

网页设计尖峰之旅丛书

JavaScript

实例教程

赵丰年 编著



光盘包括微软授权 Windows 2000 Server 免费软件，
本书的 Web 版和全部源代码，大量相关的共享或免费工具软件。



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: [http:// www.phei.com.cn](http://www.phei.com.cn)

网站建设尖峰之旅丛书

JavaScript 实例教程

赵丰年 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书通过大量实例介绍了 JavaScript 的基础知识和实际应用,使读者可以按部就班地系统掌握 JavaScript 客户端编程技术。全书共分为 9 章,分别介绍了 JavaScript 语言基础、JavaScript 对象、JavaScript 事件处理、文档对象、窗口与浏览器、表单对象、链接与图像、DHTML 基础以及 DHTML 应用。

本书结构严谨,内容丰富,适合各层次的网页设计人员学习使用,并可以作为相关课程或培训的参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

JavaScript 实例教程/赵丰年编著. - 北京:电子工业出版社,2001.1

(网站建设尖峰之旅丛书)

ISBN 7-5053-6272-0

I. J... II. 赵... III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 80991 号

丛 书 名:网站建设尖峰之旅丛书

书 名:JavaScript 实例教程

编 著 者:赵丰年

策 划:何芳芳

责任编辑:吕 迈 何芳芳

特约编辑:谢 培

排版制作:电子工业出版社计算机排版室

印 刷 者:北京东光印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:21.25 字数:541 千字 附光盘:1 张

版 次:2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

书 号:ISBN 7-5053-6272-0
TP·3384

印 数:8 000 册 定价:37.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话 68279077

JS422/26

技术先锋 尖峰之旅

——《网页设计尖峰之旅丛书》与《网站建设尖峰之旅丛书》序

我们有幸生逢技术变革不断发生的伟大时代。几年来,信息领域里最为振奋人心的事情,便是 Internet 的迅速拓展和普及应用。Internet 作为一个蕴含巨大信息资源和人类智慧的网络空间吸引了越来越多的人。今天,Internet 的应用已经与人类生活密切相关,尤其是由信息技术推动和基于 Internet 而兴起的电子商务,已经成为人们普遍接受的一种崭新的商务方式。电子商务能够提供准确、快速、高效的商务环境,能使人们更高效、更省力、更省钱地从事社会和生产活动,代表了当今世界商务模式发展的主流方向。目前,全世界很多国家和地区都在大规模地发展电子商务,以求能为传统商务活动开创新的发展机遇,并为人们提供反应迅速、成本低廉的交易模式。

在电子商务建设中,最关键、最核心的任务是技术平台的建设,其中涉及的技术主要包括三个方面,即面向客户端的网页制作技术,面向服务器端的网站建设和 Web 数据库技术,以及实现客户端与服务器端相互关系的连接和集成技术等。为了帮助广大网页设计和网站建设的专业技术人员快速而深入地掌握这三方面的技术知识,我们组织有关大学教师和资深技术专家编写了“网页设计尖峰之旅”与“网站建设尖峰之旅”两套丛书。

这两套丛书基本上涵盖了当前电子商务建设中的主流技术与软件。其中,“网页设计尖峰之旅丛书”主要侧重于网页设计领域的流行软件和主流编程技术,如《Dreamweaver 3.0 实例教程》、《Fireworks 3.0 实例教程》、《Flash 4.0 实例教程》、《FrontPage 2000 中文版实例教程》、《VBScript 与 JScript 实例教程》、《网页制作实用素材集 I/II》、《Dreamweaver 2.0 实例教程》、《Fireworks 2.0 实例教程》等;“网站建设尖峰之旅丛书”主要侧重于网站建设、Web 数据库系统及网络系统连接与集成领域的流行软件和主流编程语言、开发技术,如《Web 数据库开发技术集成实战演练》、《ASP 实例教程》、《PHP 4 & MySQL 完全实例教程》、《Perl 5、PHP 4 与 CGI 实例教程》等。每一本新书的封底都将有《尖峰之旅丛书》最新成员的身影!

