

精品升级

畅销图书优化升级，更加符合学习需求

读者定位

对网站建设和网页设计有热爱兴趣的初学者

范例丰富

以大量实战范例深入解析常用知识点

@ 网站建设与网页设计案例指导

胡孟杰 郑延斌 岳明 等编著



实例源文件及相关
素材文件

网页设计师的修炼秘笈
个性网页设计的成功阶梯

JavaScript 动态网页开发 案例指导



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

@ 网站建设与网页设计案例指导

JavaScript 动态网页开发 案例指导

胡孟杰 郑延斌 岳 明 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

JavaScript 是 Sun Microsystems 和 Netscape 共同开发的一种非常重要的脚本语言，可以创建具有动态效果的人机交互功能的 Web 浏览页面。对于 HTML 开发来说，JavaScript 有助于构建用户交互的 HTML 系统。

本书以技术讲解+实例演示的方式，选取 JavaScript 在 HTML 页面应用的典型实例，循序渐进地介绍使用 JavaScript 语言开发动态网页的方法和技巧。全书共 19 章，第 1 章到第 17 章分别介绍了 JavaScript 语言的基本概念、特性、编写方法、浏览器环境、HTML 和 CSS 的基本应用、数据类型、运算符、语句函数、对象、事件、窗口、文档、表单、DOM 模型、正则表达式、插件通信、多媒体特效和 Ajax 等技术。第 18 章和第 19 章介绍了网络视频点播系统和客户服务系统两个综合案例，从项目计划需求分析到页面设计和实现并解析编程思想和技术。

本书内容丰富、结构清晰、由浅入深，实例的选择紧贴实际应用，具有很强的实用性和针对性，是一本简单易学而又极具参考价值的实例教程。本书主要面向广大对网页编程有兴趣并且有一定计算机基础的读者。对于使用 JavaScript 语言开发动态页面的工程人员，也是一本不可多得的参考资料。

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 动态网页开发案例指导 / 胡孟杰等编著. —北京：电子工业出版社，2009.4
(网站建设与网页设计案例指导)

ISBN 978-7-121-08609-0

I. J… II. 胡… 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 048629 号

责任编辑：祁玉芹 郑志宁

印 刷：北京市天竺颖华印刷厂

装 订：三河市鑫金马印装有限公司

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：32 字数：819 千字

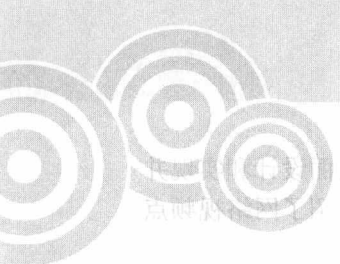
印 次：2009 年 4 月第 1 次印刷

定 价：58.00 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 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。



前言

JavaScript是由 Netscape 公司开发的一种脚本语言 (scripting language), 或者称为描述语言。在 HTML 基础上, 使用 JavaScript 可以开发实现服务器和用户动态数据交换的交互式 Web 网页。Javascript 的出现使得网页和用户之间实现了一种实时性的、动态的、交互性的关系, 使网页包含更多活跃的元素和更加精彩的内容。JavaScript 短小精悍, 又是在客户机上执行的, 大大提高了网页的浏览速度和交互能力。同时它是专门为制作 Web 网页而量身定做的一种简单的编程语言。

本书以实例的形式介绍了 JavaScript 的各种技术和相关应用, 对于 JavaScript 技术, 均先讲解技术要点和参数设定, 再通过具体的范例对技术进行阐述和演示, 并“举一反三”供读者练习。整个的讲解过程条理清晰、循序渐进, 使读者在阅读完一个章节后, 可以全面了解 JavaScript 的基础知识, 并掌握相应的技术应用和编程思想, 可以灵活应用 JavaScript 语言创造属于自己的动态网页的 JavaScript 脚本。

