

## <<JavaScript宝典>>

### 图书基本信息

书名：<<JavaScript宝典>>

13位ISBN编号：9787302303220

10位ISBN编号：7302303223

出版时间：2013-1

出版时间：清华大学出版社

作者：(美)古德曼(Goodman, D.) 等著

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<JavaScript宝典>>

### 内容概要

您想创建更具魅力的交互网页吗？

您想更好地控制浏览器呈现内容的方式吗？

经典畅销书 JavaScript宝典(第7版) 将帮您实现这些夙愿，它在大量示例代码和可运行脚本的引导下，指导您快速掌握JavaScript基础知识，并制定出最符合预定网站目标的策略。

通过认真研读本书，您将可以编写脚本来实现翻转等效果，并熟练使用Web 2.0和JavaScript子例程库等功能。

### 主要内容

- 探讨最新的JavaScript编程技术

- 制定脚本编写策略并选择合适的工具

- 深入了解终止器、生成器和迭代器

- 应用最新的JavaScript异常处理和自定义对象技术

- 利用DOM的强大功能

- 使用Ajax、E4X/XML和JSON推动Web 2.0应用

- 执行数据输入验证和提高安全性

### 配书光盘

配书光盘的精彩内容如下：

29个附赠章节，这些章节讨论了数据输入验证、脚本调试和跨浏览器动态HTML等高级主题。

可供立即使用的300多个脚本。

本书的示例代码。

## 作者简介

Danny Goodman 撰写了许多颇受欢迎的畅销书，包括The Complete HyperCard Handbook、Danny Goodman ' s AppleScript Handbook、 Dynamic HTML: The Definitive Reference 和JavaScript & DHTML Cookbook。

他是声誉卓著的计算机脚本语言权威人士和专家级教师。

他的写作风格和教育方式会继续为他赢得全球各地读者和教师的赞美。

Michael Morrison是一位作家、开发人员、玩具发明家，还是多部Java、C++、Web脚本、XML、游戏开发和移动设备等图书的作者，Michael 撰写的一些著名图书有Faster Smarter HTML and XML、 Teach Yourself HTML & CSS in 24 Hours 和Beginning Game Programming。

Michael 还是Stalefish 实验室([www.stalefishlabs.com](http://www.stalefishlabs.com)) 的创始人，这是一家专门开发非常游戏、玩具和互动产品的娱乐公司。

