

第3章

脚本语言——JavaScript

使用 JavaScript 脚本语言,可使 HTML 网页具有简单的动态性与计算功能。JavaScript 可以嵌入到 HTML 文档与超文本标识语言一起显示页面信息,实现在页面中与用户的交互功能,开发能在客户端完成的 Web 应用程序。

本章主要解决以下问题:

- 什么是 JavaScript;
- 如何将 JavaScript 脚本代码嵌入到 HTML 文档;
- 什么是 JavaScript 对象;
- 如何使用 JavaScript 对象的属性与方法;
- 如何定义 JavaScript 变量;
- 如何创建 JavaScript 函数;
- 如何控制 JavaScript 语句的流程;
- 什么是 JavaScript 事件与事件处理方法;
- 如何编写与使用单独保存的 JS 文件。

3.1 在 HTML 中插入 JavaScript

本节主要解决什么是 JavaScript、如何将 JavaScript 脚本代码嵌入到 HTML 文档、什么是 JavaScript 对象,如何使用 JavaScript 对象的属性与方法等问题。

3.1.1 什么是 JavaScript

JavaScript 是 Java 的子集,具有 Java 语言的基本语句,具有 Java 语言的基本特性,是由对象、对象属性与对象方法(函数)构成的,是一种基于对象和事件驱动、具有安全性的脚本语言,其主要特点是比 Java 语言容易理解,能与 HTML 语言一起完成显示页面信息的任务。

可能有人会问,运行 JavaScript 需要新的软件环境支持吗?回答是不用,与 HTML 语言相同,还是使用 Web 浏览器。目前大部分的 Web 浏览器都支持 JavaScript 的功能。

3.1.2 在 HTML 文档中插入 JavaScript 语言的方式

1. 将 JavaScript 代码插入到 HTML 文档

例 3.1 一个包含有 JavaScript 语言的 HTML 文档(3-1. htm)。

```
<html>
<body>
<script language="JavaScript">
/* 隐藏下面 JavaScript 语句
多行注释
多行不显示
多行不显示的情况 */
document.write("<h4>这是我们第一次使用 JavaScript 语言</h4><hr>")
//隐藏一行
</script>
</body>
</html>
```

保存文件为 3-1. htm,在浏览器打开 3-1. htm 网页,可以看到如图 3.1 所示页面。

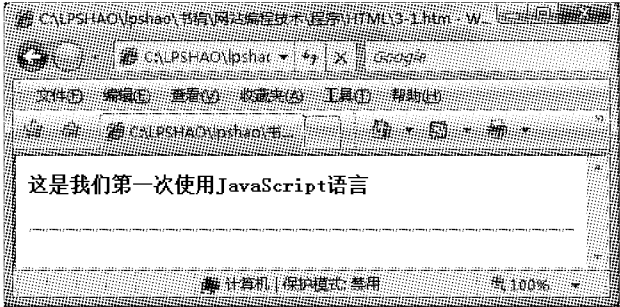


图 3.1 网页 3-1. htm 的运行结果

2. 知识点归纳

1) 插入 JavaScript 语言的<script></script>标记码

从 3-1. htm 中可以看出 JavaScript 语言是通过<script></script>标记码对插入到 HTML 文档中的,而且可以放在 HTML 文档中的任何地方。它以<script language="JavaScript">开始,以</script>结束。

插入 JavaScript 语言的语法格式:

```
<script language="JavaScript">
    JavaScript 语句
</script>
```

2) JavaScript 命令的大小写之别

JavaScript 与 Java 命令一样区别大小写。因此,在 3-1. htm 中的 JavaScript 命令代码都是小写字母,如例子中的 document.write() 语句。

3) JavaScript 的注释标记

JavaScript 有自己的注释标记。用来告诉浏览器将 JavaScript 语句隐藏起来不显示。JavaScript 注释标记的语法格式:

```
//一行注释
/* 多行注释 */
```