全书共分 19 章, 第 1 章和第 2 章介绍进行 JavaScript 语言开发需要了解的基础知识和概念, 并就相应的知识点选取典型的范例进行讲解分析, 以便读者可以顺利地消化知识。第 1 章主要介绍 JavaScript 语言概述、特性、编写和执行方法等。第 2 章分为两部分, 第一部分介绍 HTML 语言的和基本标记、框架、表单、表格和多媒体应用, 第二部分介绍 CSS 样式表的实现, 以及文本、颜色、背景、表格等样式如何设定。第 3 章到第 12 章为 JavaScript 语言的基础部分, 详细介绍了构成 JavaScript 语言的各个组成部分的基础知识和应用方法。第 3 章介绍了 JavaScript 中的数据类型和变量, 包括基础数据、复合数据、特殊数据, 以及变量的声明和使用。第 4 章介绍了 JavaScript 的各种运算符的定义、优先级以及结合性。第 5 章介绍 JavaScript 的语句, 包括表达式语句、复合语句、标记语句和空语句。第 6 章介绍了 JavaScript 的特殊内部函数和用户自定义函数, 以及自定义函数的定义和调用。第 7 章介绍了 JavaScript 的内部对象和生成对象, 以及生成对象的类型、构造函数、属性和方法。第 8 章介绍了 JavaScript 的数组和方法, 以及数组对象的各种方法和属性。第 9 章介绍 document 对象的属性、方法、子对象的应用和 cookie 的使用。第 10 章分别介绍各种事件, 包括浏览器事件、鼠标事件、键盘事件和错误事件, 还介绍了如何使用事件处理器及其绑定。第 11 章重点介绍了表单和各种表单元素的应用, 以及相应的表单事件。第 12 章介绍了窗口和框架的各种使用方法。

第 13 章到第 17 章为 JavaScript 语言的扩展应用, 作为基础知识部分的补充。其中第 13 章介绍了 DOM 模型和相应的对象的属性和方法。第 14 章介绍了正则表达式的建立和规则, 以及字符串对象对正则表达式的支持。第 15 章介绍 JavaScript 的浏览器插件和插件通信安装, 以及 Java Applet 的应用。第 16 章介绍了如何应用 JavaScript 脚本语言实现页面中的文字、图像和其他各种特效。第 17 章重点介绍了 Ajax 的基础知识以及功能实现。

第 18 章和第 19 章解析编程思想和技术, 为广播系统的设计和开发提供思路。

本书由胡孟杰、郑延斌和岳明等编写。本书致力于让多层次的读者阅读后都能有所收获，但是由于编者的水平有限，加之创作时间有限，本书疏漏之处在所难免，欢迎读者与专家批评指正。

