

实用计算机编程技术丛书

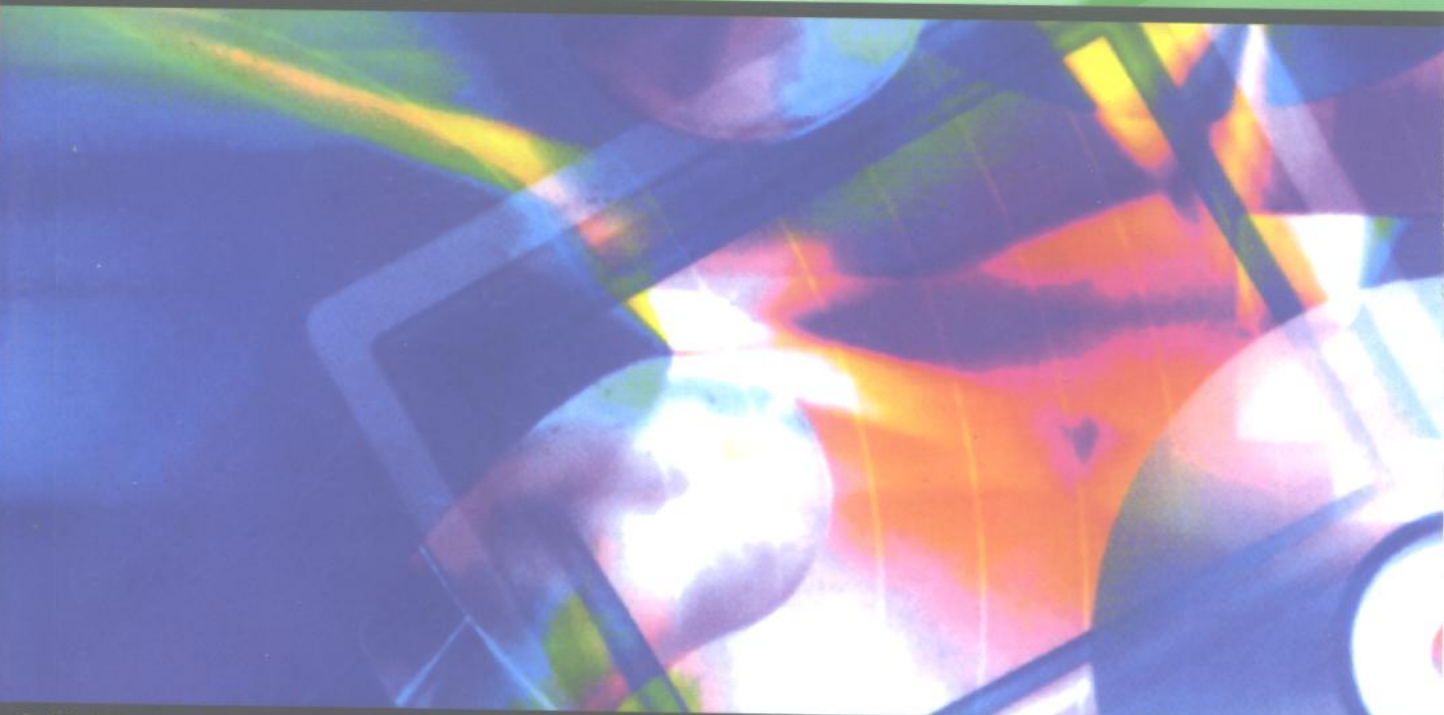
JavaScript 编程指南

2
/3

实用计算机编程技术丛书

JavaScript 编程指南

姜晓铭 主编



SHIYONG JISUANJI BIANCHENG JISHU CONGSHU



中国石化出版社

7P3:2
JXM/3

实用计算机编程技术丛书

JavaScript 编程指南

姜晓铭 主编

中国石化出版社

157708

内 容 提 要

本书由浅入深全面讨论了使用 JavaScript 进行 Web 页面制作的系列方法。全书共分 16 章, 主要内容包括 JavaScript 基本编程概念、JavaScript 中可使用的各种对象属性方法和事件、HTML 元素与 JavaScript 脚本的结合、利用 JavaScript 进行 Web 页面改善、多媒体及插件的基本概念、以及 JavaScript1.3 中的一些最新特性。本书循序渐进, 收容了 JavaScript 语言的许多细节, 并且包含大量新颖实用的范例, 具有很好的参考价值。