说明: // 标记注释一行,即一行不显示,而 /* ... */ 标记可以将多行隐藏起来不显示。

4) document 对象及 write 方法

document 对象为 JavaScript 内置的文档对象,它具有 write() 方法,可以在页面中显示指定的文字。

3.1.3 使用 JavaScript 对象属性与方法

1. 创建一个包含多个 JavaScript 对象的文档

例 3.2 一个使用 JavaScript 的 document 对象的 lastModified 属性显示最后修改日期、使用 window 对象的 alert 方法显示一个提示对话框、使用 window 对象的 prompt() 方法显示一个输入信息对话框的文档(3-2. htm)。

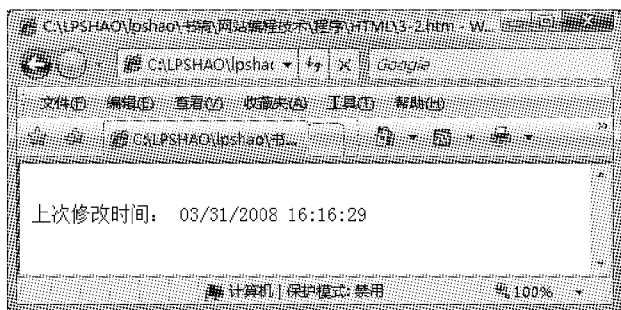
```
<html>
<body>
<br>上次修改时间:
<script language="JavaScript">
document.write(document.lastModified)
alert("欢迎你第二次使用 JavaScript")
prompt("请输入你的姓名:", "张驰")
</script>
</body>
</html>
```

在浏览器打开 3-2. htm 网页,可以看到如图 3.2 所示页面。

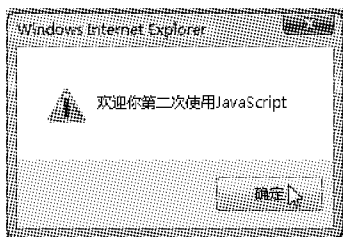
2. 知识点归纳

1) 调用对象方法和属性的方式

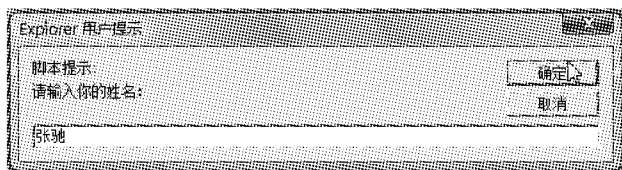
在 3-2. htm 中使用了 document 对象的 write() 方法和 lastModified 属性。在 document.write() 语句点(.) 左边是“document”对象名,点(.) 右边 write 为 document 对象的方法名,write() 方法的功能是输出圆括号内的参数值到窗口。JavaScript 语言通过对象定义其方法,通过对象名. 方法名来调用对象的方法完成指定的功能。例如通过



(a) 网页 3-2.htm 的运行结果



(b) 提示对话框



(c) 输入信息对话框

图 3.2 例 3.2 程序运行结果

document.write()调用了 document 对象的 write()方法。write()方法完成显示信息的任务,显示的信息由圆括号对内的参数提供。

调用对象方法和使用对象属性有如下语法格式:

对象名.方法名 ([参数])

对象名.属性名

2) alert 方法的功能

在 3-2.htm 中还使用了 window 对象的 alert 方法,它的功能是显示一个带有提示信息的对话框,如图 3.2 所示。alert()方法一般用于提示或提醒用户某一事情。

调用 alert 方法的语法格式:

alert("提示信息")

提示对话框中的提示信息由 alert()括号内的“提示信息”定义。单击对话框“确定”按钮,对话框会消失。

alert()方法前面没有对象名,这是因为 alert()方法为 window 对象的方法,没有指明对象名,JavaScript 默认为 window 对象。

3) prompt()方法的功能

在 3-2.htm 中还使用了 window 对象的 prompt()方法,它的功能是显示一个可以让用户输入信息的对话框。输入信息对话框由“提示信息”,“初值”、一个“确定”按钮和一个“取消”按钮组成,单击“确定”按钮可以提交用户输入的信息,同时关闭对话框。

调用 prompt()方法的语法格式:

prompt("提示信息", "初值")

prompt()方法的括号中包括了两个字符串,一个是要显示的信息,一个是输入文本框内显示的初始信息,本例中为“张驰”,也可以什么都不写或写成任意字符。所用的引号一定要成对使用,不能是一个双引号一个单引号。引号一定为西文的。

