

高等学校计算机教材

# JavaScript 程序设计基础教程

阮文江 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

JavaScript 程序设计基础教程 / 阮文江编著. —北京: 人民邮电出版社, 2004.8

高等学校计算机教材

ISBN 7 - 115 - 12388 - 8

. J ... . 阮 ... . JAVA 语言—程序设计—高等学校—教材 . TP312

中国版本图书馆 CIP 数据核字 (2004) 第 083178 号

## 内 容 提 要

本书是学习 JavaScript 程序设计的基础教材, 共分 10 章, 主要内容包括: HTML 制作、JavaScript 编程基础、流程控制、函数、对象编程、文档对象模型与事件驱动、使用浏览器对象、级联样式表 (CSS) 技术和 JavaScript 网页特效等。为便于教学, 每章均附有练习题。

本书内容丰富, 讲解循序渐进、深入浅出, 简明易懂。本书为高等院校本、专科各专业 JavaScript 程序设计、动态网页制作、计算机应用基础 (编程入门部分) 等课程的教材, 也可作为电子商务、电子政务的辅助培训教材。

高等学校计算机教材

### JavaScript 程序设计基础教程

◆ 编 著 阮文江

责任编辑 滑 玉

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线: 010-67129259

北京汉魂图文设计有限公司制作

北京隆昌伟业印刷有限公司

新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16

印张: 17

字数: 402 千字 2004 年 8 月第 1 版

印数: 1-5 000 册 2004 年 8 月北京第 1 次印刷

ISBN 7-115-00000-0/TP · 0000

定价: 22.00 元

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

## 编者的话

JavaScript 是 Web 上的一种功能强大的编程语言,用于开发交互式的 Web 页面。它不仅可以直接应用于 HTML 文档以获得交互效果或其他动态效果,而且可以运行于服务器端,从而替代传统的 CGI 程序。

目前,关于介绍 JavaScript 技术的书籍有很多,但大多适用于开发人员,偏重于动态网页制作的技巧,而对基本的编程技术的介绍却很少,不适合作为初学者的教学用书。

与目前同类书籍相比,本书有以下几个特点。

- 为读者提供程序设计的入门级语言教材(如早期的 Pascal, Pascal 是学习程序设计的首选入门语言,但现在已很少使用),因此强调程序设计的基础性、简单性和趣味性,从代码这一级培养学生对计算机的应用兴趣和水平,以便为学生进一步掌握计算机技术打下良好的语言基础。

- 两个引导:即引导读者使用软件开发工具(如调试器、帮助系统),引导读者使用 JavaScript 的对象编程技术。

- 全书贯穿了大量典型实例,深入浅出地介绍了 JavaScript 动态网页制作技术,并且各章配有大量练习题。

通过本书的学习可使读者在以下几个方面获得收益:

- 掌握 HTML 网页制作技术;
- 初步了解网站建立和维护的方法;
- 掌握 JavaScript 的基于对象和事件驱动的程序设计技术;
- 掌握 JavaScript 的动态网页制作技术。

另外,通过本书的学习,读者能够掌握以下工具的使用:

- 网页制作工具 FrontPage、脚本编辑器及其帮助系统;
- Windows 环境下的 Internet 信息服务器 IIS。

为了便于教师使用本教材,编者还准备了本书的教学辅助材料,包括各章的电子讲稿、例题文件以及习题答案,并发布在中山大学计算机基础教育中心的教学网站上,其网址为:

<ftp://cec.zsu.edu.cn/public/book/js>

也可在人民邮电出版社网站([www.ptpress.com.cn](http://www.ptpress.com.cn))的“下载区”中下载本书电子教案。

在本书的编写过程中,得到中山大学计算机基础教育中心全体老师的支持和帮助,林卓然、何丁海、彭金泉、梁广德、潘小轰、薛春香、李聪明等老师还对书中内容提出了宝贵意见,在此表示衷心感谢。

编者电子邮件地址: [pusrwj@zsu.edu.cn](mailto:pusrwj@zsu.edu.cn)