两套丛书紧紧围绕“深入、实用、精炼”的创作主题,以轻松、简练、明快的行文笔调,深入浅出地讲解各书相应的内容,语言流畅、活泼、通俗易懂,举例新颖、实用并易于操作,能使学习者在比较短的时间里快速学习到丰富的软件使用方法和开发技术,收到卓有成效的学习效果,并迅速提高自己的技术开发水平。

两套丛书特色和风格是一致的,从总体上来讲,主要包括以下 5 个方面:

其一,立足技术前沿。两套丛书所选软件和编程语言、数据库系统等都是国内外著名软件公司的知名产品,也是当前国内应用最新、最广泛的产品。

其二,读者定位明确。丛书中“实例教程”类主要针对电子商务的初、中级专业技术人员,“实战演练”类主要针对电子商务的中、高级专业技术人员。

其三,实例讲解内容。两套丛书均注重通过大量的实例,以实际的技术问题和开发环境,讲解开发案例、编程方法、软件使用技术与操作技巧等。

其四,版式灵活醒目。两套丛书在力求文字精炼、脉络清晰的同时,注意通过大量的图表直观地说明问题,并在正文之外设计了一些特殊的段落,来讲解具有技巧性、提示性或重要性的内容。我们希望通过这种灵活醒目的版式,减小读者的阅读难度,增加学习兴趣,真正让学习变成一种乐趣。

其五,随书附带光盘。为了节省读者的学习和练习时间,两套丛书的每一部书都配有光盘。光盘包括3部分内容:多媒体操作指导教程、书中所有举例及例程源代码、相应的系统软件(由出品公司授权)及常用工具软件(共享或免费)。

由于技术在不断地发展,所以我们这两套丛书也采取了不断进取、不断推陈出新的创作与出版思路。也就是说,在已推出各书的基础上,一方面我们要继续创作和出版网络技术领域中其他流行软件和主流编程语言、数据库开发技术的“实例教程”或“实战演练”;另一方面,当软件或语言版本升级时,我们将及时推出更新版本的相应书籍,以保证我们丛书“立足技术前沿”的鼎力追求!

“严谨、实用、高质量、新技术”是我们两套丛书出版中孜孜以求的目标,尽管我们精心而为,刻意而志,但书中错误和不足之处难免,所以恳请广大读者不吝批评指正,多提宝贵意见。我们一定会认真听取读者心声、意见和建议,把后续工作做好,把两套丛书出好!

巅峰风景无限好。紧跟时代潮流,勇攀技术高峰,打造技术先锋,请与我们共赴尖峰之旅!

电子工业出版社

前 言

随着 Internet 的飞速发展,我们已经深深地陷入了“网”中央。在这一“网”无尽的 Internet 海洋中,我们一方面尽力在其中遨游,一方面又渴望亲手编织这张网。在这样的潮流下,不但计算机专业人员争相涌入网页设计和编程这一热门行业,而且许多普通网民也希望通过网络来表现自己甚至挖掘商机。于是,网页设计技术几乎成为一种时尚。

虽然使用 HTML 语言或者 FrontPage、Dreamweaver 等设计工具已可以设计出丰富多采的网页,但如果要创建出功能更强大、更吸引人的网页,就还需要更强大的工具,也就是说,我们必须过渡到网页设计的第二个层次——客户端脚本编程。通过客户端脚本编程,我们可以使网页具有更强的交互性和动态特征,甚至达到与应用程序类似的效果,而 JavaScript 可以说是客户端脚本的最佳选择。一方面 JavaScript 得到绝大多数浏览器的支持,另一方面其类 C 特性也使众多具有 C 语言编程经验的用户能够更快地掌握它。

本书正是在这种背景下应运而生,以帮助读者系统而全面地掌握 JavaScript 客户端脚本编程这一实用技术。全书在逻辑上可以分为三部分:第一部分是前三章,包括“JavaScript 语言基础”、“JavaScript 对象”和“JavaScript 事件处理”,介绍了 JavaScript 语言的基础知识和如何在网页中使用该脚本语言;第二部分是中间四章,包括“文档对象”、“窗口与浏览器”、“表单对象”和“链接与图像”,介绍了各种浏览器对象,使读者可以充分利用它们的属性和方法编写出符合自己需要的网页;第三部分是最后两章,包括“DHTML 基础”、“DHTML 应用”,介绍了 JavaScript 的一个重要的应用领域——DHTML。除了正文之外,本书的两个附录:“HTML4.0 快速参考”和“CSS 属性参考”也为读者制作网页提供了很大的方便。

本书的一个典型特点是其实用性,书中的绝大多数示例都是根据实际需要编写或改编的。例如:导航列表,网页导航条,翻滚图,JavaScript 动画,动态折叠菜单等示例都可以直接应用于读者自己的网页中。书中的所有示例均在 Internet Explorer 5.0 中进行了严格的测试,读者可以放心使用。

