



张立峰
飞思数码产品研发中心

编著
监制



JavaScript

动态网页 技术详解

本书特色：

涵盖JavaScript动态网页开发技术的方方面面，是一本从入门到精通的宝典

本书作者为国内软件开发界的资深软件工程师，在Java、.NET、JavaScript和Web编程方面颇有心得

76个大型实例，221个小型实例，在学习的同时，丰富知识点的理解

150种提示技巧，贯穿于各个章节知识点间

光盘特色：

本书实例的源代码清单

JavaScript特效源代码集合

1364个精彩网页欣赏

21套电子商务网页模板

203种配色方案和配色辞典



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

CD-ROM



随书所附光盘包含书中实例源文件



张立峰
飞思数码产品研发中心

编著
监制

JavaScript

动态网页 技术详解

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本 JavaScript 综合开发技术大全,几乎囊括了使用 JavaScript 进行网页设计开发的全部知识,同时在讲解的过程中结合了大量实用而有代表性的实例。本书全部采用知识点与实例相结合的讲解方式,对每个实例的关键知识点都做了细致的注释,让读者通过阅读本书,可以快速掌握用 JavaScript 制作动态网页的技巧,在短时间内独立建立起有着友好用户体验的网页。附赠光盘中包含有本书实例的源代码清单、特效源代码集合,以及其他网页设计和制作的相关资源。

本书适合动态网页开发的初学者阅读,也可以作为大、中专院校相关专业学生的培训教材和参考资料。

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,侵权必究。

图书在版编目(CIP)数据

JavaScript 动态网页技术详解/张立峰编著. —北京:电子工业出版社, 2009.10

(精通)

ISBN 978-7-121-08676-2

I. J... II. 张... III. ①JAVA 语言—程序设计②主页制作 IV. TP312 TP393.092

中国版本图书馆 CIP 数据核字 (2009) 第 060255 号

责任编辑:王树伟 田志虹

印 刷:北京天宇星印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:27.5 字数:704 千字

印 次:2009 年 10 月第 1 次印刷

印 数:4 000 册 定价:55.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

JavaScript 是一种简洁的、基于对象的、跨平台的描述语言,专门用来开发 Web 客户端和服务端的应用程序。使用 JavaScript 可以快速创建类似 CGI (通用网关接口) 的程序,实现动态网页的交互功能,因而它可以说是网络时代最重要的语言之一。

时至今日,JavaScript 在互联网领域的应用已经非常普遍,大部分的网页都或多或少地含有 JavaScript 代码。近几年 Web 2.0 的兴起,也受到了 AJAX 这种 JavaScript 新应用的推动。无论是程序员、网页设计人员,还是其他非专业的网页编程爱好者,都会接触和应用到 JavaScript。本书主要面向 JavaScript 初学者,将指导读者从最基本的语法开始,逐步迈入 JavaScript 的应用殿堂。

本书内容

全书共分为 4 个部分 21 章:

第 1 章至第 3 章主要讲解了 JavaScript 的基础知识,包括 JavaScript 的概述、HTML 语言与 JavaScript 的关系,以及 JavaScript 的开发环境等内容。

第 4 章至第 12 章讲解了 JavaScript 的核心内容,包括 JavaScript 的语言结构、数据类型、字符串、函数、对象、正则表达式、事件处理、级联样式表和数据存储等内容。

第 13 章至第 18 章主要讲解了 JavaScript 的高级进阶功能,包括 JavaScript 与 XML 的应用、AJAX 应用、框架的应用、脚本编辑的相关技术、JavaScript 应用技巧和 JavaScript 的安全性。通过该部分的学习,读者能够轻松掌握 JavaScript 流行特性的实现。

第 19 章至第 21 章主要讲解了多种 JavaScript 流行应用实例的实现方法,包括源码分析、服务器端编程和网上购物系统。通过该部分的学习,可以提高读者的学习兴趣,加强对知识点的理解,并能够与实际工作相结合。

本书特点

本书通过大量的实例代码,由浅入深、循序渐进地介绍了 JavaScript 的知识要点和编程设计技巧。本书全部采用知识点与实例相结合的讲解方式,理论结合实际,对每个实例的关键知识点都做了细致的注释。读者阅读本书后,可以快速掌握用 JavaScript 制作动态网页的基本知识,使读者能够在短时间内掌握相关的方法和技巧,能够快速独立地建立具有友好用户体验的网页。

读者对象

由于 JavaScript 代码是嵌入在 HTML 代码中执行的脚本语言,因此要求读者具有一定的 HTML 基础。本书适合学习动态网页制作的初学者,也可以作为网络技术培训教材和网络软件开发人员的参考资料,还可以作为大、中专院校相关专业学生的自学教材。