编者

2004年6月

# 目 录

第 1 章 JavaScript 简介	1
1.1 JavaScript 是什么	1
1.1.1 什么是 JavaScript	1
1.1.2 JavaScript 的基本特点	1
1.1.3 其他常用的 Web 开发语言	1
1.2 JavaScript 的应用	2
1.2.1 客户端应用	2
1.2.2 服务器端应用	3
1.3 在 Web 页面中使用 JavaScript	3
1.3.1 HTML 的基本结构	3
1.3.2 在 HTML 中嵌入 JavaScript	4
1.3.3 链接 JavaScript 文件	5
1.4 编写 JavaScript 的工具	6
1.4.1 使用纯文本编辑器	6
1.4.2 使用专业化脚本编辑工具	6
1.4.3 使用 Microsoft 脚本编辑器	6
习题	8
第 2 章 HTML 制作	10
2.1 基本概念	10
2.1.1 什么是 Web	10
2.1.2 超链接	10
2.1.3 HTML 语言	11
2.1.4 DHTML 语言	11
2.2 制作环境	11
2.2.1 网页设计工具	11
2.2.2 建立 Web 服务器	12
2.3 HTML 标记	15
2.4 结构标记	16
2.5 常用标记和属性	17
2.5.1 标题	17
2.5.2 段落	18
2.5.3 列表	20

---

2.5.4 水平线 .....	23
2.6 设置颜色、字体 .....	24
2.6.1 RGB 颜色 .....	24
2.6.2 通过<BODY>设置颜色 .....	25
2.6.3 使用<FONT>标记 .....	25
2.7 建立超链接 .....	26
2.7.1 文本链接 .....	26
2.7.2 图形链接 .....	27
2.7.3 链接文档中的特定位置 .....	27
2.8 表格 .....	29
2.8.1 创建表格 .....	29
2.8.2 指定行和单元格 .....	29
2.8.3 指定表格标题和表格头 .....	30
2.9 框架网页 .....	31
2.9.1 框架布局 .....	31
2.9.2 框架内容 .....	31
2.9.3 对不支持框架的浏览器的支持 .....	32
2.10 表单 .....	33
2.11 多媒体 .....	35
2.12 <META>标记 .....	35
习题 .....	36
第 3 章 JavaScript 编程基础 .....	39
3.1 常量 .....	39
3.1.1 数值型 .....	39
3.1.2 字符串 .....	40
3.1.3 布尔型 .....	41
3.1.4 空值型 .....	41
3.1.5 未定义值 .....	41
3.2 变量 .....	41
3.2.1 变量取名 .....	41
3.2.2 变量声明 .....	42
3.2.3 变量赋值 .....	42
3.2.4 变量类型 .....	43
3.2.5 简述变量的作用域 .....	43
3.3 表达式 .....	44
3.3.1 运算符与表达式 .....	44
3.3.2 算术运算符 .....	44
3.3.3 比较运算符 .....	45

---

3.3.4	逻辑运算符	46
3.3.5	字符串运算符	47
3.3.6	赋值运算符	48
3.3.7	条件运算符	49
3.3.8	其他运算符	50
3.3.9	运算符的优先级	50
3.3.10	表达式中的类型转换	51
	习题	52
第 4 章	流程控制	55
4.1	使用对话框	55
4.1.1	警示对话框	55
4.1.2	确认对话框	56
4.1.3	提示对话框	57
4.2	JavaScript 程序的基本流程	58
4.3	分支结构	59
4.3.1	if 语句	59
4.3.2	switch 语句	64
4.4	循环结构	66
4.4.1	for 语句	66
4.4.2	while 语句	68
4.4.3	do while 语句	69
4.4.4	在循环中使用 break 和 continue 语句	70
4.4.5	循环的嵌套	72
4.5	调试程序	74
4.5.1	排除语法错误	74
4.5.2	排除语义错误	75
4.5.3	使用 Microsoft 脚本编辑器调试程序	75
	习题	77
第 5 章	函数	80
5.1	什么是函数	80
5.2	使用预定义函数	81
5.3	函数定义和函数调用	82
5.3.1	函数定义	82
5.3.2	函数调用	82
5.4	函数参数的使用	84
5.4.1	给函数添加参数	84
5.4.2	使用多个参数	85