本书叙述清楚, 通俗易懂, 是使用 JavaScript 语言的用户甚为有用的参考书, 也是计算机应用人员和大中专院校师生必备的参考书。

图书在版编目 (CIP) 数据

JavaScript 编程指南/姜晓铭主编. —北京:
中国石化出版社, 2000
(实用计算机编程技术丛书)
ISBN 7-80043-943-7

I. J… II. 姜… III. Java 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 05028 号

JS396/14

中国石化出版社出版发行

地址: 北京市东城区安定门外大街 58 号

邮编: 100011 电话: (010) 84271859

<http://press.sinopec.com.cn>

东远先行彩色图文中心排版

海丰印刷厂印刷

新华书店北京发行所经销

*

787×1092 毫米 16 开本 20.5 印张 516 千字 印 1—3000

2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

定价: 34.50 元

前 言

随着世纪末 World Wide Web 的迅速发展, Web 开发工具也有了很大的发展。JavaScript 便是其中的一员, 它在简单的记录工具 (如 HTML) 中开始加入真正交互式的内容。同时, JavaScript 相比要有一定编程能力才能运用自如的 Java 语言来说, 它提供了网页开发和使用应用程序更为简单的方法。

JavaScript 可以做很多的事情来使 Web 页面更为生动, 甚至可以创建多页的、完整的应用程序。本书中包含了这些东西, 包括页面的自动导航, 甚至完整成熟的俄罗斯方块游戏。

这本书是为那些想学习如何用 JavaScript 来加强改善 Web 页面的人而作。读者在进行这项学习之前, 最好要有 World Wide Web 的基本概念和使用 HTML 创建 Web 页面的基本知识, 因为能从这本书中学到多少东西与读者自身的程度有关:

- 初学者会了解他人使用 JavaScript 在 Web 上干些什么, 如何使用简单的 JavaScript 语句来改善自己的 Web 页。
- 中级用户会了解使用 JavaScript 开发应用脚本以充分利用 HTML 页面的潜能。
- 高级用户会学会如何利用这一功能强大的开发工具使自己的 Web 页面开发过程随心所欲, 让自己的每一个想法都成为现实。

本书的目的在于帮助读者深入理解 JavaScript 在 Navigator、Internet Explorer 中的应用。学习本书后, 读者将能够在 HTML 文档中嵌入 JavaScript, 开发出高级的客户端应用程序。

本书主要内容如下:

- JavaScript 语言的基础概念, 包括 JavaScript 历史、特性、JavaScript 语言基本要素。
- JavaScript 对象及其功能, 以及它们与 HTML 的元素的结合。
- JavaScript 的一些应用, 包括窗体、框架、Cookie 以及图像及图像映像, 改善您的 Web 页面。
- JavaScript 的高级话题部分, 包括 JavaScript 风格单、高级事件、以及多媒体和插件的使用。
- JavaScript 的调试技术及相关调试工具, 以及最新版本的 JavaScript 的一些新特性。

在附录中, 提供了 JavaScript 的一些开发工具的介绍, 读者可在开始自己的 JavaScript 之旅前浏览一下其中的内容, 相信会有裨益。附录中还提供了较为完整的 JavaScript 语言参考, 以及在后半程学习实例中经常碰到的颜色名以及对应值的详细列表。

在每一章的最后, 除了有对该章内容的小结以外, 多数章节还附有对该章节内容最常提问的问题 (FAQ), 相信这也是大多数读者常碰到的问题。

本书由姜晓铭主编, 唐治、田小康、瞿详洲、何蕾、姜汉鲁等同志承担了主要的编写工作, 另外, 刘军、张旭东、柏丽、曹国权、关春梅、魏宏刚、韩治学、杨作梅、张联明、安俊峰等同志也参加了本书的编写工作。同时, 对在本书的编辑过程中给予我们帮助的同志表示感谢。