Paul Novitski 自从1981年就开始作为一位自由职业的程序员编写软件。

他曾经自学BASIC，来编写机器语言反汇编器，所以他可以砍掉一些Wang ' s OIS 微码。

他自从90年代后期开始专注于互联网编程。

他的公司Juniper Webcraft 开发的HTML-strict 网站便于访问，使用语义标记、分隔的开发层和直观的用户界面。

在生活中，他享受着甜美的安比拉琴音乐和抚养双胞胎儿子的快乐。

Tia Gustaff Rayl 是一位数据库和Web 技术的开发和培训顾问。

最近她发布了XHTML、CSS、JavaScript 和SQL的课件。

她在获得佛罗里达大学的英语和教育博士学位后开始了其软件生涯。

与大多数这个领域的新手一样，她最初的计算任务是维护软件。

她在软件业呆的时间很长，完全了解了软件的整个生命周期、应用程序和数据库开发、项目管理、PC 和大型机环境的培训。

90年代中期，她开始开发早期的支持Web 的数据库应用程序，并在其代码中添加JavaScript。

她继续承接开发项目，以磨练自己的编程技巧。

她梦想着可以利用业余时间与丈夫及两条狗一起周游世界。

## &lt;&lt;JavaScript宝典&gt;&gt;

## 书籍目录

## 目 录

## 第I部分 JavaScript入门

## 第1章 JavaScript在万维网和其他领域所起的作用 3

## 1.1 Web流量的竞争 4

## 1.2 其他Web技术 4

## 1.2.1 超文本标记语言(HTML和XHTML) 5

## 1.2.2 CSS 7

## 1.2.3 服务器编程 7

## 1.2.4 辅助程序和插件程序 8

## 1.3 JavaScript是一门综合性语言 9

## 1.3.1 LiveScript蜕变成JavaScript 10

## 1.3.2 微软的JavaScript版本 10

## 1.3.3 JavaScript版本 10

## 1.3.4 核心语言标准ECMAScript 11

## 1.4 JavaScript: 灵活易用的工具 12

## 第2章 脚本开发策略 13

## 2.1 浏览器的竞争 13

## 2.2 相互包容 14

## 2.3 当今存在的兼容性问题 14

## 2.3.1 将核心JavaScript语言从文档对象中独立出来 15

## 2.3.2 核心语言标准 15

## 2.3.3 文档对象模型 16

## 2.3.4 通过标记打下良好的基础 17

## 2.3.5 层叠样式表 17

## 2.3.6 标准兼容模式(DOCTYPE转换) 18

## 2.3.7 动态HTML和定位 19

## 2.4 开发脚本编写策略 19

## 2.4.1 功能降低和渐进增强 19

## 2.4.2 开发层的分离 20

## 2.4.3 延伸阅读 21

## 第3章 选择和使用工具 23

## 3.1 软件工具 23

## 3.1.1 选择文本编辑器 23

## 3.1.2 选择浏览器 24

## 3.2 建立编写环境 24

## 3.2.1 Windows 25

## 3.2.2 Mac OS X 25

## 3.2.3 重载问题 26

## 3.3 验证 26

## 3.4 创建第一个脚本 27

## 3.4.1 第一步: 静态HTML 27

## 3.4.2 第二步: 连接JavaScript 28

## &lt;&lt;JavaScript宝典&gt;&gt;

- 3.4.3 第三步：用CSS指定样式 29
- 第4章 JavaScript基础 31
  - 4.1 合并JavaScript和HTML 31
    - 4.1.1 标记 31
    - 4.1.2 旧式内联JavaScript 35
    - 4.1.3 容纳不支持JavaScript的用户代理 35
    - 4.1.4 隐藏脚本 39
    - 4.1.5 给不同的浏览器编写脚本 40
  - 4.2 兼容性设计 44
    - 4.2.1 处理beta版浏览器 44
    - 4.2.2 参考章节中的兼容性等级 45
  - 4.3 资深程序员的语言基础 46
- 第 部分 JavaScript教程
- 第5章 第一个JavaScript脚本 53
  - 5.1 第一个脚本的功能 53
  - 5.2 输入第一个脚本 54
    - 5.2.1 第一步：HTML文档 54
    - 5.2.2 第二步：添加JavaScript 57
    - 5.2.3 第三步：添加样式 63
  - 5.3 进行改动 65
  - 5.4 习题 65
- 第6章 浏览器对象和文档对象 67
  - 6.1 脚本运行初步 67
  - 6.2 使用JavaScript的场合 68
  - 6.3 文档对象模型 69
    - 6.3.1 HTML结构和DOM 69
    - 6.3.2 浏览器窗口中的DOM 70
  - 6.4 文档的载入 71
    - 6.4.1 简单文档 72
    - 6.4.2 添加段落元素 72
    - 6.4.3 添加段落文本 72
    - 6.4.4 生成新元素 73
  - 6.5 对象引用 73
    - 6.5.1 对象命名 74
    - 6.5.2 引用特定对象 74
  - 6.6 节点术语 75
    - 6.6.1 节点 75
    - 6.6.2 父子节点 76
  - 6.7 对象的定义 76
    - 6.7.1 属性 76
    - 6.7.2 方法 77
    - 6.7.3 事件 79
  - 6.8 习题 80
- 第7章 脚本和HTML文档 83
  - 7.1 把脚本连接到文档上 83
    - 7.1.1 script标记的位置 84