评指正。如蒙交稿，请寄：中国图书馆学会图书馆技术委员会（http://www.cnlib.org.cn），邮编：100071。在《信息资源管理》发表后，作者将获稿酬并赠样书。来稿恕不退件。投稿邮箱地址：http://www.cnlib.org.cn。英网：http://www.cnlib.org.cn。地址：北京海淀区中关村大街28-1号B座10层1009室。电话：(010)62529511。E-mail: cnlib@cnlib.org.cn。编著者注：本馆自建有“中国图书馆学会图书馆技术委员会”网站以来，一直为国内外图书馆界提供免费的中文、英文、日文、韩文、俄文、德文、法文、西班牙文、葡萄牙文、意大利文、希腊文、土耳其文、阿拉伯文、希伯来文、印地安文、泰文、越南文、老挝文、柬埔寨文、缅甸文、尼泊尔文、孟加拉文、斯里兰卡文、印度尼西亚文、马来西亚文、菲律宾文、泰国文、新加坡文、香港文、澳门文、台湾文、大陆文、海外华人、华侨、留学生、学者、专家、教授、研究员、博士生、硕士生、本科生、专科生、中专生、高中生、初中生、小学生、幼儿、儿童、少年、青年、中年、老年、残疾人、贫困人口、下岗职工、失业人员、农民工、进城务工人员、流动人口、留守儿童、流动儿童、流动青少年、流动青年、流动青年女性、流动青年男性、流动青年儿童、流动青年父母、流动青年祖父母、流动青年外祖母、流动青年外祖父、流动青年舅舅、流动青年舅母、流动青年姑姑、流动青年姑父、流动青年叔叔、流动青年婶婶、流动青年伯伯、流动青年姨妈、流动青年姨夫、流动青年外公、流动青年外婆、流动青年爷爷奶奶、流动青年姥爷、流动青年姥姥、流动青年爸爸妈妈、流动青年兄弟姐妹、流动青年亲戚朋友、流动青年同事同学、流动青年邻居街坊、流动青年社区成员、流动青年社会组织成员、流动青年志愿者、流动青年义工、流动青年公益人士、流动青年爱心人士、流动青年慈善家、流动青年企业家、流动青年成功人士、流动青年精英阶层、流动青年社会名流、流动青年公众人物、流动青年意见领袖、流动青年网络红人、流动青年网红、流动青年大V、流动青年KOL、流动青年意见领袖、流动青年网络红人、流动青年网红、流动青年大V、流动青年KOL。

2009年2月

若蒙惠顾，请向单商函。



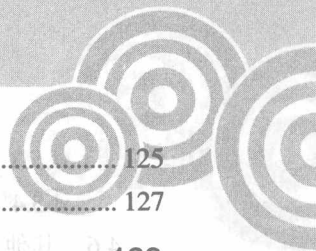
目 录

第 1 章	JavaScript 概述	1
1.1	什么是 JavaScript	1
1.2	JavaScript 特性	2
1.3	如何编写 JavaScript 程序	3
1.3.1	语句	3
1.3.2	注释	3
1.3.3	表达式	3
1.4	JavaScript 使用	4
第 2 章	HTML 与 CSS 概述	5
2.1	HTML 简介	5
2.2	HTML 标记	7
2.2.1	基本标记	7
2.2.2	文本格式化	10
2.2.3	字符	13
2.2.4	链接	15
2.3	框架	18
2.4	表格	20
2.5	表单	23
2.6	图像与多媒体	26
2.7	CSS 层叠样式表	27
2.7.1	CSS 基本语法	28
2.7.2	css 的类别	29
2.7.3	CSS 的使用	33
第 3 章	数据类型与变量	35
3.1	基本数据类型	35
3.1.1	数值	35
3.1.2	字符串	37
3.1.3	布尔	38

3.2 复合数据类型.....	40
3.2.1 数组.....	40
3.2.2 对象.....	41
3.3 特殊数据类型.....	43
3.3.1 Null.....	43
3.3.2 Undefined.....	44
3.4 变量.....	45
3.4.1 变量声明.....	45
3.4.2 变量范围.....	45
3.4.3 变量命名.....	47
3.4.4 强制转换.....	47
第4章 运算符.....	51
4.1 赋值运算符.....	51
4.1.1 赋值.....	51
4.1.2 运算赋值.....	52
4.2 算术运算符.....	54
4.2.1 负值.....	54
4.2.2 四则运算.....	55
4.2.3 递增与递减.....	57
4.2.4 取模运算.....	59
4.3 逻辑运算符.....	61
4.3.1 逻辑非.....	61
4.3.2 逻辑与.....	62
4.3.3 逻辑或.....	64
4.4 关系运算符.....	65
4.4.1 大于.....	66
4.4.2 小于.....	67
4.4.3 等于、大于等于、小于等于.....	69
4.4.4 不等于.....	70
4.4.5 条件（三元运算符）.....	71
4.4.6 逗号.....	73
4.4.7 严格相等.....	74
4.4.8 非严格相等.....	76
4.5 位运算符.....	77
4.5.1 按位取反.....	77
4.5.2 按位左移与按位右移.....	79