由于时间较仓促, 加之作者水平有限, 难免有不足和粗陋之处, 恳请广大读者谅解, 并批评指正。

目 录

第 1 章 JavaScript 简介	1
1.1 JavaScript 概念.....	1
1.1.1 JavaScript 历史.....	1
1.1.2 JavaScript 特性.....	2
1.2 JavaScript 与其它语言比较	2
1.2.1 JavaScript 和 CGI.....	2
1.2.2 JavaScript 和 Java	3
1.2.3 JavaScript 和 VBScript	3
1.3 JavaScript 用途.....	3
1.4 编写第一个 JavaScript 程序	4
1.4.1 创建 JavaScript 唯一真正需要的工具.....	4
1.4.2 创建 JavaScript 脚本	4
1.4.3 脚本的嵌入	4
1.4.4 程序的输出	6
1.4.5 注释与兼容问题	6
1.5 JavaScript 的局限性	7
1.6 小结.....	7
第 2 章 JavaScript 语法基础.....	9
2.1 JavaScript 语言结构	9
2.1.1 标识符	10
2.1.2 文字和变量	10
2.1.3 函数.....	11
2.1.4 对象和性质	11
2.2 数据类型	11
2.2.1 整型或全数字	12
2.2.2 浮点数或小数	14
2.2.3 字符串或字符序列	15
2.2.4 Booleans 或 true/false 数值	15
2.2.5 特殊字符	15
2.3 变量声明	16
2.3.1 全局变量和局部变量.....	16
2.3.2 变量声明的一些说明	16
2.3.3 变量声明的实例	17
2.4 使用表达式和运算符	19
2.4.1 表达式	19

2.4.2 运算符	19
2.4.3 运算符的优先级	22
2.5 编程语句	23
2.5.1 条件测试语句	23
2.5.2 重复操作语句	25
2.6 计算实例	27
2.6.1 实例一 使用条件测试语句或逻辑运算符判断是否闰年	27
2.6.2 实例二 使用 for 循环语句生成 Fibonacci 数列	29
2.6.3 实例三 使用 while 循环进行迭代求平方根数值计算	31
2.7 小结	32
第 3 章 在自己的主页中加入弹出窗口	33
3.1 面向对象和基于对象的编程技术基础	34
3.2 在自己主页中实现弹出窗口	36
3.2.1 JavaScript 对象模型中的 Window 对象	36
3.2.2 JavaScript 对象模型中 Window 对象的性质	37
3.2.3 JavaScript 对象模型中的 Window 对象的方法	38
3.2.4 使用 Window 对象的方法在自己的主页中加入弹出窗口	39
3.2.5 JavaScript 对象模型中的 Window 对象的事件及事件句柄	42
3.2.6 使用 Window 对象的事件句柄在自己的主页中加入弹出窗口	43
3.2.7 弹出窗口实现所涉及的其它因素	45
3.3 小结	46
第 4 章 使用窗体进行页面设计	48
4.1 创建 HTML 窗体	49
4.1.1 <FORM>	49
4.1.2 <TEXTAREA>	49
4.1.3 <SELECT>	50
4.1.4 <INPUT>	51
4.1.5 实例: 创建 HTML 窗体	52
4.2 窗体对象及其对象元素的属性、方法和事件句柄	56
4.2.1 Form 对象的属性、方法和事件句柄	56
4.2.2 窗体元素的属性、方法和事件句柄	57
4.3 窗体设计与 JavaScript 的结合	63
4.3.1 自动计算订货单	63
4.3.2 创建交互窗体	70
4.4 小结	77
第 5 章 给 Web 页面增加导航	79
5.1 使用按钮进行页面导航	80

5.1.1 使用窗体的 ACTION 属性.....	80
5.1.2 使用按钮的 onClick 事件.....	82
5.2 使用选择列表.....	83
5.3 自动导航.....	90
5.3.1 Window 对象定时器方法.....	90
5.3.2 自动导航.....	91
5.4 小结.....	95
第 6 章 给 Web 页增加信息提示.....	96
6.1 使用状态栏进行信息提示.....	96
6.1.1 用状态栏实现静态信息提示.....	96
6.1.2 用状态栏实现滚动信息提示.....	99
6.1.3 滚动实例中字符串对象的使用.....	101
6.2 使用文本框进行信息提示.....	103
6.2.1 使用文本框实现信息提示的一种方法.....	103
6.2.2 用户自定义对象.....	105
6.3 使用 Netscape/Microsoft 扩展功能.....	109
6.3.1 Netscape 扩展.....	110
6.3 小结.....	116
第 7 章 使用通用内置对象.....	117
7.1 使用 Array 对象.....	117
7.1.1 创建数组的一些说明.....	117
7.1.2 实例: 创建用户数组.....	118
7.2 使用 String 对象.....	121
7.2.1 String 对象的方法.....	121
7.3 使用 Date 对象.....	127
7.3.1 Date 对象的创建.....	127
7.3.2 Date 对象的方法.....	127
7.4 Math 对象.....	129
7.5 小结.....	132
第 8 章 在 Web 页面中加入框架.....	134
8.1 框架的创建.....	134
8.1.1 FRAMESET 标记.....	135
8.1.2 FRAME 标记.....	136
8.1.3 NOFRAMES 标记.....	137
8.2 框架的目标超链接.....	137
8.3 JavaScript 与框架的结合.....	139
8.4 框架间管理的实例.....	141

