 x 和 y 坐标。

各属性值用逗号隔开。如：

```
newWindow = window.open("window1.html","new1",
      "toolbar=0,width=300,height=300,top=500,left=500");
```

3. 定时器

window 对象负责管理和控制页面的定时器，定时器的作用是让某个函数隔一段时间之后运行一次，格式为：

```
timer = window.setTimeout("需要运行的函数","时间(用毫秒计)");
```

如果要清除定时器，则可以：

```
clearTimeout(timer);
```

下面来看一段代码：

timer.html

```
<html>
    <body>
```

```
        <script type="text/javascript">
            //setTimeout 让函数某段时间之后运行 1 次,参数 2 是毫秒
            timer = window.setTimeout("fun1()","1000");
            var i = 0;
            function fun1(){
                i++;
                window.status = i;
                if(i == 100){
                    window.clearTimeout(timer);     //清除定时器,否则会一直运行
                    return;
                }
                timer = window.setTimeout("fun1()","1000");
            }
        </script>
    </body>
</html>
```

图 3-5　定时器运行效果

运行结果如图 3-5 所示。

在状态栏中,每隔 1 秒钟,状态栏中的数字将会加 1,直到 100,此时将一直持续 100 的状态,不再改变。

设置定时器可以使得网页定时刷新,在一些要求计时功能的网页中经常被用到。

3.2.2　history 对象

history 对象包含用户的浏览历史等信息,用到这个对象,是因为它可以代替后退(前进)按钮访问历史记录,该对象从属于 window。

history 最常用的函数如下。

history.back():返回上一页,相当于单击了浏览器上的“后退”按钮。

history.forward():返回下一页,相当于单击了浏览器上的“前进”按钮。

window.history.go(n):n 为整数,正数表示向前进 n 格页面,负数表示向后退 n 格页面。下面来看一段代码:

history.html

```
<html>
    <body>
        <a onclick="history.forward()">前进</a>
        <a onclick="history.back()">后退</a>
    </body>
</html>
```

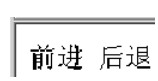

图 3-6　history.html 运行效果

运行“history.html”,结果如图 3-6 所示。

单击“前进”或者“后退”按钮,其效果和单击了浏览器上的按钮一样。

注意此处用到了网页元素的事件,由于篇幅所限,本章仅仅用到单击事件(onclick),其他事件,读者可以参考相应文档。

3.2.3 document 对象

document 对象从属于 window，其功能如下。

1. 在网页上输出

在网页输出方面，最常见的函数是 writeln()，下面看一段代码：

document1.html

```
<html>
    <body>
        <script type="text/javascript">
            document.writeln("你好");
        </script>
    </body>
</html>
```

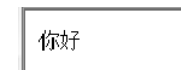

图 3-7 document1.html 运行效果

运行效果如图 3-7 所示。

writeln()函数为输出一些简单却重复的代码提供很大的便利，在下面一个例子中，将要使用表格显示出一个 8×8 的国际象棋棋盘，正常的方法需要写一个 8 行 8 列的表格代码，那样会使源代码非常冗长。下面是用 writeln()函数的做法：

chess.html

```
<html>
    <body>
        <script type="text/javascript">
            document.writeln("<table width=400 height=400 border=1>");
            for(i=1;i<=8;i++){
                document.writeln("<tr>");
                for(j=1;j<=8;j++){
                    color = "black";
                    if((i+j)%2==0){
                        color = "white";
                    }
                    document.writeln("<td bgcolor=" + color + "></td>");
                }
                document.writeln("</tr>");
            }
            document.writeln("</table>");
        </script>
    </body>
</html>
```

借助 writeln()和循环方法，省去了很多 HTML 代码的编写。运行结果如图 3-8 所示。

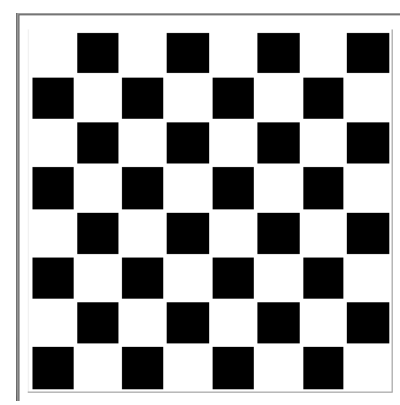

图 3-8 棋盘运行效果

2. 设置网页的属性

document 可以进行一些简单网页属性的设置，如

网页标题、颜色等，并且可以得到网页的某些属性，如当前地址。比较常用的有：通过“document.title”来访问标题，通过“document.location”来获取网页当前的地址等。下面来看一段代码：

document2.html

```
<html>
    <body>
        <script type="text/javascript">
            function fun(){
                document.title = "新的标题"; //设置网页标题
                window.alert(document.location);//得到当前网页的地址
            }
        </script>
        <input type="button" onclick="fun()" value="按钮">
    </body>
</html>
```