本书的读者对象是:了解一般的 Internet 常识,具有 HTML 语言基础的网页设计和编程人员。如果读者具有 C 语言的编程基础,那么有关语言基础的部分学习起来就不会有什么障碍,否则需要详细阅读第 1 章。另外,鉴于国内大多数浏览者使用的都是 Internet Explorer,因此本书默认的目标浏览器是 IE。如果需要在多浏览器平台上使用本书中的示例,则可能重新编写和测试。

建议读者在使用本书的过程中采用边读边实践的方式,这样不但可以学得更有效率,也会学得更有乐趣。

本书由赵丰年编写,参加相关工作的还有胡长清、赵承志、赵念东、刘浩、谢小强、白峰、孙志勇、李伟、吴亚东、孙海、陈熙、李江军、刘东、曹海燕和李涛等。

由于时间仓促,书中疏漏和不妥之处在所难免,希望广大读者谅解。欢迎读者就书中的问题与作者讨论,作者的电子邮件地址是:zhaofengnian @ 263.net。同时欢迎读者访问作者的个人网站(<http://zhaofengnian.yeah.net>),在该网站上可以下载本书中的所有源代码和其他一些相关参考资料。

编 著 者

2001 年 1 月

目 录

第 1 章 JavaScript 语言基础	(1)
1.1 什么是 JavaScript	(1)
1.1.1 JavaScript 的基本特点	(1)
1.1.2 JavaScript 与 JScript	(2)
1.1.3 JavaScript 应用	(3)
1.1.4 为什么要使用 JavaScript	(4)
1.2 在 Web 页中使用 JavaScript	(5)
1.2.1 嵌入 JavaScript	(6)
1.2.2 链接 JavaScript	(9)
1.2.3 确保兼容性	(10)
1.3 JavaScript 变量	(11)
1.3.1 变量的命名约定	(11)
1.3.2 变量的类型	(12)
1.3.3 变量的作用域	(18)
1.4 JavaScript 运算符	(20)
1.4.1 运算符与表达式	(20)
1.4.2 算术运算符	(21)
1.4.3 比较运算符与逻辑运算符	(23)
1.4.4 字符串运算符	(26)
1.4.5 位操作运算符	(27)
1.4.6 赋值运算符	(29)
1.4.7 条件运算符	(31)
1.4.8 其他运算符	(32)
1.4.9 运算符的优先级	(32)
1.5 JavaScript 语句	(34)
1.5.1 概述	(34)
1.5.2 条件语句	(34)
1.5.3 循环语句	(38)
1.5.4 其他语句	(45)
1.6 JavaScript 函数	(46)
1.6.1 使用函数	(46)
1.6.2 JavaScript 全局函数	(49)

第2章 JavaScript 对象	(54)
2.1 什么是对象	(55)
2.1.1 对象的属性与方法	(55)
2.1.2 基于对象的 JavaScript	(55)
2.2 使用 JavaScript 对象	(56)
2.2.1 内置对象与浏览器对象	(57)
2.2.2 使用自定义对象	(58)
2.2.3 对象运算符与语句	(61)
2.3 Array 对象	(64)
2.3.1 创建数组	(64)
2.3.2 Array 对象的属性和方法	(68)
2.4 Boolean 对象	(71)
2.5 Date 对象	(73)
2.5.1 创建日期对象	(73)
2.5.2 Date 对象的属性和方法	(75)
2.6 Function 对象	(80)
2.7 Global 对象	(81)
2.8 Math 对象	(83)
2.8.1 Math 对象的属性与方法	(83)
2.8.2 示例	(84)
2.9 Number 对象	(86)
2.10 Object 对象	(88)
2.11 RegExp 对象	(89)
2.12 String 对象	(90)
2.12.1 创建 String 对象	(90)
2.12.2 字符串格式设置	(92)
2.12.3 通用字符串操作	(94)
第3章 JavaScript 事件处理	(97)
3.1 什么是事件	(97)
3.2 处理 JavaScript 事件	(98)
3.2.1 事件处理属性	(98)
3.2.2 事件处理函数	(102)
3.2.3 通过对象指定事件处理函数	(105)
3.3 event 对象	(106)
3.3.1 event 对象的属性	(106)
3.3.2 事件浮升	(111)
3.4 错误处理	(114)
3.4.1 error 事件	(114)
3.4.2 错误处理语句	(117)