本书附赠光盘包含本书绝大部分实例的源代码清单、特效源代码集合,以及其他网页设计和制作的相关资源。

本书由张立峰执笔,另外张晓景、李晓斌、王权、王大远、张航、罗廷兰、张智英、王明、王延楠、于海波、肖阁、刘钊、魏华、孟权国、张国勇、贾勇、梁革、邹志连、贺春香等也参与了部分编写工作。由于时间仓促,书中难免存在错误和疏漏之处,希望广大读者朋友批评、指正。

编 著 者

光盘说明



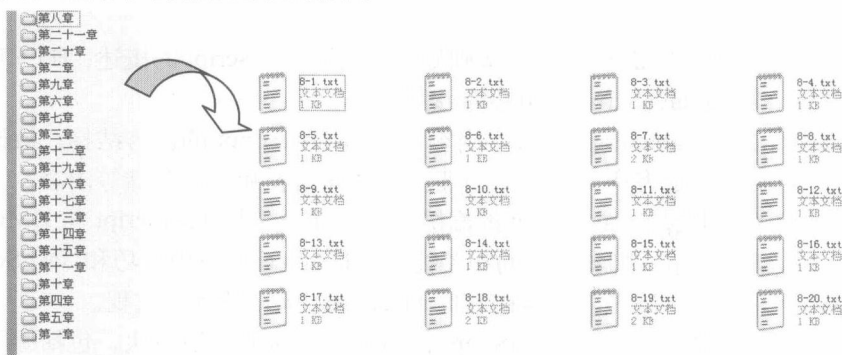
本书附赠光盘包含本书绝大部分实例的源代码清单、特效源代码集合，以及其他网页设计和制作的相关资源。