8.5 小结	144
第 9 章 在 Web 页中加入记忆块 (Cookie)	146
9.1 Cookie 的基本概念	146
9.2 Cookie 的两面性以及 Cookie 的去除	147
9.3 使用 Cookie 保存窗体的内容	148
9.4 使用数组建立 Cookie	151
9.5 小结	152
第 10 章 在 Web 页中加入图像	154
10.1 在 Web 页中使用图像以及与 JavaScript 的结合	154
10.1.1 在 Web 页中使用图像的一般性原则	154
10.1.2 Web 页面中加载图像的 HTML 格式	156
10.1.3 在 JavaScript 中使用图像	156
10.2 使用 JavaScript 创建页面特殊效果实例: 鼠标跟踪	160
10.2.1 图像文件的预处理	160
10.2.2 不同浏览器平台的处理	161
10.2.3 层或风格单的创建	161
10.2.4 获取鼠标位置以及层或风格单的移动	161
10.2.5 程序一览	162
10.3 使用图像映像	164
10.3.1 定义客户方的图像映像	165
10.3.2 引用客户方图像映像	165
10.3.3 使用客户方图像映像的实例	166
10.4 小结	170
第 11 章 在 Web 页面中加入多媒体	172
11.1 在 JavaScript 使用声音和视频	172
11.1.1 在 Web 页中加入音频文件	172
11.1.2 在 Web 页中加入视频文件	175
11.1.3 在 Web 页中加入背景音乐	176
11.2 插件的使用	177
11.2.1 在 Netscape 中使用插件	178
11.2.2 在 Internet Explorer 中使用插件	178
11.3 插件和 JavaScript	179
11.4 使用插件相关对象实例: 测试浏览器的音频播放功能	181
11.5 小结	183
第 12 章 制作 Web 页面附件	184
12.1 Web 页面日历记事本	184

12.1.1 当前时间信息的获得.....	185
12.1.2 日历记事本的绘制.....	185
12.1.3 记事功能的实现.....	185
12.1.4 补充内容.....	193
12.2 Web 页面计算器.....	195
12.2.1 创建标准型计算器界面.....	195
12.2.2 窗体元素与 JavaScript 的关联.....	197
12.3 Web 页面检索器.....	200
12.3.1 组成文件.....	201
12.3.2 关键字数据数组的建立.....	204
12.3.3 用户输入关键字处理.....	204
12.3.4 查询结果的处理.....	205
12.4 小结.....	206
第 13 章 Web 页面文字特效.....	207
13.1 文字特效预处理部分.....	208
13.2 主体控制函数.....	209
13.2.1 start () 函数.....	209
13.2.2 stop () 函数.....	209
13.2.3 resz () 函数.....	209
13.3 文本运动控制函数.....	210
13.3.1 up () 函数.....	210
13.3.2 down () 函数.....	210
13.4 事件句柄与函数的连接.....	210
13.5 程序源代码及运行结果.....	210
13.6 JavaScript 风格单.....	219
13.6.1 在 HTML 中包含风格.....	220
13.6.2 JSSS 语法.....	221
13.6.3 使用 JSSS 的实例.....	226
13.7 小结.....	228
第 14 章 Web 游戏制作.....	229
14.1 制作基本功 I——使用事件属性和高级事件.....	229
14.1.1 Navigator4.x 新增事件.....	230
14.1.2 事件对象及其属性——事件属性.....	232
14.2 制作基本功 II——事件对象.....	233
14.2.1 事件对象的属性.....	233
14.2.2 捕获事件对象.....	236
14.2.3 有关事件处理的其它方法.....	238
14.2.4 使用事件的捕获进行模拟游戏制作.....	238

