第5章 JavaScript编程

5.1 JavaScript概述

5.1.1 JavaScript简介

HTML定义网页的内容，CSS定义网页的表现，JavaScript则定义特殊的行为。JavaScript是一种脚本语言，脚本语言的特点是简单、易学，即使是程序设计新手也可以非常容易地使用JavaScript进行简单的编程。

JavaScript作为一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器称为JavaScript引擎，为浏览器的一部分，广泛用作客户端的脚本语言，最早是在HTML网页上使用，用来给HTML网页增加动态效果。

其前身是LiveScript，JavaScript的正式名称是ECMAScript，由Netscape公司的Brendan Eich发明。从1996年开始，它就出现在所有的Netscape和Microsoft浏览器中。现在几乎所有浏览器都支持JavaScript，如Internet Explorer、Firefox、Netscape、Mozilla、Opera等。

使用JavaScript的目的是，与HTML和Java小程序一起实现在一个Web页面中链接多个对象，与Web客户进行交互作用，从而开发出客户端的应用程序。它可以嵌入HTML文件中，不需要经过Web服务器就可以对用户操作做出响应，使网页更好地与用户交互；在利用客户端个人电脑性能资源的同时可以减小服务器端的压力，并减少用户等待时间。JavaScript的出现弥补了HTML的缺陷。

1. JavaScript的特点

(1) JavaScript是一种脚本编程语言。它采用小程序段的方式实现编程。像其他脚本语言一样，JavaScript同样也是一种解释性语言，它提供了一个非常方便的开发过程。

(2) JavaScript 的语法基本结构形式与 C、C++、Java 的语法结构十分类似。但在使用前,不像这些语言要先编译,而是在程序运行过程中被逐行地解释。JavaScript 与 HTML 结合在一起,方便了用户的使用及操作。

(3) JavaScript 是一种基于对象的语言,也可以看作一种面向对象的语言。这意味着 JavaScript 能运用它已经创建的对象。因此,许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。

(4) JavaScript 简单。首先,JavaScript 是一种基于 Java 基本语句和控制流之上的简单而紧凑的设计,从而对于学习 Java 或 C/C++语言是一种非常好的过渡,而对于具有 C/C++语言编程功底的程序员来说,JavaScript 上手也非常容易。其次,其变量类型是采用弱类型,并未使用严格的数据类型。

(5) JavaScript 具有非常高的安全性。JavaScript 作为一种安全性语言,不允许访问本地的硬盘,且不能将数据存入服务器,不允许对网络文档进行修改和删除,只能通过浏览器实现信息浏览或动态交互。从而有效地防止数据的丢失或对系统的非法访问。

(6) JavaScript 是动态的,可以直接对用户或客户输入做出响应,无须经过 Web 服务程序。JavaScript 对用户的反映响应,是采用事件驱动方式进行的。在网页中执行了某种操作所产生的动作称为“事件”(Event)。例如,按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后,可能会引起相应的事件响应,执行某些对应的脚本,这种机制称为“事件驱动”。

(7) JavaScript 具有跨平台性。JavaScript 是依赖于浏览器本身,与操作环境无关,只要能运行浏览器的计算机并支持 JavaScript 的浏览器就可以正确执行。

① JavaScript 可直接写入 HTML 输出流。例如:

```
document.write("<h1>JavaScript 输出的标题。</h1>");
document.write("<p>JavaScript 输出段落。</p>");
```

注意:只能在 HTML 输出中使用 document.write。如果在文档加载后使用该方法,则会覆盖整个文档。

② JavaScript 可对事件反应。例如:

```
<button type="button" onclick="alert('欢迎来到物联网的世界!')">按钮!</button>
```

说明:alert 函数在 JavaScript 中并不常用,但它对于代码测试非常方便。

onclick 事件只是即将在本书学到的众多事件之一。

③ JavaScript 可改变 HTML 内容。使用 JavaScript 来处理 HTML 内容是非常强大的功能。例如:

```
x=document.getElementById("div")      //查找元素
x.innerHTML="Hello JavaScript";       //改变内容
```

JavaScript 开发中会经常看到 document.getElementById("some id")。这个方法是 HTML DOM 中定义的。DOM(Document Object Model,文档对象模型)是用于访问