运行后，单击按钮，将会跳出一个消息框，内容为当前页面的地址，并且网页的标题也将改变为“新的标题”。关于其他功能，读者可以参考相应文档。

3. 访问文档元素，特别是表单元素

document 可以访问文档中的元素（如图片、表单、表单中的控件等），前提是元素的 name 属性是确定的。访问方法为：“document.元素名.子元素名...”。比如，名为 form1 的表单中有一个文本框 account，其中的内容可以用如下代码获得：

```
var account = document.form1.account.value;
```

下面的例子中，有两个文本框，一个按钮，输入两个数字单击按钮，显示两个数字的和。代码如下：

document3.html

```
<html>
    <body>
        <script type="text/javascript">
            function add(){
                //得到这两个文本框的内容
                n1 = Number(document.form1.txt1.value);
                n2 = Number(document.form1.txt2.value);
                document.form1.txt3.value = n1 + n2;
            }
        </script>
        <form name="form1">
            <input name="txt1" type="text"><BR>
            <input name="txt2"  type="text"><BR>
            <input type="button" onclick="add()" value="求和"><BR>
            <input name="txt3" type="text"><BR>
        </form>
    </body>
</html>
```

运行后，文本框为空，在第一行和第二行文本框填入数字后，单击“求和”按钮，结果如图 3-9 所示。

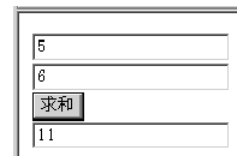

图 3-9 求和效果

由于 document 可以得到网页中的元素的值，所以在客户端的验证中用得非常广，比如在注册或登录中，可以使用 JavaScript 得到表单中的值，然后通过判断，做出相应的反应。下面举的一个例子就是通过 JavaScript 判断表单中的值是否为空。

validate.html

```
<html>
    <body>
        <script type="text/javascript">
            function validate(){
                //得到这两个文本框的内容
                account = document.loginForm.account.value;
                password = document.loginForm.password.value;
                if(account == ""){
                    alert("账号不能为空");
                    document.loginForm.account.focus();//聚焦
                    return;
                }
                else if(password == ""){
                    alert("密码不能为空");
                    document.loginForm.password.focus();
                    return;
                }
                document.loginForm.submit();
            }
        </script>
        欢迎您登录:
        <form name="loginForm">
            输入账号:<input name="account" type="text"><BR>
            输入密码:<input name="password"  type="password"><BR>
            <input type="button" onclick="validate()" value="登录">
        </form>
    </body>
</html>
```

图 3-10 验证效果

特别提醒

“document.loginForm.account.focus();”为聚焦函数，使光标移动到调用这个函数的元素位置；“document.loginForm.submit();”为提交表单，与按下提交按钮的效果一样。

这样进行验证，可以减少服务器遭到恶意登录的可能。

运行网页，不输入账号就进行登录，效果如图 3-10 所示。

从上面的程序可以看出，当用户没有输入账号或密码就单击“登录”按钮时，将跳出提示填写账号或密码的信息框，直到都填写完整，表单才能提交。

3.2.4 location 对象

location 对象可以访问浏览器地址栏，也是从属于 window，最常见的功能就是跳转到另一个网页。跳转的方法是修改 location 的 href 属性，看下面的代码：

location1.html

```
<html>
    <body>
        <script type="text/javascript">
            function locationTest(){
                window.location.href = "image.jpg";
            }
        </script>
        <input type="button" onclick="locationTest()" value="按钮">
        <a href="image.jpg">到图片</a>
    </body>
</html>
```

运行结果如图 3-11 所示。

单击链接和单击按钮的效果是一样的，都会跳转到图片页面，如图 3-12 所示。

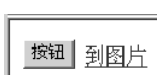

图 3-11 location1.html 运行效果

图 3-12 跳转目标页面

比较常见的另一个例子是定时跳转，可以结合 window 的定时器，代码如下：

location2.html

```
<html>
    <body>
        欢迎您登录,3 秒钟转到首页……
        <script type="text/javascript">
            window.setTimeout("toIndex()","3000");//在 3 秒钟后运行一次 toIndex()
            function toIndex(){
                window.location.href = "image.jpg";
            }
        </script>
```