第 4 章 文档对象	(120)
4.1 浏览器对象简介	(120)
4.1.1 文档对象模型	(120)
4.1.2 对象引用方法	(122)
4.2 document 对象的属性	(122)
4.2.1 属性列表	(123)
4.2.2 用 all 属性访问 HTML 元素	(123)
4.2.3 其他属性示例	(127)
4.3 document 对象的事件	(130)
4.3.1 处理键盘事件	(130)
4.3.2 处理鼠标事件	(132)
4.3.3 处理加载卸载事件	(134)
4.4 document 对象的方法	(135)
4.4.1 方法列表	(135)
4.4.2 方法示例	(135)
第 5 章 窗口与浏览器	(137)
5.1 window 对象的属性	(137)
5.1.1 属性列表	(137)
5.1.2 窗口的状态信息	(138)
5.1.3 窗口代名词	(139)
5.2 window 对象的方法	(142)
5.2.1 方法列表	(142)
5.2.2 打开和关闭窗口	(143)
5.2.3 使用对话框	(147)
5.2.4 定时设置	(151)
5.2.5 其他窗口操作	(155)
5.3 frame 对象	(163)
5.3.1 概述	(163)
5.3.2 属性、方法与事件	(164)
5.3.3 示例	(165)
5.4 navigator 对象	(166)
5.5 screen 对象	(167)
第 6 章 表单对象	(169)
6.1 Form 对象	(169)
6.1.1 属性、方法与事件	(169)
6.1.2 表单处理	(174)
6.2 文本型表单控件	(177)
6.2.1 概述	(177)
6.2.2 属性、方法和事件	(179)

6.2.3 示例	(179)
6.3 单选框与复选框	(182)
6.3.1 概述	(182)
6.3.2 属性、方法与事件	(184)
6.3.3 示例	(184)
6.4 按钮对象	(188)
6.4.1 概述	(188)
6.4.2 属性、方法与事件	(192)
6.5 选项菜单	(193)
6.5.1 概述	(193)
6.5.2 option 对象	(194)
6.5.3 select 对象	(198)
6.6 隐藏字段与 cookie	(200)
6.6.1 使用隐藏字段	(200)
6.6.2 使用 cookie	(205)
第 7 章 链接与图像	(212)
7.1 link 对象	(212)
7.1.1 属性与事件	(212)
7.1.2 示例	(213)
7.2 anchor 对象	(216)
7.3 location 对象	(219)
7.3.1 属性与方法	(219)
7.3.2 示例	(219)
7.4 history 对象	(221)
7.4.1 属性与方法	(221)
7.4.2 示例	(221)
7.5 area 对象	(223)
7.5.1 属性与事件	(223)
7.5.2 客户端图像映射	(224)
7.6 image 对象	(226)
7.6.1 创建 image 对象	(226)
7.6.2 属性与事件	(227)
7.6.3 制作 JavaScript 动画	(230)
第 8 章 DHTML 基础	(237)
8.1 DHTML 概述	(237)
8.2 CSS 基础	(239)
8.2.1 样式定义	(239)
8.2.2 使用样式	(243)
8.3 CSS 属性	(245)

8.3.1	CSS 属性单位	(245)
8.3.2	字体与文本属性	(246)
8.3.3	颜色与背景属性	(251)
8.3.4	布局属性	(253)
8.3.5	定位和显示属性	(261)
8.3.6	列表属性	(265)
8.3.7	鼠标属性	(267)
8.4	过滤器属性	(267)
8.4.1	属性列表	(268)
8.4.2	效果示例	(270)
第 9 章	DHTML 应用	(274)
9.1	style 对象	(274)
9.1.1	概述	(274)
9.1.2	示例	(275)
9.2	动态定位与显示	(279)
9.2.1	动态定位	(279)
9.2.2	显示属性	(285)
9.3	动态过滤器效果	(295)
9.3.1	示例 1	(295)
9.3.2	示例 2	(297)
9.3.3	示例 3	(298)
9.4	DHTML 小脚本应用	(300)
9.4.1	概述	(300)
9.4.2	一个简单的小脚本	(300)
9.4.3	导航条示例	(302)
附录 1	HTML4.0 快速参考	(310)
附录 2	CSS 属性参考	(322)
参考文献		(327)