1. 光盘使用方法

将本书的配套光盘放入光驱中，会自动运行多媒体程序，进入光盘的主界面。如果光盘没有自动运行，可以在“我的电脑”中双击 CD 光盘的盘符，进入配套光盘即可。

2. 本书实例的源代码清单

在光盘的主界面单击左侧的“浏览光盘”，即可进入下一级界面，在其中双击“本书实例源代码”文件夹，即可查看本书实例的源代码清单。



3. 特效源代码集合及网页设计和制作的相关资源

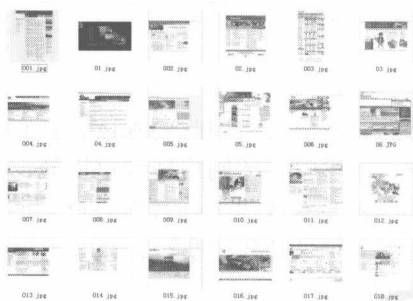
JavaScript 特效源代码集合——“特效源码”文件夹

1364 个精彩网页欣赏——“1364 个网页欣赏”文件夹

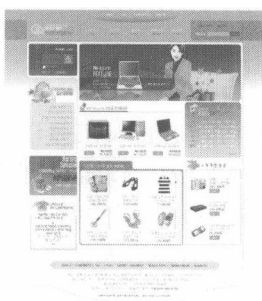
21 套电子商务网页模板——“电子商务网页模板”文件夹

203 种配色方案和配色辞典——“配色相关”文件夹

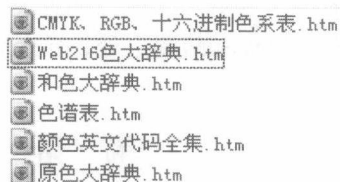
网页欣赏：



电子商务模板：



配色辞典：



第一部分 概 述

第 1 章 JavaScript 编程起步——开始

JavaScript 之旅 3

1.1 什么是 JavaScript 4

● 本书实例源代码\第一章\1-1.txt 和 1-2.txt

1.1.1 了解 JavaScript 4

1.1.2 JavaScript 是脚本编写语言 7

1.1.3 JavaScript 是基于
对象的语言 81.1.4 JavaScript 是解释
执行的语言 9

1.1.5 JavaScript 是弱类型语言 10

1.1.6 JavaScript 通过事件
驱动执行 10

1.2 为何选择 JavaScript 11

1.2.1 B/S 模式 11

1.2.2 JavaScript 的应用范围 12

1.2.3 对 JavaScript 的一些误解 15

1.3 JavaScript 的支持环境 16

● 本书实例源代码\第一章\1-3.txt

1.3.1 JavaScript 浏览器及服务器 16

1.3.2 JavaScript 的应用环境 17

1.3.3 JavaScript 的版本 22

1.4 JavaScript 和 Java 的

联系和区别 23

1.4.1 JavaScript 和 Java 的联系 23

1.4.2 JavaScript 和 Java 的区别 23

1.5 JavaScript 是不断发展的 25

1.6 JavaScript 和 VBScript 的
比较 26

● 本书实例源代码\第一章\1-4.txt

1.6.1 关于 VBScript 26

1.6.2 VBScript 的功能特点 27

1.7 本章小结 30

第 2 章 HTML 语言——学习

JavaScript 的基础 31

2.1 HTML 的基础知识 31

● 本书实例源代码\第二章\2-1.txt~2-4.txt

2.1.1 HTML 的相关概念 32

2.1.2 HTML 的基本架构 36

2.1.3 HTML 的公共属性 37

2.2 在 HTML 文档中嵌入

JavaScript 45

● 本书实例源代码\第二章\2-5.txt~2-7.txt

2.2.1 引入 JavaScript

脚本代码 45

2.2.2 嵌入 JavaScript 脚本

代码的位置 47

2.2.3 处理不支持 JavaScript

脚本的情况 49

2.2.4 JavaScript 的实现基础 50

2.3 编写 JavaScript 脚本 50

● 本书实例源代码\第二章\2-8.txt

2.3.1 编写脚本需要的工具 51

2.3.2 编写第一个脚本 51

2.4 脚本在什么时候执行 53

● 本书实例源代码\第二章\2-9.txt 和 2-10.txt

2.4.1 在打开页面时执行脚本 53

2.4.2 利用 onLoad 事件
执行脚本 54

2.4.3 通过用户事件执行脚本 55

2.5 本章小结 56

第 3 章 开发环境 57

3.1 浏览器的支持 57

3.1.1 主流浏览器 57

3.1.2 非主流浏览器 58

3.2 集成开发环境 IDE 59

3.2.1 集成开发环境的
功能和优点 593.2.2 JavaScript 的集成
开发环境 60

3.3 浏览器捕获异常 61

● 本书实例源代码\第三章\3-1.txt

3.3.1 错误与异常的关系 61

3.3.2 不同浏览器环境中的
错误报告 62

3.3.3 异常处理 62

3.4 脚本代码调试 64

3.4.1 调试工具 64

3.4.2 调试技术 65

3.5 本章小结 68

第二部分 JavaScript 核心内容

第4章 语言结构 71

4.1 JavaScript 语言的

基本组成结构 71

4.1.1 JavaScript 语言的

基本架构 71

4.1.2 JavaScript 的基本语法 72

4.2 常量 75

● 本书实例源代码\第四章\4-1.txt~4-3.txt

4.3 变量 77

● 本书实例源代码\第四章\4-4.txt 和 4-5.txt

4.3.1 变量的命名 77

4.3.2 变量的类型 77

4.3.3 变量的声明 78

4.3.4 变量的作用域 79

4.4 表达式 79

● 本书实例源代码\第四章\4-6.txt

4.5 运算符 80

● 本书实例源代码\第四章\4-7.txt~4-9.txt

4.5.1 算术运算符 80

4.5.2 关系运算符 82

4.5.3 逻辑运算符 84

4.5.4 位运算符 85

4.5.5 赋值运算符 86

4.5.6 其他运算符 86

4.5.7 运算符的优先顺序和