5.5 使用函数返回值 .....	85
5.5.1 给函数添加返回值 .....	86
5.5.2 区分函数和过程 .....	86
5.6 函数的嵌套调用 .....	87
5.6.1 函数嵌套调用的形式 .....	87
5.6.2 解决嵌套调用引起的效率问题 .....	88
5.7 递归函数 .....	89
5.8 变量的作用域 .....	91
5.8.1 全局变量和局部变量 .....	91
5.8.2 使用局部变量避免冲突 .....	93
习题 .....	94
第 6 章 对象编程 .....	96
6.1 初探对象编程 .....	96
6.2 对象的基本概念 .....	97
6.2.1 什么是对象 .....	97
6.2.2 对象的属性和方法 .....	97
6.2.3 在 JavaScript 中使用对象 .....	98
6.2.4 通过帮助系统查看内置对象和浏览器对象 .....	98
6.2.5 创建和删除对象 .....	100
6.3 Math 对象 .....	100
6.4 Date 对象 .....	101
6.4.1 Date 对象的主要方法 .....	101
6.4.2 创建 Date 对象 .....	102
6.5 Number 对象 .....	103
6.6 String 对象 .....	104
6.7 Array 对象 .....	105
6.7.1 什么是数组 .....	105
6.7.2 创建和访问数组 .....	105
6.7.3 使用 for...in 语句 .....	107
6.7.4 Array 对象的常用属性和方法 .....	107
6.7.5 二维数组 .....	108
习题 .....	109
第 7 章 文档对象模型与事件驱动 .....	111
7.1 文档对象模型 .....	111
7.1.1 认识文档对象模型 .....	111
7.1.2 引用文档对象模型中的对象 .....	112
7.2 事件驱动 .....	113

---

7.2.1 事件驱动的基本概念 .....	113
7.2.2 JavaScript 的常用事件 .....	113
7.3 处理事件 .....	114
7.3.1 使用事件的方法 .....	114
7.3.2 使用制作工具自动生成事件处理函数的模板 .....	115
7.4 使用事件 (Event) 对象 .....	117
习题 .....	120
第 8 章 使用浏览器对象 .....	121
8.1 窗口 (Window) 对象 .....	121
8.1.1 window 对象的属性和方法 .....	121
8.1.2 打开和关闭窗口 .....	122
8.1.3 使用定时器 .....	123
8.1.4 页面跳转 .....	124
8.2 文档 (document) 对象 .....	126
8.2.1 document 对象的属性和方法 .....	126
8.2.2 使用 all 属性访问 HTML 元素 .....	127
8.3 表单 (form) 对象 .....	130
8.3.1 form 对象的属性、方法和事件 .....	130
8.3.2 表单处理 .....	132
8.3.3 表单元素对象的属性、方法和事件 .....	133
8.3.4 处理表单元素示例 .....	135
习题 .....	140
第 9 章 级联样式表 (CSS) 技术 .....	142
9.1 CSS 简介 .....	142
9.2 样式定义 .....	143
9.2.1 样式定义的格式 .....	143
9.2.2 选择器的分类 .....	144
9.3 使用样式 .....	146
9.3.1 嵌入样式表 .....	146
9.3.2 链接外部样式表 .....	147
9.3.3 内嵌样式 .....	148
9.3.4 CSS 样式的优先级 .....	148
9.4 Style 对象 .....	149
9.5 CSS 属性 .....	150
9.5.1 字体属性 .....	151
9.5.2 文本属性 .....	152
9.5.3 颜色和背景属性 .....	153

---

9.5.4 容器属性 .....	154
9.5.5 列表属性 .....	158
9.5.6 鼠标属性 .....	159
9.5.7 定位和显示 .....	161
9.5.8 CSS 滤镜 .....	163
习题 .....	167
<b>第 10 章 JavaScript 网页特效</b> .....	<b>169</b>
10.1 文字特效 .....	169
10.2 图片特效 .....	171
10.3 时间特效 .....	173
10.4 窗体特效 .....	177
10.5 鼠标 .....	183
10.6 菜单 .....	187
习题 .....	192