4.5.3	无符号右移.....	80
4.5.4	按位逻辑运算.....	81
4.6	其他运算符.....	83
4.6.1	typeof 运算符.....	83
4.6.2	void 运算符.....	85
4.7	运算符的优先级与结合性.....	86
第5章	语句.....	87
5.1	复合语句.....	87
5.1.1	if 语句.....	87
5.1.2	if...else if...else...语句.....	89
5.1.3	switch 语句.....	91
5.1.4	while 语句.....	94
5.1.5	do...while 循环.....	96
5.1.6	for 语句.....	97
5.1.7	for...in 语句.....	99
5.2	标记语句.....	100
5.2.1	break 语句.....	101
5.2.2	continue 语句.....	103
5.2.3	var 语句.....	104
5.2.4	function 语句.....	105
5.2.5	return 语句.....	106
5.2.6	throw 语句.....	106
5.2.7	try...catch...finally 语句.....	107
5.2.8	with 语句.....	109
5.3	空语句.....	111
第6章	函数.....	113
6.1	用户创建函数.....	113
6.1.1	定义和调用.....	113
6.1.2	参数.....	116
6.1.3	作用域.....	118
6.2	特殊内部函数.....	118
6.2.1	escape()函数和 unescape()函数.....	118
6.2.2	encodeURIComponent()函数和 decodeURI()函数.....	120
6.2.3	encodeURIComponent()函数和 decodeURIComponent()函数.....	123
6.2.4	parseInt()函数和 parseFloat()函数.....	123



6.2.5	isNaN()函数和 isFinite()函数	125
6.2.6	eval()函数	127
第 7 章	对象	129
7.1	String 对象	129
7.1.1	字符串长度	129
7.1.2	大小写转换	131
7.1.3	字符串定位	132
7.1.4	连接与分割字符串	134
7.1.5	使用子字符串	137
7.2	内部对象	139
7.2.1	Math 对象	139
7.2.2	Date 对象	143
7.2.3	Number 对象	149
7.2.4	Boolean 对象	151
7.2.5	Object 对象	152
7.2.6	Error 对象	153
7.2.7	Function 对象	156
7.2.8	Global 对象	158
7.2.9	RegExp 对象	158
7.3	生成对象	162
7.3.1	对象类型	162
7.3.2	构造函数	163
7.3.3	对象属性和方法	165
第 8 章	数组	169
8.1	数组和数组元素	169
8.2	数组方法	172
8.2.1	数组下标	172
8.2.2	对象为关联数组	174
8.3	Array 对象	175
8.3.1	多维数组	175
8.3.2	数组的长度	177
8.3.3	合并数组	179
8.3.4	添加与删除数组元素	181
8.3.5	反转数组	183
8.3.6	数组排序	184

第9章 Document 文档对象 187

9.1 Document 对象	187
9.1.1 属性	187
9.1.2 方法	190
9.2 使用 cookie	192
9.3 Document 子对象	197
9.3.1 Document 子对象概述	197
9.3.2 Link 对象和 Anchor 对象	198
9.3.3 Image 图像对象	202

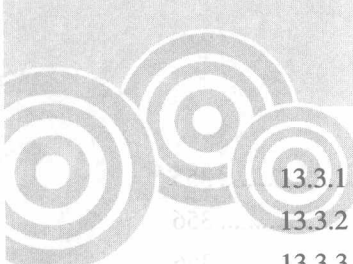
第10章 事件与事件处理 207

10.1 事件和事件类型	207
10.2 设置事件处理器	207
10.3 事件处理器的绑定	209
10.3.1 使用 HTML 标签属性绑定事件处理器	209
10.3.2 使用 JavaScript 对象属性绑定事件处理器	211
10.4 浏览器事件	213
10.4.1 Load 事件	213
10.4.2 Unload 事件	214
10.4.3 Submit 事件	216
10.5 鼠标事件	218
10.5.1 MouseDown 事件	218
10.5.2 MouseUp 事件	219
10.5.3 MouseMove 事件	220
10.5.4 MouseOver 事件	222
10.5.5 MouseOut 事件	224
10.5.6 Click 事件	225
10.5.7 Focus 事件和 Blur 事件	228
10.6 键盘事件	231
10.6.1 KeyDown 事件、KeyUp 事件和 KeyPress 事件	231
10.6.2 Change 事件	233
10.7 错误事件	234