14.3 Web 页面游戏制作实例：俄罗斯方块.....	241
14.3.1 程序组成	242
14.3.2 程序规划	242
14.3.3 完整的应用程序	243
14.3.4 程序的工作过程	257
14.4 小结	260
第 15 章 JavaScript 脚本调试	261
15.1 JavaScript 编程中常见错误	261
15.1.1 语法错误	261
15.1.2 运行错误	263
15.1.3 HTML 错误	263
15.1.4 其它的一些常见问题	264
15.2 调试技术	267
15.2.1 使用 alert () 方法	267
15.2.2 创建自己的跟踪调试函数	268
15.3 JavaScript 调试工具	269
15.3.1 JavaScript 命令行	269
15.3.2 使用 Netscape Visual JavaScript 进行脚本调试	270
15.4 脚本编制的一些建议	272
15.5 处理缺陷 (Bugs)	273
15.5.1 MSIE3.0	273
15.5.2 MSIE4.0	273
15.5.3 NS2.0	273
15.5.4 NS3.x	273
15.5.5 NS4.x	274
15.6 小结	274
第 16 章 JavaScript1.3 中的新特性	275
16.1 JavaScript 和 ECMA	275
16.2 JavaScript1.3 新特性	277
16.2.1 Unicode	277
16.2.2 新顶层属性	278
16.2.3 全局函数 isFinite ()	279
16.2.4 toSource () 方法	279
16.2.5 Date 对象的一些新增特性	280
16.2.6 函数对象的新方法：call () 和 apply ()	281
16.2.7 严格等同操作符	282
16.3 JavaScript1.3 的变化	283
16.3.1 等同操作符	283

16.3.2 数组.....	283
16.3.3 String 对象的 replace () 方法	284
16.3.4 布尔对象	285
16.3.5 toString () 方法	285
16.4 关键字和字面量	286
16.4.1 JavaScript1.3 新增保留字	286
16.4.2 JavaScript 规定	286
16.5 JavaScript 控制台	288
16.6 JavaScript 版本问题	289
16.7 小结	290
附录 A JavaScript 开发工具	291
附录 B JavaScript 语言参考	301
附录 C 颜色名称及其十六进制值	310

第 1 章 JavaScript 简介

本章要点:

- ◇ 了解 JavaScript 的历史
- ◇ 了解 JavaScript 的特点, 包括其优点及局限之处
- ◇ JavaScript 与其它语言比较
- ◇ 初步掌握建立一个包含 JavaScript 脚本的 HTML 文档的基本步骤

在正式学习之前, 先对 JavaScript 的概念、功能及特点等有一个大致的了解对其后的学习是很有必要的, 同时这对于您的 Web 制作中诸如工具的选择等工作也是不可缺少的。本章将对以上内容作较详细的介绍。

1.1 JavaScript 概念

JavaScript (Microsoft 版称做 JScript) 是一种脚本制作语言, 由于具有开发简单、功能灵活等特点, 现今已在 Internet 上得到了广泛的应用。JavaScript 可用来处理 Web 上的事件, 将对象集成, 并可给页内容本身编写脚本, 从而一改 Web 页以往“只读”特性, 为 Internet 上交互提供了方便的一笔。

1.1.1 JavaScript 历史

JavaScript 是一种简单而又灵活的脚本语言, 它的创立要从 Java 的历史说起。1991 年 Sun Microsystem 公司的一位工程师 James Gasling 创建了一种能用来给蜂窝电话及电视遥控器之类设备编程的语言 Oak, 但在 1993 年 Sun 公司采用该技术建立遥控器样机时发现, “Oak”已经作为一个商标在使用。待到 1994 年 Internet 已在全球广为流行时, Oak 小组开始认识到它的可下载技术可用在 Web 上, 于是决定在已有工作基础上研制一种新的 Web 语言, 同时将这种语言改名为“Java”。Java 由此诞生。

使用过 Java 的人都知道, Java 语言是面向对象的, 很类似 C++。它吸收了 C++ 的许多优点, 同时也作了很多简化以易于编程。但即便如此, 程序员本身还是要做很多基础性的工作, 比如对象的定义及扩展。这样使得程序开发周期变得较长。同时, 虽然在 1995 年时 Java 已成为 Internet 上的最热门的新词, 但很少有人真正了解如何用它来编程。Netscape Communications 认识到人们需要有一种用于 Web 页的简单明了的语言, 于是引入 LiveScript, 这即是 JavaScript 的前身。1995 年底, Netscape 公司与 Sun 公司达成协议, Sun 公司也认识到开发一种简单的脚本语言的必要性, 于是认可 LiveScript, 同时出于知名的原因便将其名称改为我们现在所熟知的 JavaScript。