# 第 1 章 JavaScript 简介

JavaScript 是 Web 页面中的一种脚本编程语言，可用于 Web 系统的客户端和服务端编程。JavaScript 程序通常出现在 Web 页面的<SCRIPT>标记中，要提高编写 JavaScript 程序的效率可使用专业化的脚本编辑工具。

## 1.1 JavaScript 是什么

### 1.1.1 什么是 JavaScript

JavaScript 是 Web 上的一种功能强大的编程语言，用于开发交互式的 Web 页面。它不仅可以直接应用于 HTML 文档以获得交互效果或其他动态效果，而且可以运行于服务器端，从而替代传统的 CGI 程序。

JavaScript 的前身叫做 LiveScript，是 Netscape 公司开发的脚本语言。后来在 Sun 公司推出著名的 Java 语言之后，Netscape 公司和 Sun 公司于 1995 年一起重新设计了 LiveScript，并把它改名为 JavaScript。

### 1.1.2 JavaScript 的基本特点

JavaScript 是一种基于对象 (Object) 和事件驱动 (Event Driven) 并具有安全性能的解釋型脚本语言。具有以下几个基本特点。

(1) JavaScript 是脚本编程语言：JavaScript 采用小程序段的方式实现编程，与 HTML 代码结合在一起，通常由浏览器解释执行。

(2) JavaScript 是基于对象的语言：JavaScript 的许多功能来自于脚本环境中对象的方法与脚本的相互作用。在 JavaScript 中，既可以使用预定义对象，也可以使用自定义对象。

(3) 安全性：在 HTML 页面中 JavaScript 不能访问本地硬盘，也不能对网络文档进行修改和删除，而只能通过浏览器实现信息浏览或动态交互。

(4) 跨平台性：在 HTML 页面中 JavaScript 的执行依赖于浏览器本身，与操作环境无关。只要在计算机上安装了支持 JavaScript 的浏览器，那么 JavaScript 程序就可以正确执行。

### 1.1.3 其他常用的 Web 开发语言

在 Web 应用的开发中，除了 JavaScript 之外，还有许多其他语言。如：Sun 公司的 Java、Microsoft 公司的 VBScript 等。

#### 1. Java

Java 是由 Sun 公司开发的一种与平台无关的、面向对象的程序设计语言。Java 可以用

来设计独立的应用程序，也可以用来创建一种称为 Applet 的小应用程序。经过编译后，Applet 成为一种平台的字节代码，这种字节代码可以运行在拥有 Java 虚拟机的任何平台上。在 HTML 文档中，用标记<applet>来标明 Java Applet。在 JavaScript 中，可以通过 document.applet-

etname.appletfun( )调用 Java Applet 的方法，其中 appletname 和 appletfun 分别指 Java 小应用程序名和方法名。

### 2. VBScript

VBScript 是 Microsoft 公司开发的脚本语言，与 Visual Basic 的语法基本相同。VBScript 嵌在 HTML 页面中，由浏览器解释执行。使用<SCRIPT language=“VBScript”>和</SCRIPT>标识 VBScript 程序。与 JavaScript 不同，VBScript 还可以作为普通的脚本语言在其他的应用程序中使用。但是，目前只有 IE (Internet Explorer) 浏览器支持 VBScript。

### 3. JScript

Microsoft 公司在 Netscape 公司发布的 JavaScript 的基础上，也开发了自己的 JavaScript 规范，叫做 JScript。JScript 与 JavaScript 在基本功能和语法上是相同的，但它结合了 IE 浏览器特性，因此早期版本的 JScript 与 JavaScript 存在兼容性问题。随着这两种语言的升级，其兼容性也在不断改善，Netscape 6.0 支持的 JavaScript 1.5 与 IE 5.5 支持的 Jscript 5.5 几乎完全兼容。现在，Web 开发人员可以编写出在这两种浏览器上都能正常工作的 JavaScript 脚本程序，尽管它们还是存在一些微小的差异。