结合顺序 88

4.6 程序的流程控制 89

● 本书实例源代码\第四章\4-10.txt~4-21.txt

4.6.1 条件语句 90

4.6.2 循环语句 93

4.6.3 switch 语句 99

4.6.4 异常处理语句 100

4.6.5 其他语句 101

4.7 JavaScript 弹出的对话框 105

● 本书实例源代码\第四章\4-22.txt~4-25.txt

4.7.1 警告 (alert) 对话框 106

4.7.2 确认 (confirm) 对话框 106

4.7.3 提示 (prompt) 对话框 107

4.8 本章小结 108

第5章 数据类型 109

5.1 基本数据类型 109

5.1.1 数值 110

5.1.2 字符串 111

5.1.3 布尔型 113

5.2 复合数据类型 114

● 本书实例源代码\第五章\5-1.txt~5-4.txt

5.2.1 数组 114

5.2.2 对象 118

5.2.3 JavaScript 函数 120

5.3 其他数据类型 121

5.3.1 null 122

5.3.2 undefined 122

5.3.3 Date 对象 123

5.3.4 Error 对象 123

5.4 正则表达式 123

5.4.1 正则表达式常量 123

5.4.2 正则表达式对象 124

5.5 类型识别与类型转换 125

5.6 实例 125

● 本书实例源代码\第五章\5-5.txt

5.7 本章小结 126

第6章 字符串 127

6.1 字符串的构造 127

6.1.1 创建字符串 127

6.1.2 转义序列 127

6.1.3 字符串构造函数 128

6.2 字符串的使用 129

6.3 字符串的处理 134

● 本书实例源代码\第六章\6-1.txt

6.4 本章小结 139

第7章 函数——和对象

交织在一起 141

7.1 函数的定义和调用 141

● 本书实例源代码\第七章\7-1.txt~7-3.txt

7.1.1 定义函数 141

7.1.2 调用函数	144	8.7 实例	175
7.2 全局变量和局部变量	146	● 本书实例源代码\第八章\8-7.txt	
● 本书实例源代码\第七章\7-4.txt 和 7-5.txt		8.8 浏览器中的 BOM 和 DOM ...	177
7.3 函数的参数	148	8.8.1 BOM	177
● 本书实例源代码\第七章\7-6.txt		8.8.2 DOM	178
7.4 函数的属性和方法	149	8.9 window 对象——最基本的	
7.5 参数个数可变的函数	151	浏览器对象	182
● 本书实例源代码\第七章\7-7.txt		● 本书实例源代码\第八章\8-8.txt~8-10.txt	
7.6 创建动态函数	152	8.9.1 window 对象的属性	182
● 本书实例源代码\第七章\7-8.txt		8.9.2 window 对象的方法	183
7.7 JavaScript 中的系统函数	152	8.9.3 创建和关闭窗口	185
7.8 实例	153	8.10 document 对象——窗口	
● 本书实例源代码\第七章\7-9.txt		文档内容的代表	186
7.9 本章小结	155	● 本书实例源代码\第八章\8-11.txt~8-16.txt	
第 8 章 对象——JavaScript 具有		8.10.1 document 对象的属性 ...	189
面向对象的特性	157	8.10.2 document 对象的方法 ...	191
8.1 面向对象编程的简介	157	8.11 其他对象	192
8.1.1 什么是面向对象	158	● 本书实例源代码\第八章\8-17.txt	
8.1.2 面向对象的设计原则	159	8.11.1 navigator 对象	192
8.2 什么是对象	160	8.11.2 location 对象	193
8.3 对象的基础知识	161	8.11.3 history 对象	194
8.3.1 对象的基本结构	162	8.11.4 screen 对象	195
8.3.2 对象的分类	162	8.12 JavaScript 常用	
8.3.3 对象的特点	163	对象的实例	196
8.3.4 对象的继承	163	● 本书实例源代码\第八章\8-18.txt	
8.4 引用对象的途径	164	8.12.1 用户验证的实例	196
8.4.1 对象属性 Properties 的		8.12.2 使用单选按钮和	
引用	165	复选框的实例	197
8.4.2 对象方法 Method 的引用 ...	166	8.12.3 JavaScript 字符串对象 ...	199
8.4.3 删除对象的属性	167	8.13 本章小结	200
8.5 JavaScript 的内置对象	167	第 9 章 正则表达式	201
● 本书实例源代码\第八章\8-1.txt~8-4.txt		9.1 什么是正则表达式	201
8.5.1 Object 对象	168	9.1.1 正则表达式的概念	201
8.5.2 String 对象	168	9.1.2 正则表达式的定义方法 ...	202
8.5.3 Math 对象 (静态对象) ...	170	9.1.3 正则表达式的作用	203
8.5.4 Date 对象	171	9.2 正则表达式的语法参考	203
8.5.5 toString 方法	172	● 本书实例源代码\第九章\9-1.txt	
8.6 有关对象的操作语句	173	9.2.1 限定符	204
● 本书实例源代码\第八章\8-5.txt 和 8-6.txt		9.2.2 选择匹配符	204
		9.2.3 分组组合符和反向	
		引用符	205