HTML 元素的正式 W3C 标准。

2. JavaScript 的作用

(1) 校验用户输出的内容。
(2) 有效地组织网页内容。
(3) 动态地显示网页内容。
(4) 弥补静态网页不能实现的功能。
(5) 动画显示。

3. JavaScript 与 Java 的区别

JavaScript 与 Java 区别如表 5-1 所示。

表 5-1 JavaScript 与 Java 的区别

序 号	JavaScript	Java
1	在客户端运行时被解释	由编写者编译后变成机器代码,运行在服务器端或客户端
2	程序源代码嵌入在 HTML 文件中	由 Java 开发的 Applets 与 HTML 无关
3	弱数据类型	严格的数据类型
4	由美国 Netscape 公司的 Brenddan Eich 发明	由美国 Sun microsystems 公司的 James Gosling 发明
5	只能在浏览器中应用	可作为独立的应用程序
6	只作用于 HTML 的对象元素	只作用于 HTML 的对象元素外的元素,如多媒体

5.1.2 引入 JavaScript 文件

1. 添加嵌入脚本

嵌入脚本位于 HTML 文档之内,同嵌入样式表相似。HTML 中的脚本必须位于<script>与</script>标签之间。<script>和</script>会告诉 JavaScript 在何处开始和结束。脚本可被放置在 HTML 页面的<body>或<head>中。

一些老的示例可能会在<script>标签中使用 type="text/javascript"。现在已经不必这样做了。JavaScript 是所有现代浏览器以及 HTML5 中的默认脚本语言。

1) <head>中的 JavaScript 函数

例 5-1 把一个 JavaScript 函数放置到 HTML 页面的<head>部分,该函数会在单击按钮时被调用。

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function myFunction()
```

```
        {
            document.getElementById("example").innerHTML="我是图片 2";
        }
    </script>
</head>
<body>
    <p id="example">我是图片 1</p>
    <button type="button" onclick="myFunction()">修改</button>
</body>
</html>
```

页面显示效果如图 5-1 所示。

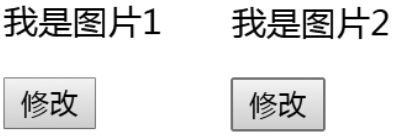

图 5-1 按钮调用函数页面显示

2）＜body＞中的 JavaScript 函数

例 5-2 把一个 JavaScript 函数放置到 HTML 页面的＜body＞部分，该函数会在单击按钮时被调用。

```
<!DOCTYPE html>
<html>
<body>
    <p id="example">我是图片 1</p>
    <button type="button" onclick="myFunction()">修改</button>
    <script>
        function myFunction()
        {
            document.getElementById("example").innerHTML="我是图片 2";
        }
    </script>
</body>
</html>
```

2. 从外部文件加载脚本

同为页面添加样式表一样，从外部文件加载脚本通常比在 HTML 中嵌入脚本要好一些。这样做的好处是，可以在需要某一脚本的每个页面加载同一个 JavaScript 文件。需要对脚本进行修改时，可以仅编辑一个脚本，而不必在各个单独的 HTML 页面更新相似的脚本。无论是加载外部脚本还是添加嵌入脚本，均使用 script(脚本)元素。

外部 JavaScript 文件的文件扩展名是.js。如需使用外部文件，请在＜script＞标签的 src 属性中设置该.js 文件。例如：

```
<!DOCTYPE html>
<html>
<body>
<script src="myScript.js"></script>
</body>
</html>
```

可以将脚本放置于<head>或<body>中，放在<script>标签中的脚本与外部引用的脚本运行效果完全一致。

myScript.js 文件代码如下。

```
function myFunction()
{
   document.getElementById("example").innerHTML="我是图片 2";
}
```

注意：外部脚本不能包含<script>标签。

5.1.3 JavaScript 语法基础

1. JavaScript 显示

JavaScript 不提供任何内建的打印或显示函数，但 JavaScript 能够以下面不同方式“显示”数据。

- 使用 innerHTML 写入 HTML 元素。
- 使用 document.write 写到 HTML 输出。
- 使用 window.alert 写入警告框。
- 使用 console.log 写入浏览器控制台。

1）使用 innerHTML 写入 HTML 元素

访问 HTML 元素，JavaScript 可使用 document.getElementById(id)方法。其中，id 属性定义 HTML 元素，innerHTML 属性定义 HTML 内容。

例 5-3 通过指定的 id 来访问 HTML 元素，并改变其内容。