3.2 JavaScript 变量、数据类型与表达式

本节主要解决如何声明保存数据的 JavaScript 变量,有哪些类型的 JavaScript 数据,有哪些类型的数据运算符,有哪些类型的表达式等问题。

3.2.1 声明 JavaScript 变量的方式

1. 变量与常量

在例 3.2 中使用对话框输入的消息都丢失了,怎样才能保存用户在页面上输入的信息呢?可以使用变量。

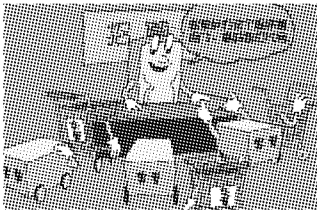
变量是计算机提供的存放数据的内存空间的代号。变量的功能就是保存数据,存放在变量中的数据可以改变,也就是说变量可以保存不同的数据。

常量是计算机中存放固定不变数据的内存空间的代号。

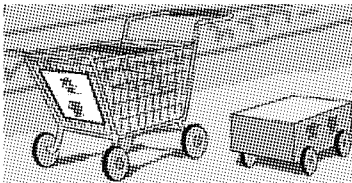
在 JavaScript 中使用变量和常量前要先声明其代号,即变量名,如图 3.3(a)所示,在应用程序中通常用拼音字母或英文单词作为变量名(变量的代号)。

常量在使用过程中存放的是固定不变的数据,如图 3.3(b)所示。

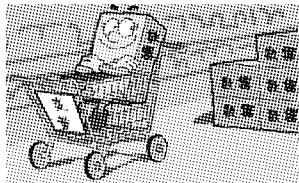
变量在使用过程中可以不存放数据(为空),也可以存放不同数据,如图 3.3(b)与(c)所示。



(a) 数据常量



(b) 空变量



(c) 有数据的变量

图 3.3 变量与常量

2. 在程序中声明变量

例 3.3 带有 JavaScript 变量,并给变量赋值的文档(3-3.htm)。

```
<html>
<body>
<script language="JavaScript">
var name=prompt("请输入你的姓名:", "张驰")
var x=7
```

```

var y=8
document.write("欢迎你,"+name)
document.write("<p>x+y="+(x+y))
</script>
</body>
</html>

```

在浏览器打开 3-3. html 网页,可以看到如图 3.4 所示页面。

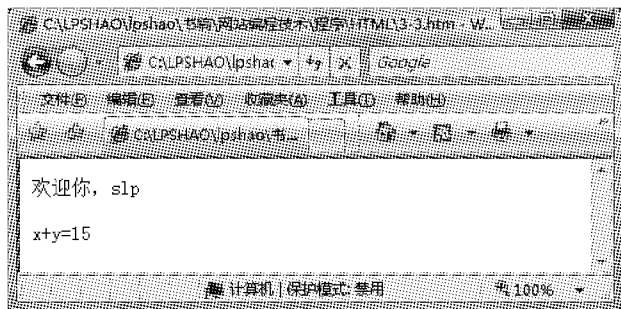


图 3.4 网页 3-3. html 的运行结果

在图 3.4 所示的输入对话框输入 slp,单击“确定”按钮,在浏览器中可以看到如图 3.4 所示的结果。可以输入任意一个名字,会看到输入的名字显示在页面上。

3. 知识点归纳

1) 声明变量的方法

JavaScript 声明变量的方法和变量赋值的语法规则:

var 变量名=数据值

JavaScript 的 var 关键字用来声明一个 JavaScript 变量,并将“数据值”赋给变量。例如,在 3-3. html 中用 var 定义了 3 个变量 x,y 和 name,并将 7,8 数据值赋给了变量 x,y。另外将 prompt()方法得到的数据“slp”赋给了变量 name。给变量赋值所用的等号(=)称为赋值运算符。

2) 选取变量名的规则

变量名对大小写非常敏感,大小写字母代表不同的变量。另外,变量名称的长度是任意的,但必须遵循以下规则:

- 第一个字符必须是一个字母(大小写均可)、或一个下划线(_)或一个美元符(\$)。
- 后续的字符可以是字母、数字、下划线或美元符。
- 变量名称不能是 JavaScript 的关键字。

变量可以用来保存数据,JavaScript 中的数据指的是什么呢? 数据有什么不同吗?