1.1.2 JavaScript 特性

JavaScript 简化的语言结构使得改变一个脚本就像改变一个 HTML 页一样简单,同时不需要创建新类而仅仅访问 JavaScript 中已经存在的对象就可以完成相当的功能。它一问世便以其简单明了受到广大网页制作爱好者们的青睐,它使得非专业 Web 程序员也能轻松明晰地为自己的网页增加交互的额外特性。如果您已有一门编程语言基础并且对 HTML 也有所了解,JavaScript 应该是让您感到轻松而兴奋的。HTML 给网页制作提供了灵活性,但 HTML 本身是静态的,一旦 HTML 文档被写成,它便不允许用户干预。加入 JavaScript 后,可以给网页制作者提供一种编写在用户浏览器而不是服务器上运行的小脚本的方法。例如,从窗体 (Form) 中收集数据并在数据传送到服务器之前,可以用 JavaScript 小程序校验数据的完整性和合理性之后再发送到服务器,这样可大大改善浏览过程的执行情况。

通常,JavaScript 的长处是用于客户端的编程,但它还是提供了一些工具使 JavaScript 的浏览器编程功能能满足一些服务器方编程要求。尽管如此,JavaScript 还需要有一些服务器方程序来支持更高级的 Web 应用程序,比如访问数据库信息、支持电子商务或进行特殊处理。为了帮助开发人员建立这些服务器程序,Netscape 公司提供了 LiveWire 和 LiveWire Pro 工具集,它们提供了开发 JavaScript 服务器程序的集成环境。

下面列出了 JavaScript 的一些优点:

- 可以将 JavaScript 与嵌入 Web 的大多数对象 (甚至是位置点 (target)) 的事件 (例如鼠标单击、移动等) 相关联,然后用您自己的方式处理这些事件。
- 用 JavaScript 可建立活动的 HTML,即是说将浏览器装入 Web 页时,JavaScript 可产生 HTML。影响所产生 HTML 的变量可能是如记忆块 (Cookie) 或所使用的浏览器类型之类的东西。
- 使用 JavaScript 可实现原来需要服务器方能处理的许多特性。如使用处理图像的鼠标单击事件的脚本,可以创建客户方的图像映像。
- JavaScript 被嵌入到 HTML 文件中,因此对脚本的编辑、运行以及复制是很容易的。与 Java 及 ActiveX 控制不一样,您不必担心代码会传到自己的机器当中,也不必担心<OBJECT>标记是否正确地插到 Web 页当中。使用 JavaScript 所需要的一切就是好的文本编辑器和一小点技巧。

1.2 JavaScript 与其它语言比较

在开始自己的 Web 页制作之前,慎重地选择一个好的制作工具是必要的。大多数情况下,您不需要创建 ActiveX 对象或者是制作 Netscape 插件这类繁复的工作,而需要的仅仅是脚本制作从而改善 Web 页的生动性、交互性而已。然而脚本制作是用 CGI、VBScript 还是 JavaScript? 下面的比较也许会在以上的决择中给您提供一些帮助。

1.2.1 JavaScript 和 CGI

CGI (Common Gateway Interface, 通用网关接口) 是服务器方脚本制作接口标准,它一般用在服务器上作窗体数据处理,其具体实现可用 C、Shell 或者是 Perl 等语言。CGI 的出

现使得任何应用工程师都可能将应用程序与服务器相接。不要以为像 JavaScript 和 Java 这样的新一类语言出现使得 CGI 显得过时,实际上,CGI 能完成的工作 JavaScript 大都无法实现,因为 JavaScript 与 CGI 相比有以下致命弱点:

- JavaScript 只能在客户机上运行,CGI 则在服务器上运行。
- CGI 返回整个文档,而 JavaScript 可能只在页上的一个小范围起作用,如一个文本域或状态栏。
- CGI 可访问一些诸如用户 IP 地址这样的信息,而 JavaScript 却不行。
- JavaScript 可以访问诸如当前页的连接等信息,而 CGI 却不行。

1.2.2 JavaScript 和 Java

Java 是一种面向 Web 的语言,像传统 C 和 C++ 语言一样,用 Java 编写的程序必须通过编译才能运行,同时还要使它在服务器上可用。而 JavaScript 是一种不严格的以 Java 为基础的解釋性语言,JavaScript 脚本程序被存储在 HTML 页的资源窗口中。运行时,具有 JavaScript 的页被下载到客户机上并且 JavaScript 被解释执行。

除了以上区别,JavaScript 和 Java 还有以下几点不同:

- 静态连接在 JavaScript 中,您可以指向一个对象,同时该对象可以在脚本载入并查错时不存在。而在 Java 中,只有在所有对象都存在后才进行编译。
- 面向对象 JavaScript 实际上是基于有限的对象模型,而 Java 则进了一大步,它包含了真正面向对象编程的两个最重要的特征:类和继承。
- 图形和 GUI 能力:Java 提供了 JavaScript 所不具备的产生 GUI 要素的图形原始处理功能。