## &lt;&lt;JavaScript宝典&gt;&gt;

- 7.1.2 非JavaScript的浏览器和XHTML 85
- 7.2 JavaScript语句 86
- 7.3 脚本语句的执行时间 87
  - 7.3.1 文档载入时即刻执行 87
  - 7.3.2 延时脚本 88
- 7.4 查找脚本错误 90
- 7.5 脚本和编程 91
- 7.6 习题 92
- 第8章 程序设计基础(一) 93
  - 8.1 JavaScript语言 93
  - 8.2 处理信息 93
  - 8.3 变量 94
    - 8.3.1 创建变量 94
    - 8.3.2 变量的命名 95
  - 8.4 表达式和求值 95
    - 8.4.1 脚本中的表达式 96
    - 8.4.2 表达式和变量 97
  - 8.5 数据类型转换 97
    - 8.5.1 将字符串转换成数值 98
    - 8.5.2 将数字转换成字符串 99
  - 8.6 操作符 99
    - 8.6.1 算术操作符 99
    - 8.6.2 比较操作符 100
  - 8.7 习题 100
- 第9章 程序设计基础(二) 103
  - 9.1 决策和循环 103
  - 9.2 控制结构 103
    - 9.2.1 if结构 104
    - 9.2.2 if ... else结构 104
  - 9.3 重复循环 105
  - 9.4 函数 106
    - 9.4.1 函数的参数 107
    - 9.4.2 变量的作用域 108
  - 9.5 大括号 109
  - 9.6 数组 110
    - 9.6.1 创建数组 110
    - 9.6.2 访问数组的数据 111
    - 9.6.3 关联数组 111
    - 9.6.4 数组中的document对象 113
  - 9.7 习题 114
- 第10章 window和document对象 115
  - 10.1 顶层对象 115
  - 10.2 window对象 115
    - 10.2.1 访问窗口的属性和方法 116
    - 10.2.2 创建窗口 117
  - 10.3 window对象的属性和方法 119

## &lt;&lt;JavaScript宝典&gt;&gt;

- 10.3.1 window.alert()方法 119
- 10.3.2 window.confirm()方法 119
- 10.3.3 window.prompt()方法 120
- 10.3.4 load事件 120
- 10.4 location对象 121
- 10.5 navigator对象 122
- 10.6 document对象 122
  - 10.6.1 document.getElementById()方法 123
  - 10.6.2 document.getElementsByTagName()方法 123
  - 10.6.3 document.forms[ ]属性 124
  - 10.6.4 document.images[ ]属性 124
  - 10.6.5 document.createElement()和document.createTextNode()方法 125
  - 10.6.6 document.write()方法 126
- 10.7 习题 130
- 第11章 表单和表单元素 131
  - 11.1 form对象 131
    - 11.1.1 将表单作为对象和容器 133
    - 11.1.2 访问表单属性 134
    - 11.1.3 form.elements[ ]属性 135
  - 11.2 将表单控件作为对象 136
    - 11.2.1 与文本相关的输入对象 136
    - 11.2.2 按钮输入对象 139
    - 11.2.3 复选框输入对象 139
    - 11.2.4 单选输入对象 141
    - 11.2.5 select对象 143
  - 11.3 用this向函数传递元素 146
  - 11.4 提交和预验证表单 149
  - 11.5 习题 152
- 第12章 String、Math和Date对象 155
  - 12.1 核心语言对象 155
  - 12.2 String对象 155
    - 12.2.1 连接字符串 156
    - 12.2.2 字符串方法 157
  - 12.3 Math对象 159
  - 12.4 Date对象 160
  - 12.5 日期计算 161
  - 12.6 习题 163
- 第13章 编写框架和多窗口脚本 165
  - 13.1 框架：父框架和子框架 165
  - 13.2 家庭成员之间的引用 167
    - 13.2.1 父到子的引用 167
    - 13.2.2 子到父的引用 167
    - 13.2.3 子到子的引用 168