3.2.2 JavaScript 的数据类型

JavaScript 有 6 种数据类型。不同类型的数据可以表示不同的信息,进行不同的数据计算。

1. 数值数据类型

数值数据类型包括整型 `int`,例如 12、22。整型数可以为正数、0 或者负数;浮点型 `float`,例如 7.254、-3.4、2E5,浮点数可以包含小数点、也可以包含一个 `e`(大小写均可,在科学记数法中表示“10 的幂”)、或者同时包含这两项。

2. 字符串类型 `string`

例如“你好!”、“Hello!”、“254”都是 `string` 类型的数据。字符串数据必须使用单引号或双引号括起来。

3. 逻辑类型 `boolean`

`boolean` 数据类型的值只有两个: `true` 与 `false`。这是两个特殊值,不能使用 1 和 0 代替。

4. 空数据类型 `null`

`null` 数据类型只有一个值 `null`,表示没有任何值,它表示什么也没有。空 `null` 与 0 值或空字符串 `""` 是不一样的,0 值或空字符串 `""` 是一个实际值。

5. `undefined` 未定义数据类型

`undefined` 数据类型只有一个值 `undefined`,表示变量被创建后未给该变量赋值。如果对对象属性不存在,声明了变量但从未赋值,将返回 `undefined` 值。例如可使用语句 `if(typeof(x) == "undefined")` 判别变量 `x` 是否赋值。

6. `Object` 对象类型

除了上面提到的各种常用类型外,JavaScript 中包含对象类型。表示数据为某种对象(类)。

变量和常量也具有数据类型。变量的数据类型根据其赋值的数值的数据类型来确定。

3.2.3 JavaScript 的运算符

运算符即对数据进行运算的符号,JavaScript 按照运算元素的个数可分为一元、二元和三元 3 种运算符,一元运算符只有一个运算元素,如 `x++` 或 `++x`;二元运算符有两个运算元素,如 `5 * 4`;三元运算符具有三个运算元素。

按照运算符的运算规律来划分有如下 6 种运算符:

1. 赋值运算符

赋值运算符有如下 6 种类型：

(1) “=”表示将右边数赋值给左边变量。例如 $X=5$, 5 被赋给了 X。

(2) “+=”表示左、右两边数相加,结果赋给左边变量。例如 $X=5, Y=5, X+=Y$ 将 10 赋给了 X。

(3) “-=”表示左、右两边数相减,结果赋给左边变量。例如 $X=5, Y=5, X-=Y$ 将 0 赋给了 X。

(4) “*=”表示左、右两边数相乘,结果赋给左边变量。例如 $X=5, Y=5, X*=Y$ 将 25 赋给了 X。

(5) “/=”表示右边数除左边数,结果赋给左边变量。例如 $X=5, Y=5, X/=Y$ 将 1 赋给了 X。

(6) “%=”表示右边数除左边数,余数赋给左边变量。例如 $X=25, Y=7, X\%=Y$ 将 4 赋给了 X。

2. 算术运算符

算术运算符有加“+”、减“-”、乘“*”、除“/”、取余“%”和取负 $X=-X$ 、加 1“++”, 减 1“--”等。

例如 $Y=++5$, Y 的值为 6; 例如 $Y=--6$, Y 的值为 5。其他运算符的含义与平时的数值运算一样, 例如 $X=54+5*2$, X 的值为 44。