9.2.4 特殊字符	205	● 本书实例源代码\第十一章\11-1.txt~	
9.2.5 字符匹配符	206	11-4.txt	
9.2.6 定位符	207	11.1.1 CSS 的语法规则	236
9.2.7 原义字符	208	11.1.2 样式表的几种设置方式	237
9.2.8 操作符的运算优先级	209	11.1.3 样式表中的注释	239
9.3 RegExp 对象	211	11.1.4 样式规则的继承	240
9.3.1 创建 RegExp		11.1.5 样式规则的优先级	240
对象实例的方法	211	11.2 样式表的属性	241
9.3.2 RegExp 对象的属性	212	● 本书实例源代码\第十一章\11-5.txt~	
9.3.3 RegExp 对象的方法	213	11-10.txt	
9.4 正则表达式的应用实例	213	11.2.1 字体	241
● 本书实例源代码\第九章\9-2.txt~9-7.txt		11.2.2 背景	243
9.4.1 string 对象中与正则		11.2.3 文本	244
表达式有关的方法	214	11.2.4 列表	245
9.4.2 检验用户邮件地址格式	215	11.2.5 位置	246
9.4.3 校验 IP 地址是否合法	216	11.2.6 div 和 span	247
9.4.4 从 URL 中提取各组		11.2.7 链接	248
元素信息	217	11.2.8 盒子	249
9.4.5 常用的正则表达式	218	11.2.9 CSS 滤镜	251
9.5 本章小结	219	11.3 样式规则的选择器	253
第 10 章 事件处理	221	● 本书实例源代码\第十一章\11-11.txt 和	
10.1 事件处理的基本概念	221	11-12.txt	
10.1.1 事件和事件处理的概念	222	11.4 本章小结	258
10.1.2 事件与处理代码关联	223	第 12 章 数据存储	259
10.2 JavaScript 处理事件的方法	225	12.1 cookie	259
10.2.1 事件的分类	225	12.1.1 介绍 cookie	259
10.2.2 设置对象事件的方法	226	12.1.2 cookie 的属性	261
10.3 实例	229	12.1.3 创建与读取 cookie	263
● 本书实例源代码\第十章\10-1.txt		12.1.4 cookie 中信息的	
10.3.1 自动装载和卸载的示例	229	存放方法	265
10.3.2 给事件定时的示例	229	12.2 userData	265
10.3.3 定时循环的示例	230	12.3 实例	267
10.3.4 识别用户浏览器的示例	232	● 本书实例源代码\第十二章\12-1.txt	
10.4 本章小结	234	12.3.1 源代码	267
第 11 章 级联样式表	235	12.3.2 功能概述	272
11.1 什么是级联样式表	235	12.4 本章小结	272

第三部分 JavaScript 的高级进阶功能

第 13 章 XML——可扩展		13.1 XML 概述	275
标记语言	275	● 本书实例源代码\第十三章\13-1.txt	
		13.1.1 XML 简介	275

13.1.2 XML 的文档规则	277	15.1.2 框架开发的技术	311
13.1.3 XML 文档的编辑方法	280	15.2 Prototype 框架	312
13.1.4 一个 XML 文档的实例	281	● 本书实例源代码\第十五章\15-1.txt~	
13.1.5 XML 的功能	282	15-3.txt	
13.2 XML 如何表示信息	284	15.2.1 什么是 Prototype	312
● 本书实例源代码\第十三章\13-2.txt 和		15.2.2 Prototype 的常用函数	312
13-3.txt		15.3 JQuery 框架	316
13.2.1 XML 的基本组件	284	● 本书实例源代码\第十五章\15-4.txt 和	
13.2.2 XML 的模式	285	15-5.txt	
13.2.3 常见的 XML 信息 建模错误	287	15.4 框架窗口 (Frame)	319
13.3 XML DOM 对象	288	● 本书实例源代码\第十五章\15-6.txt	
● 本书实例源代码\第十三章\13-4.txt 和		15.5 本章小结	323
13-5.txt		第 16 章 脚本编程的相关技术	325
13.3.1 XML DOM 的结点	288	16.1 用 Microsoft Visual Studio.NET	
13.3.2 XMLHTTP 对象	289	编写 JavaScript 程序	325
13.4 本章小结	290	16.1.1 查看 Microsoft Visual Studio	
第 14 章 AJAX 简介——异步		.NET 文档	325
JavaScript 和 XML	291	16.1.2 使用 Microsoft Visual Studio	
14.1 认识 AJAX	291	.NET 编写脚本程序	325
14.1.1 什么是 AJAX	291	16.2 编写兼容 IE 和 NN 的	
14.1.2 AJAX 的组成技术	292	JavaScript 代码	326
14.1.3 实现 AJAX 的步骤	293	● 本书实例源代码\第十六章\16-1.txt	
14.1.4 AJAX 的应用	295	16.3 插件技术	329
14.2 AJAX 的核心——		16.3.1 Netscape 插件	330
XMLHttpRequest	296	16.3.2 Java Applet	330
● 本书实例源代码\第十四章\14-1.txt~		16.4 本章小结	332
14-4.txt		第 17 章 JavaScript 的常用技巧	333
14.2.1 实例分析	296	17.1 开启新窗口	333
14.2.2 XMLHttpRequest		● 本书实例源代码\第十七章\17-1.txt 和	
对象的属性	298	17-2.txt	
14.2.3 XMLHttpRequest		17.2 输入验证	334
对象的方法	298	● 本书实例源代码\第十七章\17-3.txt~	
14.2.4 处理 AJAX 响应		17-4.txt	
——innerHTML	299	17.3 页面跳转	337
14.3 Prototype 对 AJAX 的支持	302	● 本书实例源代码\第十七章\17-5.txt	
● 本书实例源代码\第十四章\14-5.txt~		17.4 数据选取	337
14-9.txt		● 本书实例源代码\第十七章\17-6.txt	
14.4 本章小结	308	17.5 密码保护页	338
第 15 章 常见框架的介绍	309		
15.1 应用框架	309		
15.1.1 应用框架的优点	310		