### 4. Perl

Perl 是 Practical Extraction and Report Language 的缩写，它是由 Larry Wall 设计的，并由他不断更新和维护。它主要由 C 语言，其次由 sed、awk、UNIX shell 和至少数十种其他的语言和工具所演化而来。Perl 是一种解释型语言，且具有许多高级语言（如 C）的强大能力和灵活性。与脚本语言一样，Perl 不需要编译器和链接器来运行代码，只要写出程序并告诉 Perl 来运行即可。Perl 在 Web 中只能用作 CGI 脚本，这是与 JavaScript 最大的不同点。Perl 运行在服务器而不是客户机上，通过返回一个动态创建的 HTML 页面来实现交互性。

## 1.2 JavaScript 的应用

作为一种脚本语言，JavaScript 在 Web 系统中的应用非常广泛。它不但可以用于编写客户端的脚本程序，在 Web 浏览器端解释执行；而且还可以编写在服务器端执行的脚本程序，在服务器端处理用户提交的信息并动态地向浏览器返回处理结果。

### 1.2.1 客户端应用

JavaScript 的典型应用是开发客户端的 Web 应用程序，将客户端的 JavaScript 脚本程序嵌入或链接到 HTML 文件。当用户使用浏览器请求这样的 HTML 页面时，JavaScript 脚本程序与 HTML 一起被下载到客户端，由客户端的浏览器读取 HTML 文件，并分辨其中是否含有 JavaScript 脚本。如果有，就解释并执行它，并以页面方式显示出来。

---

图 1-1 显示了 JavaScript 的客户端应用。

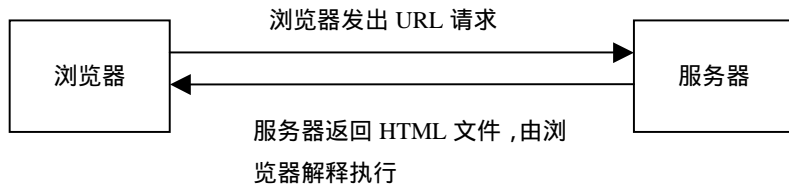


图 1-1 JavaScript 的客户端应用

### 1.2.2 服务器端应用

使用 JavaScript 还可以开发服务器端的 Web 应用程序。例如, 近年来在 Microsoft 的服务器上使用一种比较实用的技术 ASP (Active Server Pages) 或 JSP (Java Server Pages), 它用于开发服务器端的 Web 应用程序。JavaScript 可以作为 ASP 的实现脚本来开发服务器端的应用程序。

服务器端脚本的工作过程如下: 用户使用浏览器请求 URL 时, 服务器执行脚本, 将生成的数据以 HTML 文件的形式返回浏览器, 如图 1-2 所示。

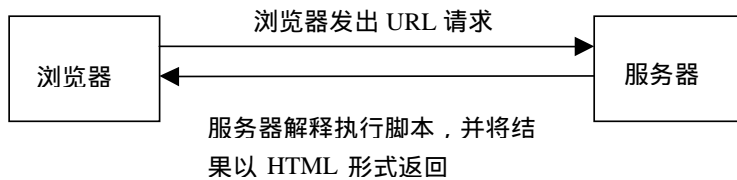


图 1-2 JavaScript 的服务器端应用

对于 JavaScript 的客户端应用和服务器应用, 要根据实际情况进行选择。客户端应用程序是在客户端解释执行, 而服务器端的应用程序则是在服务器端进行处理。从程序保密性的角度来看, 采用服务器端应用更为妥当。如果只是对数据进行验证, 那么若是服务器端的应用, 则每一次的验证都要发回服务器端, 这势必会增加网络流量和延迟, 所以数据验证通常由客户端应用来实现, 这样效率更高。

## 1.3 在 Web 页面中使用 JavaScript

### 1.3.1 HTML 的基本结构

HTML (Hypertext Markup Language) 是一种超文本标记语言, 用来描述 Web 上的超文本文件。HTML 通过在正文文本中嵌入各种标记, 使普通正文文本具有超文本的功能。HTML 文件必须由 Web 浏览器进行翻译和执行才能正确显示。

例 1.1 在页面上显示一行内容 “Hello World!”