1.2.3 JavaScript 和 VBScript

VBScript 是基于 Microsoft Visual Basic 的脚本制作语言,同 JavaScript 一样,它可以用来创建客户方的脚本程序,以处理 Web 页上的事件,集成 Web 页上的对象和创建动态内容等等。您如有 Visual Basic 编程经验,那么 VBScript 可以使您在 Web 页脚本制作时重温 Basic 的驯服与便利。但 VBScript 有一个致命的缺点,那就是目前 Microsoft Internet Explorer 是唯一支持 VBScript 的浏览器。也许在高版本的 Netscape Communicator 会有所改善,但无论怎样,在 Internet 上,兼容性是不得不首先考虑的大问题。

1.3 JavaScript 用途

JavaScript 提供了丰富的内置函数及命令,它能在浏览器中显示 HTML、做商务数值计算、多媒体播放、超级链接以及简单的交互窗口等。本书将在其后的章节中逐步给出这些方面的实例。

JavaScript 还可以使在浏览器中运行的小 Java 应用程序的属性或性质改变,这样就能很容易地改变控件或其它对象的行为而不必深入研究其本身结构。例如,当您单击按钮时,便可自动播放嵌入的 QuickTime 或 AVI 文件。

1.4 编写第一个 JavaScript 程序

在这一小节中，我们将一步步完成一个完整 JavaScript 程序的编制，虽然只是一个示范程序，不能完成很多的实际工作，但通过它的编制和运行可以让我们对脚本的创建、嵌入 HTML 页、运行以及简单调试有一个初步的了解。随着学习的深入，您将会看到更多更复杂的例子。

1.4.1 创建 JavaScript 唯一真正需要的工具

由于 JavaScript 是一种面向对象的文本脚本语言，使用的标准 ASCII 文本，所以建立 JavaScript 程序真正需要的只是一个好的文本编辑器，如果您的操作系统是 Windows 9x 或是 Windows NT，那么 NotePad 就是一个很不错的文本编辑器。

当然，好的工具会使我们的工作变得更为简洁明了，一套精选的软件工具可以使建立 HTML 文档及 JavaScript 程序更加容易。本书附录 A 大致介绍了几种加快和简化 JavaScript 程序编制和嵌入 HTML 文档的工具。

对于 JavaScript 的运行，由于 JavaScript 程序解释器是内置在浏览器自身当中，所以运行 JavaScript 程序除了一套支持 JavaScript 的浏览器程序之外，无需其它的编译运行软件。现今两大流行浏览器 Netscape Communicator(较新版本 4.5、4.6)和 Microsoft Internet Explorer(较新版本 5.0)对 JavaScript 的许多对象方法进行了扩充，兼容性也很好，是装载运行 JavaScript 理想的浏览器软件。

1.4.2 创建 JavaScript 脚本

首先看一个简单的 JavaScript 应用程序，它完成了一段文字的输出。

程序清单 1-1 用 JavaScript 进行文字输出

```
document.write("<B>This begins our JavaScript exploring!</B>"+"<BR>")
document.write("\n<I>It is too simple you'll see.</I>")
```

后面第三章对象和方法我们将进一步学习 document 对象及其 write 方法的使用。在完成上述脚本的编制后，下面的工作就是将其嵌入 HTML 文档中。

1.4.3 脚本的嵌入

JavaScript 命令可以直接用<SCRIPT>标记和其 LANGUAGE 属性将脚本嵌入 HTML 文档内，其中 LANGUAGE 属性说明解释脚本所要使用的脚本语言（例如 JavaScript 或 VBScript）。能够放<SCRIPT>...</SCRIPT>脚本段的地方一般有<BODY>...</BODY>超文本文档主体段，也有放<HEAD>...</HEAD>头段的。如放在文档主体段，Web 页被载入的同时脚本立即被执行，程序清单 1-1 中的两句输出将出现在页中；如果放在文档头段，则可以百分之百地保证脚本函数在使用之前都已被声明。这在函数是通过事件句柄而激发的情况下显得尤为重要。当在一个框架集（frameset）中使用 JavaScript 时，您必须把所有的 JavaScript 代码放在头段，否则便不能保证框架正确地显示，这点在 Internet Explorer 中更是这样。

还有一种方法是用脚本将要装载的 URL 嵌入 HTML 文档。这即是利用<SCRIPT>的另

一属性 SRC。SRC 属性说明来自于可能要装载的脚本的 URL。除非 SRC 属性的 URL 说明了一种语言, 否则 LANGUAGE 属性是必需的。LANGUAGE 和 SRC 两属性也可同时使用。下面是将程序清单 1-1 中 JavaScript 脚本嵌入 HTML 文档两种方法的完整源程序。