● 本书实例源代码\第十七章\17-7.txt	
17.6 定时消失的弹出框	339
● 本书实例源代码\第十七章\17-8.txt	
17.7 定时执行函数	340
● 本书实例源代码\第十七章\17-9.txt	
17.8 其他技巧	341
● 本书实例源代码\第十七章\17-10.txt	
17.9 本章小结	346

第 18 章 JavaScript 的网络安全性...	347
18.1 威胁类别	347
18.1.1 STRIDE 威胁和对策	347
18.1.2 网络威胁和对策	348
18.1.3 主机威胁和对策	350
18.2 Internet 安全性威胁	351
18.3 安全措施	353
18.4 JavaScript 的安全性	355
18.5 本章小结	357

第四部分 JavaScript 应用实例

第 19 章 JavaScript 的常用特效 ... 361

19.1 鼠标特效	361
● 本书实例源代码\第十九章\19-1.txt~ 19-3.txt	
19.2 图片特效	366
● 本书实例源代码\第十九章\19-4.txt 和 19-5.txt	
19.3 菜单特效	369
● 本书实例源代码\第十九章\19-6.txt~ 19-8.txt	
19.4 文字特效	372
● 本书实例源代码\第十九章\19-9.txt~ 19-12.txt	
19.5 页面特效	376
● 本书实例源代码\第十九章\19-13.txt	
19.6 其他特效	378
● 本书实例源代码\第十九章\19-14.txt~ 19-20.txt	
19.7 本章小结	383

第 20 章 服务器端编程 385

20.1 利用 JavaScript 建立 Web 应用	385
● 本书实例源代码\第二十章\20-1.txt	

20.2 JavaScript 服务器端编程.....387

● 本书实例源代码\第二十章\20-2.txt	
20.2.1 CGI 和 ASP 的 相关知识	387
20.2.2 一个简单的 ASP 脚本页面	393
20.3 实例一页面交互程序	394
● 本书实例源代码\第二十章\20-3.txt 和 20-4.txt	
20.3.1 功能概述	397
20.3.2 基本的脚本结构	398
20.3.3 客户端 JavaScript 脚本 ...	400
20.3.4 服务器端脚本详解	401
20.4 本章小结	403

第 21 章 综合案例——网上

购物系统 405

21.1 本案例的技术要点	405
● 本书实例源代码\第二十一章\21-1.txt	
21.2 源代码清单	406
21.3 功能概述	412
21.4 程序详解	415
21.4.1 banner 框架	415
21.4.2 cont 框架	416
21.4.3 cont 框架中的脚本详解...	417
21.4.4 main 框架	421
21.4.5 main 框架脚本详解	423
21.5 本章小结	428

JavaScript

第一部分 概 述



第 1 章 JavaScript 编程起步——开始 JavaScript 之旅

无论您以前学没学过 JavaScript，本教程都能带您进入 JavaScript 的殿堂，体验 JavaScript 的魅力。下面就开始我们的 JavaScript 之旅。

因特网是 Internet 的中文译名，它的前身是由美国国防部高级研究计划局（ARPA）主持研制的 ARPAnet。随着社会科技、文化和经济的发展，特别是计算机网络技术和通信技术的发展，今天的 Internet 变成了一个开发和利用信息资源的覆盖全球的信息海洋。

Internet 时代造就了新的工作方式和生活方式，其互联性、开放性和共享信息资源的模式，打破了传统信息传播方式的重重壁垒，为我们带来了新的机遇和挑战。随着信息时代的来临，人类社会前进的脚步在逐渐加快，每一天都会有新的事情发生，每一天都在创造着新的奇迹。各行各业都在加入 Internet 的队伍中，无论从哪个角度，管理方面，还是从商业角度来看，Internet 都能带来无限生机。通过 Internet，可以实现各个地域的连接，从而达到一种“和谐的统一”。

怎样把自己的或公司的信息资源加入到 WWW 服务器中实现资源共享，是我们关心的问题。实现这个目标最简单、最快速的方法就是采用超链接技术（超文本和超媒体技术），通过超文本（Hyper Text）和超媒体（Hyper Media）技术结合超链接（Hyper Link）的链接功能将各种信息组织成网络结构（Web），构成网络文档（Document），实现 Internet 上的“漫游”。而这种手段的支持环境，就是 HTML 语言。HTML（超文本标识语言）的描述就可以实现文字、声音、图像、表格、动画等多媒体信息的检索。通过它就可以制作出我们所需的 Web 网页。