在一个文本编辑器 (如 Windows 环境下的记事本) 中编写以下内容, 然后以文件名 S01\_01.htm 保存。

```
<HTML>
<HEAD>
<TITLE>HTML 的基本结构</TITLE>
</HEAD>
<BODY>
<P>Hello World!</P>
</BODY>
</HTML>
```

这个简单的 HTML 文件在 IE 浏览器中显示的效果如图 1-3 所示。

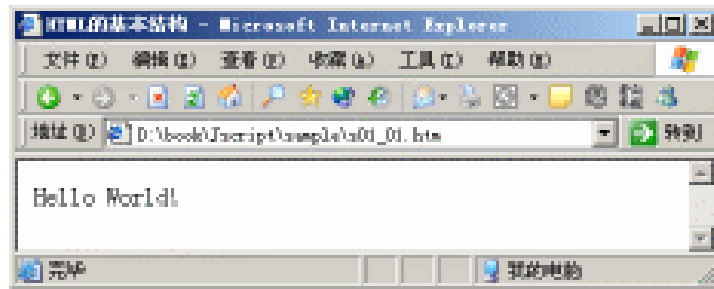


图 1-3 文件 S01\_01.htm 的显示结果

从上例可看出 HTML 文件具有以下几个最基本的结构特征。

- 一个元素的元素体可以嵌套另外的元素。例如第二行开始的头元素<HEAD>嵌套着标题元素<TITLE>。
- HTML 文件由一个<HTML>元素组成，即文件以<HTML>开始，以</HTML>结束。
- <HTML>元素的元素体又由两部分组成，即头元素<HEAD>...</HEAD>和体元素<BODY>...</BODY>。
- 头元素和体元素的元素体又由其他的元素、文本及注释组成。

有关 HTML 的更多描述，请阅读第 2 章。

### 1.3.2 在 HTML 中嵌入 JavaScript

在 HTML 中通过标记<SCRIPT>...</SCRIPT>引入 JavaScript 代码。当浏览器读取到<SCRIPT>标记时，就解释执行其中的脚本。

在使用<SCRIPT>标记时，还必须通过 Language 属性指定<SCRIPT>块中包含的是何种类型的脚本。例如 Language 属性的值是 VBScript，表示<SCRIPT>块中包含的是 VBScript 脚本；而 Language 属性的值是 JavaScript，则表示<SCRIPT>块中包含的是 JavaScript 脚本。如果在<SCRIPT>块中没有指定 Language 属性，那么浏览器将使用它的默认脚本语言对<SCRIPT>块中的代码进行解释（注：IE 的默认脚本语言是 VBScript）。

例 1.2 在文件 S01\_02.htm 中有以下代码，它通过嵌入 JavaScript 脚本在页面上显示一行内容“Hello World!”

```
<HTML><HEAD>
<TITLE>嵌入 JavaScript 代码</TITLE><!-- 定义页面标题 -->
</HEAD>
```

```
<BODY>
<SCRIPT language="javascript"> //脚本标记
    document.write("Hello World!") //在页面上显示一行文字
</SCRIPT>
</BODY></HTML>
```

这个 HTML 文件在 IE 中的显示结果与例 1.1 相同。对于这个例子，说明以下几点。

- (1) JavaScript 代码被嵌入到 HTML 文档的<SCRIPT>标记对之间。
- (2) <SCRIPT>块既可放到<HEAD>部分，也可放到<BODY>部分。在这个例子中，<SCRIPT>块被放到<BODY>部分中。
- (3) 在<SCRIPT>块中的 language 属性告诉浏览器<SCRIPT>块中包含的是 JavaScript 脚本。
- (4) 在 JavaScript 程序中，“//”为单行注释符，表示在其之后的内容为注释信息，它不会执行（如果要注释多行，则可以使用标记对（/\*...\*/））。在<!-- 定义页面标题 -->中的标记<!--...-->是 HTML 文档的注释符。
- (5) 包含在脚本中的语句 document.write()是 JavaScript 语句，它调用对象 document 的方法 write，其功能是直接在页面上显示括号中的字符串内容。