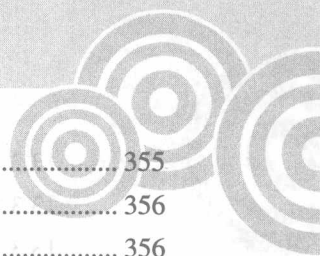
第11章 表单对象与表单元素 239

11.1 Form 对象	239
11.1.1 Form 对象属性	239

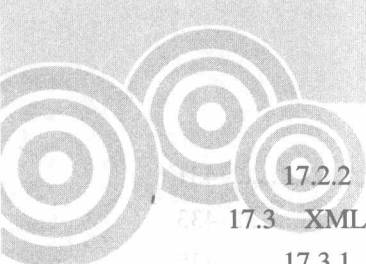
11.1.2	访问表单.....	242
11.1.3	访问表单元素.....	244
11.2	表单事件.....	247
11.2.1	Form 对象的事件.....	247
11.2.2	表单元素的事件.....	249
11.3	表单元素.....	251
11.3.1	文本输入对象.....	251
11.3.2	按钮对象.....	255
11.3.3	单选按钮.....	258
11.3.4	复选框.....	260
11.3.5	选择列表.....	262
11.3.6	文件域.....	265
第 12 章 窗口和框架		271
12.1	使用窗口.....	271
12.1.1	打开和关闭窗口.....	271
12.1.2	移动窗口位置.....	274
12.1.3	改变窗口大小.....	275
12.1.4	设置状态栏.....	276
12.1.5	超时功能.....	278
12.1.6	使用对话框.....	279
12.1.7	窗口事件.....	282
12.2	使用框架.....	285
12.2.1	frames 对象.....	285
12.2.2	框架命名.....	288
12.2.3	访问窗口.....	289
12.2.4	引用变量和函数.....	289
12.2.5	框架间的互相引用.....	290
12.2.6	框架嵌套.....	291
12.2.7	框架交叉通信.....	291
第 13 章 DOM 对象模型		295
13.1	DOM 概述	295
13.2	Document 节点	296
13.2.1	document 的属性.....	296
13.2.2	document 节点的方法.....	297
13.3	Node 节点	299



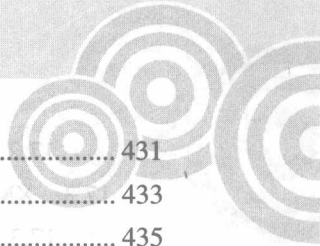
13.3.1	Node 节点的属性	299
13.3.2	Node 节点的方法	301
13.3.3	获取节点的方法	303
13.3.4	节点之间的关系	305
13.4	window 对象	305
13.4.1	window 对象的方法	305
13.4.2	window 对象的事件	308
13.5	frame 对象	310
13.6	document 对象	311
13.6.1	document 对象的属性	312
13.6.2	document 对象的方法	314
13.6.3	document 对象的对象属性	315
13.6.4	cookie 属性	318
13.7	history 对象	319
13.7.1	history 对象的属性	319
13.7.2	history 对象的方法	319
13.8	location 对象	321
13.8.1	location 对象的属性	321
13.8.2	location 对象的方法	322
13.9	navigator 对象	322
13.9.1	navigator 对象的方法	322
13.9.2	navigator 对象的属性	322
第 14 章 正则表达式		325
14.1	正则表达式	325
14.1.1	正则表达式概述	325
14.1.2	建立正则表达式	327
14.2	RegExp 对象和 String 对象对正则表达式的支持	331
14.2.1	RegExp 对象	331
14.2.2	String 对正则表达式的支持	336
14.3	正则表达式的语法规则	341
14.3.1	普通字符、字符匹配符和非打印字符	342
14.3.2	操作符及其优先权顺序	345
14.3.3	转义字符与特殊字符	350
第 15 章 JavaScript 插件通信		355
15.1	浏览器插件	355