程序清单1-2 使用直接嵌入法

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <title>JavaScript Exploring</title>
</head>
<body>
  <script language="javascript">
document.write("<B>This begins our JavaScript exploring!</B>"+"<BR>")
document.write("\n<I>It is too simple you'll see.</I>")
  </script>
</body>
</html>
```

说明:

LANGUAGE 属性值可以设为 JavaScript、JavaScript1.1、JavaScript1.2 或者 JavaScript1.3。如果不指明 JavaScript 的版本号, Web 浏览器将以最近的 JavaScript 版本对其进行解释。对于使用 JavaScript 较新版本中向下不兼容特性的一些脚本程序, 应当指明 JavaScript 的版本号, 以防早期浏览器加载它时产生错误。

如果采用第二种方法, 则需先将程序清单 1-1 中 JavaScript 脚本程序段存为 c:\My Documents\javascript\jsexploring.js, 然后也可这样引用:

程序清单1-3 使用<SCRIPT>的SRC属性法

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <title>JavaScript Exploring</title>
</head>
<body>
  <script src="file:///c:/My Documents/javascript/jsexploring.js" language="javascript">
  </script>
</body>
</html>
```

注意:

如果您对 URL 的文件格式 file:///c:/My Documents/...还不熟悉, 请参阅有关 Internet 的书。

如果您使用 Navigator 3.0 以前的版本, 使用 SRC 属性的方法是不被支持的, IE 3.0 也不支持 SRC 属性。这在当时是出于安全的原因。随着浏览器安全机制的发展, 更高版本的

Navigator (或是 Communicator) 和 Internet Explorer 能够开始采用 SRC 属性。

上例中 c:/My Documents/javascript/jsexploring.js 是 JavaScript 程序文件。文件名是随意的, 但其扩展名必须是.js。

1.4.4 程序的输出

出于兼容性以及属性扩展的原因, 您在运行 JavaScript 程序时应尽量选用高版本浏览器。运行 JavaScript 程序非常简单, 只需让浏览器加载包含脚本的 HTML 页即可。由于 JavaScript 是在客户机上被解释执行, 所以您可以放心地在自己本地机上测试所有 JavaScript 例子, 记住 JavaScript 可以是与服务器完全无关的。

1.4.5 注释与兼容问题

在程序中使用注释说明该程序是一种好习惯。在 JavaScript 中, 注释有两种, 分别是单行注释和多行注释。

单行注释采用双斜杠 (//) 开始并被限制在一行以内, 多行注释则以 /* 开始, 在最后以 */ 结束, 这种注释方法对于有 C++ 编程基础的读者来说应该是相当熟悉的。下面程序清单 1-4 显示了几个注释范例。

程序清单1-4 JavaScript注释范例

```
// This is a legal comment
/ illegal - - comments start with two slashes
/* multiple line comments can be spread across more than one line,
   as long as they end. */
/* illegal - - this comment doesn't have an end!
// this is OK - - as extra slashes are ignored //
```

虽然设计 JavaScript 程序的初衷都是为兼容 JavaScript 的浏览器而创作, 但为了防止不支持 JavaScript 的浏览器把 JavaScript 命令解释成 HTML 并同时显示它们, 可以设计将 JavaScript 放在 HTML 注释中。下面程序清单 1-5 给出了例子。

程序清单1-5 对非JavaScript浏览器隐藏脚本

```
<SCRIPT LANGUAGE = "JavaScript">
<!-- this line opens an HTML comment
    document.write("You can see this script's output, "+~but not it's source. ~");
// this line closes the HTML comment -->
</SCRIPT>
```

注意:

在程序清单 1-5 中, 结束注释行使用双斜杠 (//) 开始, 这是 JavaScript 注释标记符。这里是必需的, 因为一旦 JavaScript 开始处理在<SCRIPT>标记区域内的行, 它将假定直到</SCRIPT>关闭标记之前的所有行都是 JavaScript 代码。双斜杠有效地对试图解释该行的 JavaScript 隐藏了注释关闭标记。

注释的开始 <!-- 和注释结束 --> 是 HTML 3.2 以及 HTML 4.0 的标准; 在以前, HTML 2.0 版本则以 <!! 和 -> 分别定义一个注释的开始和结束。但从另一方面讲, 不要试图用这种方