3. 字符串运算符

字符串运算符只有一个合并运算符“+”, 表示将两个字符串合并在一起, 例如: 'abc'+“你好!”, 合并后的值为“abc 你好!”。

4. 比较运算符

比较运算符有如下 6 种类型：

(1) “==”：如果两边数据相等返回 true。例如： $2==2$ 返回 true。

(2) “!=”：如果两边数据不相等返回 true; 例如：“THE”!=“TH”返回 true。

(3) “>”：如果左边数大于右边数返回 true; 例如： $2>1$ 返回 true。

(4) “<”：如果左边数小于右边数返回 true。

(5) “>=”：如果左边数大于或等于右边数返回 true。

(6) “<=”：如果左边数小于或等于右边数返回 true。

5. 逻辑运算符

逻辑运算符有如下 3 种类型：

(1) “&&”：称为逻辑与, 当表达式两边的逻辑值都是 true 时结果为 true, 否则结果为 false。例如： $2>1 \&\& 5<4$, 其结果为 true; $2>1 \&\& 5>4$, 其结果为 false。

(2) “||”：称为逻辑或，当表达式两边的逻辑值中任意一个是 true 时结果为 true，当表达式两边的逻辑值都为 false 时结果为 false。例如， $2 > 1 \ || \ 5 < 4$ ，其结果为 true。 $2 > 1 \ || \ 5 > 4$ ，其结果也为 true。

(3) “!”：称为逻辑非，当表达式的逻辑值为 true 时返回结果 false，当表达式的逻辑值为 false 时返回结果 true。例如： $!2 > 1$ 结果为 false。

6. 条件运算符

条件运算符只有一个“?”，条件运算符与上面的运算符略有不同。条件表达式的结构如下。

(条件)? 结果 1:结果 2

首先计算条件（逻辑表达式或比较表达式）的值，值为 true 表达式的值为结果 1，值为 false 表达式的值为结果 2。例如，条件表达式如下：

(day = "星期六")? "今天是周末": "今天不是周末"

当变量 day 为“星期六”时表达式的结果为“今天是周末”，否则结果为“今天不是周末”。

条件运算符为三元运算符。

7. 运算符优先级

当表达式里有多种运算符号的时候，运算的次序是什么呢？其优先级从高到低顺序为：

- (1) 乘、除、模(*、/、%)；
- (2) 加、减(+、-)；
- (3) 比较(<、>、<=、>=、==、!=)；
- (4) 逻辑与(&&)；
- (5) 逻辑或(||)；
- (6) 条件(?)。

优先级指的是复合运算进行计算时的先后顺序，对于所有的二元运算都是从左到右进行计算，可以用圆括号()来忽略优先级，即()小括号优先级最高。例如：

- 算术表达式 $(5 + 4) * 5$ 的值为 45，而 $5 + 4 * 5$ 的值为 25。
- 逻辑表达式 $false || true \&\& false$ 的值为 false；
- 条件表达式 $'4 * 5 + 18 > 22'$ 的值为“对”。

3.2.4 JavaScript 的表达式

1. JavaScript 表达式

JavaScript 表达式是 JavaScript 变量、常量运算符以及表达式的任意组合。JavaScript 表达式的计算结果为一个数据值，表达式的类型确定其值的数据类型。

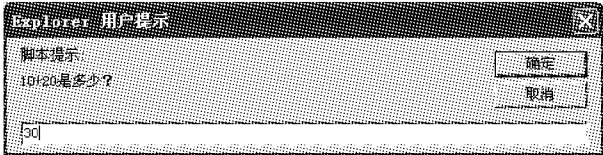
2. JavaScript 表达式的类型

根据组合表达式运算符的类型,JavaScript 的表达式可分为 6 种类型:赋值表达式、算术表达式、字符串表达式、比较表达式、逻辑表达式和条件表达式。

例 3.4 使用 JavaScript 表达式,给出表达式结果的文档(3-4. htm)。

```
<html>
<head>
<script language="JavaScript">
var question="10+ 20 是多少? ";
var answer=30;
var x1="计算正确";
var x2="计算错了";
var x5=prompt(question,"0");
var output= (x5==answer)? x1:x2;
</script>
</head>
<body>
<script language="JavaScript">
document.write(output);
</script>
</body>
</html>
```

在浏览器打开 3-4. htm 网页,可以看到如图 3.5 所示对话框,输入 30 并单击“确定”按钮后,会出现图 3.5 所示页面。



(a) 输入对话框



(b) 网页 3-4.htm 的运行结果

图 3.5 例 3.4 程序运行结果