```
<!DOCTYPE html>
<html>
<body>
    <h1>智能家居</h1>
    <p>安全防护功能</p>
    <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML="安全";
</script>
</body>
</html>
```

程序运行效果如图 5-2 所示。

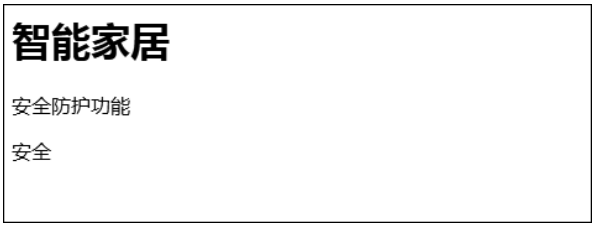

图 5-2 JavaScript 元素输出

JavaScript 由 Web 浏览器执行。在这种情况下,浏览器将访问 id="demo"的 HTML 元素,并把它的内容(innerHTML)替换为“安全”。更改 HTML 元素的 innerHTML 属性是在 HTML 中显示数据的常用方法。

2) 使用 document.write 写到 HTML 输出

例 5-4 直接把<p>元素写到 HTML 文档输出。

```
<!DOCTYPE html>
<html>
<body>
    <h1>智能家居</h1>
    <p>安全防护功能</p>
    <script>
        document.write("<p>安全</p>");
    </script>
</body>
</html>
```

页面运行效果如图 5-3 所示。

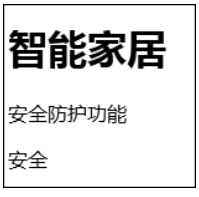

图 5-3 JavaScript 文档输出

注意:在 HTML 文档完全加载后使用 document.write 将删除所有已有的 HTML,所以请使用 document.write 仅仅向文档输出写内容。

例 5-5 在 HTML 文档完全加载后使用 document.write 将删除所有已有的 HTML。

```
<html>
<body>
    <h1>智能家居</h1>
    <p>安全防护功能</p>
    <button onclick="myFunction()">单击这里</button>
```

```
    <script>
    function myFunction()
    {
    document.write("我的 HTML 内容被覆盖");
    }
    </script>
</body>
</html>
```

页面运行效果如图 5-4 所示。

单击“单击这里”按钮后，出现如图 5-5 所示的效果。

图 5-4　文档输出调用前

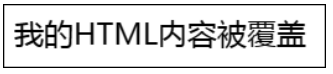

图 5-5　文档输出调用后

3）使用 window.alert 写入警告框

例 5-6　使用警告框来显示数据。

```
<html>
<body>
    <h1>智能家居</h1>
    <p>安全防护功能</p>
<script>
    window.alert("警告框");
</script>
</body>
</html>
```

显示效果如图 5-6 所示。

图 5-6　警告框显示数据

4）使用 console.log 写入浏览器控制台

在浏览器中可使用 console.log 方法来显示数据。

例 5-7 在谷歌浏览器(chrome)中，通过 F12 来激活浏览器控制台，并可在菜单中选择“控制台”查看数据。

```
<!DOCTYPE html>
<html>
<body>
    <h1>智能家居</h1>
    <p>安全防护功能</p>
<script>
    console.log("显示信息");
</script>
</body>
</html>
```

页面显示效果如图 5-7 所示。

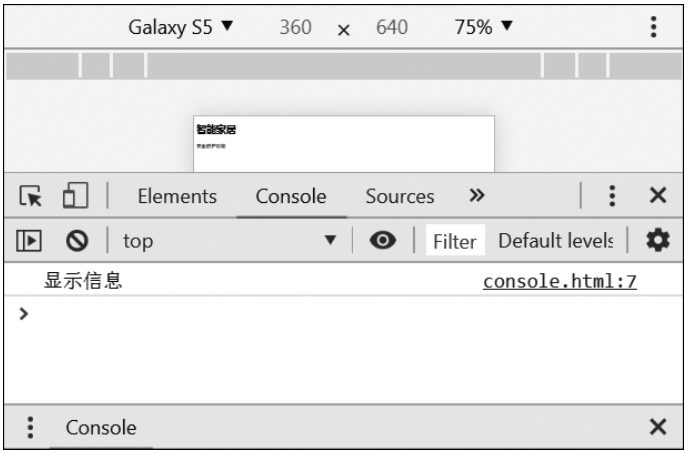

图 5-7 控制台查看数据

2. JavaScript 语句

1）JavaScript 程序

计算机程序是由计算机“执行”的一系列“指令”组成的。在 HTML 中，JavaScript 语句是由 Web 浏览器“执行”的“指令”组成的。在编程语言中，这些编程指令称为语句。JavaScript 程序就是由一系列的编程语句组成的。

说明：在 HTML 中，JavaScript 程序由 Web 浏览器执行。

2）JavaScript 语句

JavaScript 语句由值、运算符、表达式、关键词和注释构成。大多数 JavaScript 程序都包含许多 JavaScript 语句。这些语句会按照它们被编写的顺序逐一执行。例如：