然而采用这种超链接技术也存在一定的缺陷：它只能提供一种静态的信息资源，缺少客户端与服务器端的动态交互。虽然通过 CGI（Common Gateway Interface）通用网关接口可实现一定程度的交互，但该方法编程较为复杂，因而在一段时间内妨碍了 Internet 技术的发展。然而 JavaScript 的出现，无疑为 Internet 用户带来了无限生机。可以说，JavaScript 的出现是时代的需求，是当今的信息时代造就了 JavaScript。

JavaScript 的出现，打破了原有的用户和信息之间单一的浏览和显示的关系，实现了一种动态的、实时的、可交互式的表达方式。使基于 CGI 的静态的 HTML 页面可以提供实时动态的信息。JavaScript 脚本正是为了满足这种需求而产生的语言。它深受编程爱好者的欢迎和喜爱，是众多脚本语言中较为优秀的一种，因此，尽快掌握 JavaScript 脚本语言的编写方法是用户最关心的问题。

本章学习要求	学习目标	了解	理解	应用
	知识点			
	什么是 JavaScript		✓	
	JavaScript 的特点		✓	
	JavaScript 的应用范围	✓		
	JavaScript 和 Java 的区别		✓	

1.1 什么是 JavaScript

光盘\本书实例源代码\第一章\1-1.txt 和 1-2.txt

JavaScript 最早是由网景 (Netscape) 公司开发出来的一种跨平台的、面向对象的脚本语言。最初这种脚本语言只能在网景公司的浏览器——Netscape 中使用, 目前几乎所有的主流浏览器都支持 JavaScript。

JavaScript 是对 ECMA 262 语言规范的一种实现, 是一种基于对象 (Object) 和事件驱动 (Event Driven) 并具有安全性能的脚本语言。它与 HTML 超文本标记语言、Java 脚本语言 (Java 小程序) 一起实现在一个 Web 页面中链接多个对象, 与 Web 客户进行交互的作用, 从而可以开发客户端的应用程序等。它是通过嵌入到标准的 HTML 语言中实现的, 弥补了 HTML 语言的缺陷。



提示

其实 JavaScript 最杰出的地方在于可以用很小的程序做大量的工作, 不需要有高配置、高性能的计算机, 软件也仅需要一个字处理软件及浏览器。

JavaScript

JavaScript 是一种新的描述语言, 它可以被嵌入到 HTML 的文件之中。JavaScript 语言可以做到响应请求事件 (如: 表格的输入), 而不用任何网络来回传输资料。所以当使用者输入数据时, 它不用经过传给服务器端 (server) 处理, 再传回来的过程, 可以直接在客户端 (client) 进行处理。

1.1.1 了解 JavaScript

早期在 Web 页面中进行的所有操作都必须传回服务器, 然后由服务器进行集中处理, 处理完毕后, 再将处理的结果通过网络传回到客户端的浏览器中供用户查看使用。即使是最简单的验证用户输入的数据是否有效的过程, 比如通过判断输入字符串是否包含 “@” 符号来判断用户输入的 E-mail 地址是否有效, 也必须由服务器来完成。在这种模式下, 一旦 Web 访问量增加, 服务器的负担会加重。这一时期的客户/服务器结构并不是真正意义上的客户/服务器结构。人们期待一种新的技术来实现真正意义上的客户/服务器结构, 即在客户端也可以进行交互处理, 从而减轻服务器的负担, 加快网络的传输速度。JavaScript 正是在这种背景下产生的。

JavaScript 是一种新的 Web 技术, 是 Netscape 开发出的一种脚本语言, 最初的名字是

LiveScript, 是为了扩展 HTML 的功能, 用于替代复杂的 CGI 程序来处理 Web 页表单信息, 为 Web 页增加动态效果的。当 Java 出现以后, Netscape 和 Sun 公司一起开发了一种新的脚本语言, 它的语法和 Java 非常类似, 最后被命名为 JavaScript。

自诞生以来, JavaScript 已经取得了广泛的支持, 支持者包括 IBM、Oracle、Apple、Borland、Sybase、Informix 等。这不仅仅表现在浏览器中得到了越来越多的支持, 也包括在其他的各种应用程序中得到了应用。在新的 Windows 操作系统中, 也可以使用脚本来制订需要完成的任务。

JavaScript 语言也是由一种编程语言演变而来的。一种编程语言通常是由另一种编程语言演变而来的, 例如, Java 是从 C++演变来的, 而 C++是从 C 演变来的。Netscape 最初开发了 LiveScript 语言, 在 Navigator 和 Web 服务器中加入了基本的脚本功能, 当 Navigator 2.0 中加进了 Java 小程序支持后, Netscape 把 LiveScript 变成了 JavaScript。最初的 JavaScript 不过是 LiveScript 的更名, 但随后每次新的 Navigator 版本都使 JavaScript 的功能有所改进和完善。