第1章 JavaScript 语言基础

JavaScript 是目前最流行的 Web 脚本语言，用于开发动态的 Web 页面。作为一种通用的脚本语言，JavaScript 具有所有相关编程语言的特性，例如：通过变量保存数据，通过运算符进行计算，以及通过函数获得程序的模块化和功能的简化等等。本章介绍有关 JavaScript 作为一种编程语言的基础知识，使读者能初步掌握该语言（具有 C 语言基础的读者会发现该语言与 C 十分类似），以便为进一步的学习打下坚实的基础。

本章主要包括：

- 什么是 JavaScript
- 在 Web 页中使用 JavaScript
- JavaScript 变量
- JavaScript 运算符
- JavaScript 语句
- JavaScript 函数

1.1 什么是 JavaScript

JavaScript 是 WWW 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 文档以获得交互式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。

1.1.1 JavaScript 的基本特点

JavaScript 是一种基于对象并具有安全性的脚本语言，具有以下几个基本特点：

- 是一种脚本 (script) 语言

JavaScript 是一种脚本语言, 它采用小程序段的方式实现编程。与其他脚本语言一样, JavaScript 是一种解释性语言, 在程序运行过程中被逐行解释。

- 基于对象

JavaScript 是一种基于对象的语言, 它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。有关 JavaScript 的对象特性, 将从第 2 章开始逐步介绍, 并贯穿于全书。

- 安全性

JavaScript 是一种安全性语言 (有时称为具有 Web 安全特性), 它不允许访问本地的硬盘, 也不允许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。

- 跨平台性

JavaScript 的执行依赖于浏览器本身, 而与操作环境无关——只要是能运行浏览器的计算机, 而该浏览器又支持 JavaScript, 则脚本就可正确执行。

1.1.2 JavaScript 与 JScript

对任何一门编程语言略有了解的读者可能都知道: 编程语言通常分为各种不同的版本。这种不同版本的划分, 要么是基于不同组织对该语言的不同实现, 要么是基于同一组织实现的不同升级版本。对于 JavaScript 而言, 这种版本区分当然也不例外。

JavaScript 的前身是 Netscape 公司开发的 LiveScript, 也就是说, JavaScript 是由 Netscape 开发的。但随着 WWW 在世界范围内的普及, 许多其他大公司也步入了这个领域, Microsoft 公司就是其中的一个。与 Internet Explorer 和 Netscape Navigator 的浏览器大战类似, Microsoft 公司在 JavaScript 的开发上也不甘步人后尘, 于是独自开发了自己的 JavaScript 版本, 通常称为 JScript。

Netscape 在 Navigator 3 中引入了 JavaScript1.1, 在 Navigator 4 中引入了 JavaScript1.2, 而在 Navigator 4.06 和 4.5 中引入了 JavaScript1.3。对于 Microsoft 公司, 最初是在 Explorer 3 中引入了 JavaScript (称为 JScript, 与 JavaScript1.2 兼容), 后来在 Explorer 4 中升级到 JScript 3.1, 而在 Explorer 5 中则升级到 JScript 5。

虽然 JavaScript 与 JScript 是由不同的公司实现的, 但幸运的是对于大部分的功能实现, 二者几乎没有什么差别。并且, 与其他计算机领域的标准类似, JavaScript 的实现也正在走

向统一, Netscape 和 Microsoft 已将各自的脚本实现提交给了总部位于欧洲的一个标准组织 ECMA (欧洲计算机制造商协会), 由该协会制定统一的标准。ECMA 于 1997 年 6 月发布了 ECMA-262 标准, 并于 1998 年 6 月更新了该标准。ECMA-262 标准规定了 ECMA Script 语言, 该语言结合了 Netscape 的 JavaScript 和 Microsoft 的 JScript 的特点, 具有更强的适用性。JavaScript1.3 支持 ECMA Script 标准, JScript 3.1 也支持该标准。

根据以上介绍, 我们可以得知: JavaScript 和 JScript 并没有本质的差别。因此本书中不区分 JavaScript 与 JScript, 而是一律用 JavaScript 表示。也就是说, 如果没有特殊说明, 所有的内容都是同时适用于 Navigator 和 Explorer 的。虽然如此, 由于在某些使用细节上 Microsoft 与 Netscape 的实现截然不同, 因此有必要进行选择。鉴于我国绝大多数浏览者均使用 Internet Explorer, 本书中凡是涉及二者的不同, 通常都是以 Explorer 为例, 同时忽略 Navigator 的实现。