## &lt;&lt;JavaScript宝典&gt;&gt;

- 13.3 有关框架脚本编程的提示 168
- 13.4 iframe元素简介 169
- 13.5 突出显示脚注：框架集脚本示例 169
- 13.6 多窗口引用 175
- 13.7 习题 178
- 第14章 图像和动态HTML 181
  - 14.1 image对象 181
    - 14.1.1 可互换的图像 182
    - 14.1.2 图像的预缓存 182
    - 14.1.3 图像变换的创建 184
  - 14.2 无需脚本的图像变换 189
  - 14.3 JavaScript：伪URL 192
  - 14.4 主流的动态HTML技术 193
    - 14.4.1 样式表设置的修改 193
    - 14.4.2 通过W3C DOM节点实现动态内容 193
    - 14.4.3 通过innerHTML属性实现动态内容 194
  - 14.5 习题 194
- 第 部分 JavaScript核心语言参考
- 第15章 String对象 199
  - 15.1 字符串以及数值数据类型 199
    - 15.1.1 简单字符串 199
    - 15.1.2 建立长字符串变量 200
    - 15.1.3 连接字符串字面量和变量 200
    - 15.1.4 特殊的内嵌字符 201
  - 15.2 String对象 202
    - 15.2.1 语法 202
    - 15.2.2 关于String对象 203
    - 15.2.3 属性 204
    - 15.2.4 解析方法 207
  - 15.3 字符串使用函数 231
  - 15.4 URL字符串编码及解码 236
- 第16章 Math、Number和Boolean对象 237
  - 16.1 JavaScript中的数值 237
    - 16.1.1 整数和浮点数 237
    - 16.1.2 十六进制和八进制整数 240
    - 16.1.3 将字符串转换成数值 241
    - 16.1.4 将数值转换成字符串 242
    - 16.1.5 数值不是数值型时 243
  - 16.2 Math对象 243
    - 16.2.1 语法 243
    - 16.2.2 关于Math对象 243
    - 16.2.3 属性 244

## &lt;&lt;JavaScript宝典&gt;&gt;

- 16.2.4 方法 244
- 16.2.5 创建随机数 245
- 16.2.6 Math对象的快捷引用 246
- 16.3 Number对象 246
  - 16.3.1 语法 247
  - 16.3.2 关于Number对象 247
  - 16.3.3 属性 247
  - 16.3.4 方法 248
- 16.4 Boolean对象 250
  - 16.4.1 语法 250
  - 16.4.2 关于Boolean对象 250
- 第17章 Date对象 251
  - 17.1 时区和GMT 251
  - 17.2 Date对象 252
    - 17.2.1 创建date对象 253
    - 17.2.2 内部对象的属性和方法 254
    - 17.2.3 日期方法 254
    - 17.2.4 处理时区 257
    - 17.2.5 字符串日期 257
    - 17.2.6 用于以前浏览器的日期格式 258
    - 17.2.7 更多转换 259
    - 17.2.8 日期和时间运算 260
    - 17.2.9 计算天数 262
    - 17.2.10 早期浏览器中日期的错误和漏洞 266
  - 17.3 在表单中验证日期项 267
- 第18章 Array对象 273
  - 18.1 结构化的数据 273
  - 18.2 创建空数组 274
  - 18.3 填充数组 274
  - 18.4 JavaScript数组创建功能的增强 276
  - 18.5 删除数组项 276
  - 18.6 并行数组 277
  - 18.7 多维数组 281
  - 18.8 模拟Hash表 282
  - 18.9 Array对象的属性和方法 284
    - 18.9.1 Array对象属性 285
    - 18.9.2 Array对象的方法 286
  - 18.10 数组包含 311
  - 18.11 解构赋值 312
  - 18.12 与旧浏览器的兼容性 313
- 第19章 JSON — Native JavaScript Object Notation 315
  - 19.1 JSON的工作原理 315
  - 19.2 收发JSON数据 317