### 1.3.3 链接 JavaScript 文件

在 Web 页中引入 JavaScript 程序的另一种形式是采用链接 JavaScript 文件的形式。如果脚本程序较长或者同一段脚本可以在若干个 Web 页中使用，则可以将脚本放在单独的一个 .js 文件里，然后链接到需要它的 HTML 文件，这相当于将其中的脚本填入链接处。

要引用外部脚本文件，使用<SCRIPT>标记的 SRC 属性来指定外部脚本文件的 URL。如果使用了<SCRIPT>标记的 SRC 属性，则 Web 浏览器只使用在外部文件中的脚本，并忽略位于该<SCRIPT>标记之间的任何脚本。

具体做法如下：

- (1) 使用文本编辑器编写一个 JavaScript 脚本程序，其中不需要任何 HTML 标记（包括<SCRIPT>标记），而仅仅只有 JavaScript 代码；
- (2) 将该文件以扩展名 .js 保存在与要链接的 HTML 文件相同的位置上；
- (3) 在 HTML 文件中使用<SCRIPT>标记的 SRC 属性来链接该 .js 文件。

例 1.3 在文件 S01\_03.htm 中有以下代码，它采用链接 JavaScript 文件的方式显示一行内容“Hello World!”

```
<HTML><HEAD>
<TITLE>链接 JavaScript 代码</TITLE>
</HEAD>
<BODY>
<SCRIPT language="javascript" src="s01_03.js">
</SCRIPT>
</BODY></HTML>
```

其中，文件 S01\_03.js 与文件 S01\_03.htm 处于相同的文件夹中，并含有如下内容：

```
document.write("Hello World!")
```

文件 S01\_03.htm 在 IE 中的显示结果与例 1.1 相同。

## 1.4 编写 JavaScript 的工具

对于 JavaScript 脚本的编写,可以采用两种方式,一种是使用纯文本编辑器,另一种就是使用现成的专业化脚本编辑工具。

### 1.4.1 使用纯文本编辑器

使用纯文本编辑器(如 Windows 的记事本)来编写脚本,是早期脚本编程人员常用的一种方法。这种方式的优点是简单、易用;缺点是由于这种编辑器的主要用途就是编辑纯文本,不具备对 JavaScript 语言的特性支持,因此它只适用于脚本的少量编写和修改,而要进行大量的脚本编写和设计,则需要专业化的脚本开发工具。

### 1.4.2 使用专业化脚本编辑工具

使用可视化工具,例如 Frontpage、Dreamweaver 以及 Flash 等工具可以十分容易地在 Web 页面中加入脚本来完成一些功能。这些工具是处理 JavaScript 的专业化开发工具,具有许多处理 JavaScript 特性的功能,如代码自动生成、语法敏感编辑、调试等,因此现在的开发人员经常使用这些工具进行 Web 程序的开发,以提高效率。必须注意的是,这些工具在自动生成有关 JavaScript 代码时会加入一些冗余代码,但这不会妨碍熟练的脚本编程人员对 Web 页中脚本的控制。

笔者建议读者在学习 JavaScript 的过程中也要逐渐掌握一种专业化的脚本编辑器,本书推荐使用下面介绍的 Microsoft 脚本编辑器,使用它可以大大提高开发 Web 程序的效率。

### 1.4.3 使用 Microsoft 脚本编辑器

Microsoft 脚本编辑器是 FrontPage 提供的一种专业化 Web 编程工具,用于在 HTML 文档和 ASP 文件内查看和编辑 HTML 标签。还可以在文件中添加诸如 JavaScript 或 VBScript 之类的脚本,并调试该脚本。