提示

Java 的小应用程序也可以实现客户端的处理, 但这对众多的网页编写者来说不是件容易的事。因为使用 Java 需要网页编程者有较高的编程能力, 编写者愿意使用更简单的方法来实现表单或其他逻辑的处理。所以 Java 作为一种强类型的程序设计语言, 并不是制作 Web 页面的最佳选择。

JavaScript

Microsoft 作为软件界的领路者, 意识到了 Web 脚本的重要性, 自然不甘心在 Web 脚本的竞争中落后。但是由于得不到 Netscape 在技术上的许可, Microsoft 开发了一个自己的脚本语言——JScript, 并在其自己的浏览器 Microsoft Internet Explorer 3.0 及更高版本中对其提供了强有力的支持。由于 Microsoft 在软件市场中的优势, JScript 很快得到了广泛的支持和应用。早期的 JScript 1.0 只是很粗糙地和 JavaScript 1.1 兼容, Netscape 在其浏览器 Navigator 3.0 及其以后的版本中也对 JScript 提供了支持。随着 JavaScript 版本的增多和浏览器平台的不同, 让众多的网页编写者感到难以取舍, 也增加了额外的工作量。

鉴于脚本语言开发商之间的竞争给网页编写者带来的麻烦, Microsoft、Netscape 和其他脚本语言商决定成立一个国际组织, 并将其命名为 ECMA, 专门从事脚本语言标准的制订。ECMA 制订的脚本语言标准被称为 ECMAScript, 所有开发商的脚本语言都支持这一标准。尽管有 ECMA 标准的存在, Netscape 和 Microsoft 都有其各自的脚本语言——JavaScript 和 JScript, 这两种语言都对 ECMA 标准进行了扩展。

虽然有其他语言的竞争, JavaScript 还是成为了标准的 Web 脚本语言。大部分人认为 JavaScript 只是用来编写客户端的 Web 应用程序, 其实 JavaScript 还可以用来编写服务器端的应用程序。Netscape 为用户提供了服务器端的脚本语言 Netscape Server Side JavaScript (SSJS), 可以在服务器端编写需要的 Web 应用程序, 不过使用 SSJS 需要 Netscape Server 3 的支持。Microsoft 也有自己的服务器端脚本编程语言 Active Server Pages (ASP) ——需要 JScript 引擎的支持。Microsoft 除了开发了 JScript 脚本之外, 还有 VBScript, 它也是一种脚本语言。VBScript 实际上是 Visual Basic 程序设计语言的一个子集, 它使得 VB 程序员很容

易编写自己的 Web 应用程序。Netscape 并没有对 VBScript 提供支持, 所以使用 VBScript 的还仅是 Microsoft Internet Explorer 用户。



提示

JavaScript

JavaScript 支持 Web 浏览器和服务器脚本。浏览器脚本用于生成插件和 Java 小应用程序相关联的动态 Web 页面。JavaScript 支持这些特性, 定义专用事件处理函数和提供动态产生 HTML 等特殊编程功能。

在服务器方面, JavaScript 用于更方便地开发处理表单数据、进行数据库搜索和实现专用 Web 应用程序的脚本。服务器脚本比 CGI 程序更紧密地联系在 Web 服务器上。开发人员必须用 Netscape 的 LiveWire 工具库开发服务器端脚本。

JavaScript 是被嵌入到 HTML 中的, 最大的特点便是和 HTML 的结合。当 HTML 文档在浏览器中被打开时, JavaScript 代码才被执行。JavaScript 代码使用 HTML 标记 `<script>` `</script>` 嵌入到 HTML 文档中。JavaScript 扩展了标准的 HTML, 为 HTML 标记增加了事件, 通过事件驱动来执行 JavaScript 代码。在服务器端, JavaScript 代码可以作为单独的文件存在, 但也必须通过在 HTML 文档中调用才能起作用。下面程序中的例子说明了 JavaScript 代码是如何嵌入到 HTML 文档中的。

```
<html>
<head>
<title>在 HTML 文档中嵌入 JavaScript 代码</title>
<script language="JavaScript">
<!--
window.defaultStatus="使用 HTML 标记嵌入 JavaScript 代码"
function rest( )
{
document.form1.text1.value="嵌入 JavaScript 代码"
}
//-->
</script>
</head>
<body>
<center>
<h1>JavaScript 示例</h1>
<hr>
<form name="form1">
<input type="text" name="text1" size=40 value="输入信息">
<br><br>
<input type="button" value="查看信息" onclick="rest( )">
</form>
</center>
</body>
</html>
```

本例仅说明如何在 HTML 文档中嵌入 JavaScript 代码, 在后面的章节中将详细介绍 JavaScript 的使用方法。图 1-1 显示了本例在 IE 7.0 浏览器中打开的实际效果。