15.1.1	浏览器插件简介.....	355
15.1.2	Eclipse 下 JavaScript 插件 JSEclipse.....	356
15.1.3	检测已安装的插件.....	356
15.1.4	对浏览器插件的检测.....	357
15.2	JavaScript 与 Java Applet 通信.....	359
15.2.1	Java Applet 概述.....	360
15.2.2	JavaScript 脚本与 Java Applet 互访.....	362
15.2.3	Java Applet 安全措施.....	364
15.2.4	Java Applet 与浏览器间的通信.....	365
15.3	插件的安装.....	367
15.3.1	Eclipse 插件的安装与配置.....	367
15.3.2	Netscape7.0 的安装.....	367
15.3.3	Active 控件.....	368
第 16 章 JavaScript 特效.....		371
16.1	文字特效.....	371
16.1.1	自动变换文字颜色.....	371
16.1.2	文字拖动特效.....	372
16.1.3	打字机式字符输出.....	375
16.2	图像特效.....	377
16.2.1	图像闪烁.....	377
16.2.2	图像变形.....	379
16.2.3	图像翻转.....	381
16.2.4	图像淡出.....	384
16.3	其他特效.....	386
16.3.1	飞雪特效.....	387
16.3.2	跟随鼠标的文字特效.....	389
16.3.3	跑马灯式文字说明框.....	392
第 17 章 Ajax 技术基础.....		395
17.1	Ajax 简介.....	395
17.1.1	Ajax 概述.....	395
17.1.2	Ajax 的工作原理.....	396
17.1.3	基于 Ajax 的架构.....	396
17.1.4	Ajax 的服务器端.....	397
17.2	Ajax 特征解析.....	397
17.2.1	Ajax 的数据封装.....	397



17.2.2	Web 设计.....	398
17.3	XML HttpRequest 对象.....	398
17.3.1	创建 XML HttpRequest 对象.....	398
17.3.2	使用方法.....	400
17.3.3	设置属性.....	401
17.4	远程脚本.....	404
17.4.1	远程脚本概述.....	404
17.4.2	远程脚本的使用.....	405
17.5	Ajax 开发工具.....	407
17.5.1	使用 JSLint 对 JavaScript 进行语法检查.....	408
17.5.2	使用 JSDoc 建立 JavaScript 代码的文档.....	408
17.5.3	用 prototype 属性建立面向对象的 JavaScript.....	408
第 18 章 网络视频点播系统.....		411
18.1	项目计划.....	411
18.2	需求分析.....	412
18.3	系统建模.....	412
18.3.1	系统流程分析.....	412
18.3.2	网站结构设计.....	413
18.3.3	数据库结构设计.....	413
18.4	界面设计.....	413
18.4.1	网络视频点播系统首页设计.....	413
18.4.2	用户登录页面设计.....	414
18.4.3	用户注册页面设计.....	415
18.4.4	用户信息页面设计.....	415
18.4.5	视频列表页面设计.....	416
18.4.6	视频播放页面设计.....	416
18.4.7	视频下载页面设计.....	417
18.4.8	管理员登录页面设计.....	417
18.4.9	视频信息管理页面设计.....	418
18.4.10	视频信息添加页面设计.....	418
18.4.11	用户信息管理页面设计.....	419
18.5	代码实现.....	419
18.5.1	网络视频点播系统首页代码实现.....	419
18.5.2	用户登录页面代码实现.....	422
18.5.3	用户注册页面代码实现.....	426
18.5.4	用户信息页面代码实现.....	429