1.1.3 JavaScript 应用

JavaScript 虽然与 Java 的名称类似, 但二者不论在性质上还是用途上都大不相同: Java 是一种类似于 C++ 的高级语言, 可以实现各种复杂、专业化的应用 (例如, 著名的电子商务平台 Enfinity 就是基于 Java 技术构建的); 而 JavaScript 则是一种脚本语言, 只能实现有限的功能 (例如, 无法进行文件的读写)。

但是作为一种脚本语言, JavaScript 可以说是非常成功的, 它在相当多的领域中得到了广泛应用。它不但可以用于编写 Web 浏览器端 (或者称为客户端) 脚本, 实现在 Web 页面上下文中执行的程序, 而且可以在服务器端用于编写可处理浏览器提交的信息并相应地更新浏览器显示的 Web 服务器程序。

1. 客户端应用

JavaScript 最典型的应用就是开发客户端 Web 应用程序, 也就是开发所谓的客户端脚本。与高级语言不同, 客户端脚本程序通常都是解释执行的。也就是说, 在执行 JavaScript 脚本之前, 无须进行编译等预处理。

在最典型的客户端应用中, JavaScript 脚本程序被嵌入到 HTML 文件中, 随着 HTML 文件一同下载到浏览器端。浏览器读 HTML 文件, 然后解释执行并显示其中的元素。读取 HTML 文件并分辨其中的元素的过程称为语法分析或解析 (parsing)。如果解析到 JavaScript

脚本，则浏览器执行其脚本语句。该过程如图 1.1 所示。

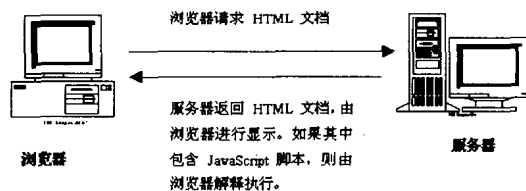


图 1.1 客户端脚本示意图

2. 服务器端应用

JavaScript 脚本不但可以运行于客户端，而且可以运行于服务器端，用于实现服务器端的某些特定功能（例如，取代传统 CGI 程序的表单处理功能）。在 Microsoft 的服务器上，典型的一种应用就是作为 ASP（Active Server Pages，活动服务器页）的实现脚本。

服务器端脚本的工作过程如下：浏览器输入 URL 请求；服务器调用脚本，生成从浏览器传递数据的对象，并向脚本提供这些对象；脚本进行处理（对于数据库应用则需要进行特定的数据库操作），并将数据以 HTML 文件的方式通过服务器返回发出请求的浏览器。该过程如图 1.2 所示。

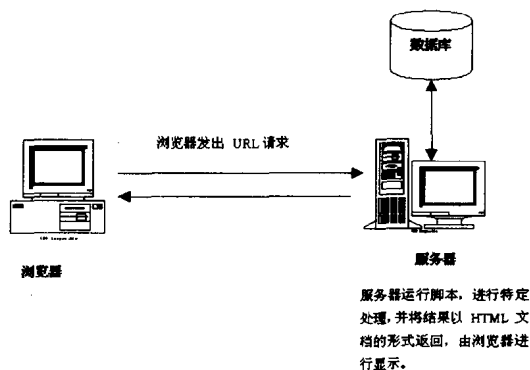


图 1.2 服务器端脚本示意图

说明：本书的重点在于客户端应用，对于如何编写服务器端脚本程序的详细信息，请读者参考其他书籍。

1.1.4 为什么要使用 JavaScript

虽然前面已经介绍了各种 JavaScript 的特点和用途，但读者可能有疑问——为什么我

一定要学习 JavaScript，难道不能有其他替代方式？对于有经验的网页设计者来说还会有一个疑问，既然现在 FrontPage、Dreamweaver 等网页设计软件已经可以自动生成 JavaScript 代码以便完成动态 Web 页的制作，那么还有必要学习 JavaScript 吗？