Microsoft 脚本编辑器是 Microsoft Office 的可选组件之一,如果在安装 Office 时没有指定要安装这个组件,就不能看到它。一个简单的安装方案是在安装 Office 2000 或以上版本时选择“完整安装”,以后就可以随时使用它了。

#### 1. 启动 Microsoft 脚本编辑器

要启动 Microsoft 脚本编辑器,执行以下步骤。

- (1) 启动 FrontPage,并打开或新建任意一个 HTML 文件。
- (2) 将工作视图切换到“网页”视图,并选择“HTML”选项卡。此时在工作区看到的是当前所编辑页面的 HTML 代码。
- (3) 从菜单栏中的“工具”菜单选择“宏”命令,然后从弹出的子菜单中选择“Microsoft 脚本编辑器”命令,此时就会打开如图 1-4 所示的 Microsoft 脚本编辑器。

注意:如果是默认安装 Office,FrontPage 就会提示安装“Microsoft 脚本编辑器”。

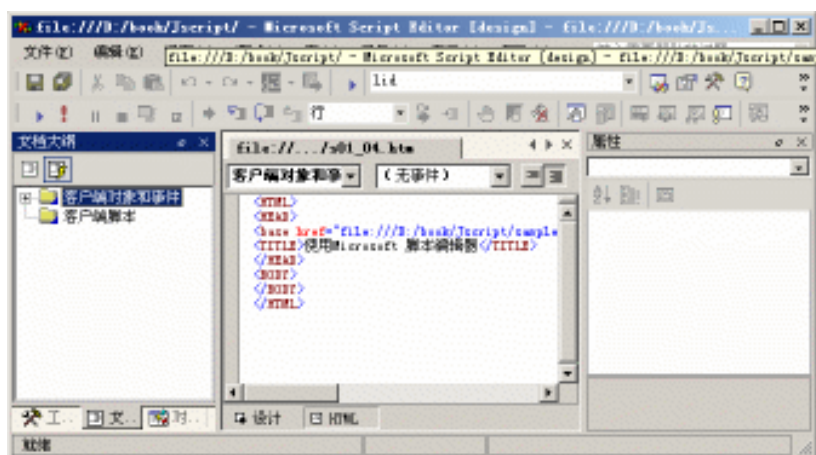


图 1-4 Microsoft 脚本编辑器

在“HTML”视图中，脚本编辑器基于文件中文本的功能对文本进行着色。例如，用棕色表示 HTML 标签，用红色表示标签属性，而用蓝色表示属性值。

## 2. 在 HTML 文档中插入<SCRIPT>块

使用 Microsoft 脚本编辑器可以很容易地加入<SCRIPT>块，步骤如下。

(1) (如果使用 VBScript，则忽略这一步) 先将工作区的插入点移至整个工作区的开始位置，即<HTML>标记之前，然后在右边的“属性”窗口中将 DOCUMENT 对象的 defaultClientScript 的属性设置为“JavaScript”，此时在<HEAD>部分将出现以下代码：

```
<meta name=vs_defaultClientScript content="JavaScript">
```

这行代码指定这个页面的默认脚本语言是 JavaScript。

(2) 将工作区的插入点移至想插入<SCRIPT>块的位置(例如，刚好在</HEAD>标记之前的位置)；然后单击鼠标右键，此时将出现一个快捷菜单，依次选择“插入脚本块”和“客户端”菜单命令，Microsoft 脚本编辑器将自动在当前插入点位置生成<SCRIPT>块标记，如下所示：

```
<script language=javascript>
<!--
/-->
</script>
```

在生成的代码中，<!-- ... //-->是 HTML 注释，而//是 JavaScript 的注释。使用这种注释标记的作用在于使较旧的不支持脚本的浏览器也能正确显示该文档的其他部分。

(3) 在<!-- ... //-->之间的空行中可以编写任意的 JavaScript 语句，如

```
<script language=javascript>
<!--
document.write("Hello World!");
/-->
</script>
```

必须注意的是，当输入对象名和句点“document.”时，Microsoft 脚本编辑器将自动弹