18.5.5	视频列表页面代码实现.....	431
18.5.6	视频播放页面代码实现.....	433
18.5.7	视频下载页面代码实现.....	435
18.5.8	管理员登录页面代码实现.....	439
18.5.9	视频信息管理页面代码实现.....	441
18.5.10	视频信息添加页面代码实现.....	443
18.5.11	用户信息管理页面代码实现.....	446
第 19 章	企业客户服务系统	449
19.1	项目计划.....	449
19.2	需求分析.....	450
19.3	系统建模.....	451
19.3.1	系统功能模块.....	451
19.3.2	系统流程分析.....	451
19.3.3	网站结构设计.....	451
19.4	界面设计.....	452
19.4.1	企业客户服务系统首页设计.....	452
19.4.2	用户登录页面设计.....	453
19.4.3	客户信息管理页面设计.....	453
19.4.4	产品列表页面设计.....	454
19.4.5	客户问题提交页面设计.....	454
19.4.6	客户服务信息管理页面设计.....	454
19.4.7	产品管理页面设计.....	455
19.4.8	问题管理页面设计.....	457
19.4.9	系统管理员信息页面设计.....	459
19.4.10	用户管理页面设计.....	459
19.5	代码实现.....	460
19.5.1	企业客户服务系统首页实现.....	460
19.5.2	用户登录页面实现.....	464
19.5.3	客户信息管理页面实现.....	468
19.5.4	产品列表页面实现.....	471
19.5.5	客户问题提交页面实现.....	475
19.5.6	客户服务信息管理页面实现.....	478
19.5.7	产品管理页面实现.....	481
19.5.8	问题管理页面实现.....	485
19.5.9	系统管理员信息页面实现.....	489
19.5.10	用户管理页面实现.....	492

第1章 JavaScript 概述

Internet 的迅猛发展，特别是 WWW 标准的普及很大程度得益于网络上的超媒体信息技术。为了满足 Web 用户对信息资源的需求，开发者不断地扩充超媒体语言的功能。从静态文本到静态图像，从静态图像到动态图像，又加入了音频、视频、Flash 动画等。同时，用户不满足于只是被动浏览，他们希望服务方提供更多的交互性。JavaScript 语言以及另外一些脚本语言（Script）的出现满足了服务方和用户的这种要求。它们为 Web 内容开发者提供了一种灵活简捷的 Web 页面设计途径。这其中，JavaScript 在众多脚本语言中脱颖而出，被越来越多的 Web 作者认可和使用。

1.1 什么是 JavaScript

JavaScript 是一种基于对象的脚本语言，可以开发 Internet 客户端的应用程序。JavaScript 在 HTML 页面中以语句形式出现，并且可以执行相应的操作。

虽然 JavaScript 与 Java 有紧密的联系，但却是两个公司开发的不同的两个产品。Java 是 SUN 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发；而 JavaScript 是 Netscape 公司的产品，其目的是为了扩展 Netscape Navigator 功能，而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言，它的前身是 Live Script；Java 的前身是 Oak 语言。下面对两种语言间的异同作如下比较。

基于对象和面向对象：Java 是一种真正的面向对象的语言，即使是开发简单的程序，必须设计对象。JavaScript 是一种脚本语言，它可以用来制作与网络无关的，与用户交互作用的复杂软件。它是一种基于对象（Object Based）和事件驱动（Event Driver）的编程语言。因而它本身提供了非常丰富的内部对象供设计人员使用。