```
document.getElementById("demo").innerHTML="查看火焰传感器状态";
```

说明：JavaScript 程序常被称为 JavaScript 代码。

3）分号

分号用于分隔 JavaScript 语句。通常在每条可执行的语句结尾添加分号。例如：

```
a=5;
b=6;
c=a+b;
```

使用分号的另一用处是在一行中编写多条语句。

说明：在 JavaScript 中，用分号来结束语句是可选的。

4）JavaScript 代码

JavaScript 代码是 JavaScript 语句的序列。浏览器会按照编写顺序来执行每条语句。例如，下面语句将操作两个 HTML 元素，效果如图 5-8 所示。

```
document.getElementById("demo").innerHTML="安防控制界面";
document.getElementById("mydiv").innerHTML="能耗管控界面";
```

图 5-8　JavaScript 语句使用

5）JavaScript 代码块

JavaScript 语句可以用花括号组合在代码块中。代码块的作用是定义一同执行的语句。JavaScript 函数是将语句组合在块中的典型例子。

下面的例子可操作代码块中的两个 HTML 元素的函数。代码块使用后显示如图 5-9 所示，JavaScript 代码改变元素如图 5-10 所示。

```
function myFunction()
{
document.getElementById("demo").innerHTML="语句 1";
document.getElementById("myDIV").innerHTML="语句 2";
}
```

后面章节将详细的讲解更多函数的知识。

6）JavaScript 对英文字母大小写敏感

JavaScript 对英文字母大小写是敏感的。当编写 JavaScript 语句时，请留意是否关闭

英文字母大小写切换键。例如，函数 getElementById 与 getElementbyID 是不同的，变量 myDiv 与 MyDiv 也是不同的。

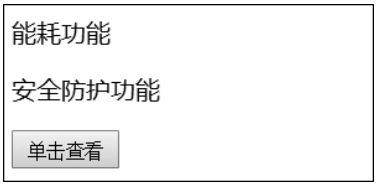

图 5-9　JavaScript 代码块使用

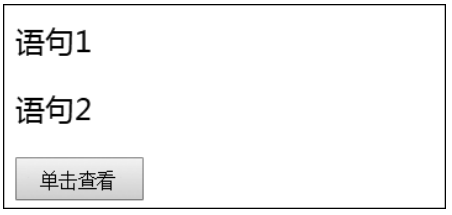

图 5-10　JavaScript 代码改变元素

7）空格

JavaScript 会忽略多余的空格。可以向脚本添加空格提高其可读性。例如，下面的两行语句是等效的。

```
var x="2";
var x="2";
```

8）对代码行进行折行

如果 JavaScript 语句太长，对其进行折行的最佳位置是某个运算符。下面的例子会正确地显示。

```
document.getElementById("demo").innerHTML=
"基于 Web 技术的物联网应用开发";
```

也可以在文本字符串中使用反斜杠(\)对代码进行换行。

例如，以下是正确写法。

```
document.write("基于 Web 技术的 \
物联网应用开发");
```

以下是错误写法。

```
document.write \
("基于 Web 技术的物联网应用开发");
```

3. JavaScript 注释

像其他所有语言一样，JavaScript 的注释在运行时也是被忽略的。注释只给程序员提供消息。JavaScript 注释可用于提高代码的可读性。

JavaScript 注释有两种：单行注释和多行注释。

1）单行注释

单行注释以//开头。下面的例子使用单行注释来解释代码。