这两个问题实际上代表了大多数网页设计者的疑问，在此做以下说明：

首先，在本章一开始就已经讲到，JavaScript 是一种最流行的 Web 脚本语言，其主要原因在于它具有最广泛的兼容性。实际上，VBScript 也是一种通用的 Web 脚本语言，而且是由软件业的老大——微软开发的，但由于它只能被 IE 浏览器所支持，因此无法与 JavaScript 抗衡。至于 VBScript 以外的其他 Web 脚本语言就更是影响微弱了。试想，连一向颐指气使的微软都忙不迭地开发出自己的 JavaScript 版本——JScript，那么对于 JavaScript 作为一种 Web 脚本语言的统治地位还能有什么疑义呢？因此，如果需要学习一种 Web 脚本语言的话，JavaScript 无疑是首选。

其次，对于是否需要学习一种 Web 脚本语言的回答显然也是一个响亮的“是”。我们先来看一下 Web 页制作工具软件的变迁：早期最流行的 Web 页制作工具无疑是 FrontPage，但近年来最火的软件却变成了 Dreamweaver，而且大有将 FrontPage 逐出市场的意思。对此，最主要的原因有两条：一是 Dreamweaver 提供的面板式界面比 FrontPage 提供的对话框式界面更加适合 Web 页面设计；二是 Dreamweaver 提供了更多的动态特性，尤其是 Dreamweaver 可以与 Fireworks、Flash 等软件紧密集成，从而开发出动感十足、亮丽眩目的 Web 页面。实际上，第二个原因代表了当今 Web 页开发的一种趋势，即不但要注重内容，而且要注重表现形式。因此开发动态 Web 页已经成为网页制作的一个基本要求。如果我们看一下 Dreamweaver 等软件生成的动态 Web 页的 HTML 源代码，会发现其中充满了大量 JavaScript 代码，可见 Dreamweaver 中相当多的动态特性是借助于 JavaScript 实现的。由于 Web 吸引人之处正在于其独特性，而网页制作工具能够提供的自动功能显然是非常有限的（虽然足以应付基本的需要），因此如果要创建出个性化的动态功能，当然必须使用一种 Web 脚本语言。可见，如果要成为一个 Web 页制作高手，掌握一门 Web 脚本语言是基本的要求。

综上所述，学习 JavaScript 势在必行，它将带领我们进入更广阔的 Web 开发世界。

1.2 在 Web 页中使用 JavaScript

实现客户端 JavaScript 应用时，脚本程序既可以直接嵌入到 HTML 文件中，也可以用

链接的方式将脚本程序和 HTML 文件结合起来。

1.2.1 嵌入 JavaScript

无论使用什么脚本语言，最常用的一种方式是在网页中使用 **SCRIPT** 标记符将脚本语句包含起来，方法是：把脚本标记符 **<SCRIPT>** **</SCRIPT>** 置于网页上的 **HEAD** 部分或 **BODY** 部分。尽管可以在网页上的多个位置使用 **SCRIPT** 标记符，但最好还是将脚本代码放在 **HEAD** 部分，以确保容易维护。

1. 指定脚本语言

在 HTML 4.0 版以前，**SCRIPT** 标记符的 **language** 属性曾被用来规定标记符内的脚本语言是 JavaScript 还是其他类型（如 VBScript、TCL）。在 HTML 4.0 中，W3C 建议使用 **type** 属性替代 **language** 属性。

在实际使用时，可以同时包含 **language** 和 **type** 这两种属性，以适应不同的浏览器。由于 Web 浏览器会忽略其不懂的标记符和属性，因此当包括了 **language** 和 **type** 两个属性时，Web 浏览器会使用其懂得的属性并忽略其不懂的属性。

因此，当使用 JavaScript 编写脚本时，基本语法如下：

```
<SCRIPT Language = "JavaScript" TYPE= "text/javascript" >  
    JavaScript 代码  
</SCRIPT>
```

网页设计者也可以在 **HEAD** 标记符内放置 **META** 标记符，将 **META** 标记符的 **HTTP-EQUIV** 属性设置为 **"Content-Script-Type"**，将其 **CONTENT** 属性设置为 **"text/javascript"** 或其他（如 **"text/vbscript"**），则可以设置整个网页的默认脚本语言。

例如，以下语句设置整个网页的默认脚本语言是 JavaScript：

```
<META http-equiv= "Content-Script-Type" content= "text/javascript" >
```

2. 其他 language 属性

如果将 **language** 属性设置为 JavaScript，则所有支持 JavaScript 的浏览器都能处理 JavaScript 代码。但 **language** 属性也可以设置为下列其他数值，以便限制能处理 JavaScript 代码的浏览器：

- JavaScript1.1 只有支持 JavaScript1.1 的浏览器才能处理 JavaScript 代码，包括