- 解释和编译：两种语言在其浏览器中所执行的方式不一样。Java 的源代码在传递到客户端执行之前，必须经过编译，因而客户端上必须具有相应平台上的仿真器或解释器，它可以通过编译器或解释器实现独立于某个特定的平台编译代码的束缚。JavaScript 是一种解释性编程语言，其源代码在发往客户端执行之前不需经过编译，而是将文本格式的字符代码发送给客户端由浏览器解释执行。
- 强变量和弱变量：这两种语言所采取的变量是不一样的。Java 采用强类型变量检查，即所有变量在编译之前必须作声明。JavaScript 中变量声明，采用其弱类型，即变量在使用前不需作声明，而是解释器在运行时检查其数据类型。
- 代码格式不一样：Java 是一种与 HTML 无关的格式，必须通过像 HTML 中引用外媒体那么进行装载，其代码以字节代码的形式保存在独立的文档中。JavaScript 的代码是一种文本字符格式，可以直接嵌入 HTML 文档中，并且可动态装载。编写 HTML 文档就像编辑文本文件一样方便。

- 嵌入方式不一样：在 HTML 文档中，两种编程语言的标识不同，JavaScript 使用 `<Script>...</Script>` 来标识，而 Java 使用 `<applet>...</applet>` 来标识。
- 静态联编和动态联编：Java 采用静态联编，即 Java 的对象引用必须在编译时进行，以使编译器能够实现强类型检查。JavaScript 采用动态联编，即 JavaScript 的对象引用在运行时进行检查，如不经编译，则就无法实现对象引用的检查。

1.2 JavaScript 特性

目前，除 JavaScript 以外，还有许多用于 Web 设计的工具，它们在一些功能上甚至超过 JavaScript（例如 QuickTimeVR、ShockWave、RealAudio、TrueSpeech）。然而，重要的是，为了使用这些工具，你需要购买相应的开发环境，而且这些工具往往并不便宜。而 JavaScript 的功能则是与浏览器相联的，无需任何附加的软件。JavaScript 具有以下几个基本特点。

- 脚本编写语言：JavaScript 是一种脚本语言，它采用小程序段的方式实现编程。像其他脚本语言一样，JavaScript 同样也是一种解释性语言，它提供了一个容易的开发过程。它的基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言一样，需要先编译，而是在程序运行过程中被逐行地解释。它与 HTML 标识结合在一起，从而方便用户的使用操作。
- 基于对象的语言：JavaScript 是一种基于对象的语言，同时可以看作是一种面向对象的。这意味着它能运用自己已经创建的对象。因此，许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。
- 简单性：JavaScript 的简单性主要体现在以下几个方面，首先它是一种基于 Java 基本语句和控制流之上的简单而紧凑的设计，从而对于学习 Java 是一种非常好的过渡。其次它的变量类型是采用弱类型，并未使用严格的数据类型。
- 安全性：JavaScript 是一种安全性语言，它不允许访问本地的硬盘，并不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互。从而有效地防止数据的丢失。
- 动态性：JavaScript 是动态的，它可以直接对用户或客户输入做出响应，无须经过 Web 服务程序。它对用户的反映响应，是采用以事件驱动的方式进行的。所谓事件驱动，就是指在主页（Home Page）中执行了某种操作所产生的动作，就称为“事件”（Event）。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后，可能会引起相应的事件响应。
- 跨平台性：JavaScript 是依赖于浏览器本身，与操作环境无关，只要能运行浏览器的计算机，并支持 JavaScript 的浏览器就可正确执行。从而实现了“编写一次，走遍天下”的梦想。

JavaScript 为 Web 内容设计人员提供了极大的灵活性和控制手段。它是一种优秀的“粘合剂”，能够将一个 Web 节点中的不同组成部分捆绑在一起，成为一个紧密结合的信息源。然而，像任何事物一样，JavaScript 也有它的局限性。

- 平台相关的局限性：尽管 JavaScript 可以运行在任何安装有 Netscape Navigator 的平台上（Windows, Macintosh 和 Unix），但并不是 JavaScript 的所有部分在这些平台上都按相同的状况运行。