## &lt;&lt;JavaScript宝典&gt;&gt;

- 19.3 JSON 对象 318
- 19.4 安全限制 319
- 第20章 E4X — Native XML Processing 321
- 20.1 XML 321
- 20.2 ECMAScript for XML (E4X) 322
  - 20.2.1 使用XML对象 322
  - 20.2.2 在HTML中嵌入E4X 328
  - 20.2.3 方法 328
- 第21章 控制结构和异常处理 331
  - 21.1 if和if...else判定语句 331
    - 21.1.1 简单判定 331
    - 21.1.2 (condition)表达式 332
    - 21.1.3 复杂判定语句 333
    - 21.1.4 嵌套的if...else语句 334
  - 21.2 条件表达式 336
  - 21.3 switch语句 337
  - 21.4 重复(for)循环 340
    - 21.4.1 使用循环计数器 342
    - 21.4.2 跳出循环 343
    - 21.4.3 使用continue继续循环 344
  - 21.5 while循环 345
  - 21.6 do-while循环 346
  - 21.7 遍历属性(for-in) 346
  - 21.8 with语句 348
  - 21.9 标签语句 349
  - 21.10 异常处理 352
    - 21.10.1 异常及错误 352
    - 21.10.2 异常机制 353
  - 21.11 使用try-catch-finally结构 353
- 现实的异常 356
  - 21.12 抛出异常 356
  - 21.13 error对象 361
    - 21.13.1 语法 361
    - 21.13.2 关于error对象 362
    - 21.13.3 属性 362
    - 21.13.4 方法 363
- 第22章 JavaScript操作符 365
  - 22.1 操作符的类别 365
  - 22.2 比较操作符 366
  - 22.3 不同数据类型的相等比较 367
  - 22.4 结合操作符 369
  - 22.5 赋值操作符 371
  - 22.6 布尔操作符 373
    - 22.6.1 布尔运算 374
    - 22.6.2 使用布尔操作符 375

## &lt;&lt;JavaScript宝典&gt;&gt;

- 22.7 按位操作符 377
- 22.8 对象操作符 377
- 22.9 其他操作符 382
- 22.10 操作符的优先级 384
- 第23章 函数和自定义对象 387
  - 23.1 Function对象 387
    - 23.1.1 语法 387
    - 23.1.2 关于Function对象 388
    - 23.1.3 创建函数 388
    - 23.1.4 嵌套函数 389
    - 23.1.5 函数的参数 390
    - 23.1.6 属性 391
    - 23.1.7 方法 395
  - 23.2 函数应用的注意事项 396
    - 23.2.1 调用函数 396
    - 23.2.2 变量的作用域：全局作用域还是局部作用域 397
    - 23.2.3 参数变量 401
    - 23.2.4 递归函数 402
    - 23.2.5 创建函数库 403
    - 23.2.6 封闭区间 404
  - 23.3 使用面向对象的JavaScript创建自定义对象 406
    - 23.3.1 对象的具体细节 407
    - 23.3.2 OO例子：行星对象 409
    - 23.3.3 进一步的封装 412
    - 23.3.4 创建对象数组 412
    - 23.3.5 利用嵌套对象 414
    - 23.3.6 创建对象的最新方法 415
    - 23.3.7 定义对象属性的提取器和设置器 415
  - 23.4 面向对象的概念 416
    - 23.4.1 增加原型 417
    - 23.4.2 原型继承 418
    - 23.4.3 嵌套对象和原型继承 418
  - 23.5 Object对象 420
    - 23.5.1 语法 420
    - 23.5.2 关于该对象 421
    - 23.5.3 属性 422
    - 23.5.4 方法 423
- 第24章 全局函数和语句 425
  - 24.1 函数 426
  - 24.2 语句 435
  - 24.3 WinIE对象 438
    - 24.3.1 ActiveXObject 438
    - 24.3.2 Dictionary 439
    - 24.3.3 Enumerator 440

## &lt;&lt;JavaScript宝典&gt;&gt;

- 24.3.4 VBArray 441
- 第 部分 文档对象参考
- 第25章 文档对象模型基础 445
  - 25.1 对象模型层次结构 445
    - 25.1.1 作为路径图的层次结构 446
    - 25.1.2 第一个浏览器文档对象  
路径图 446
  - 25.2 产生文档对象的过程 447
  - 25.3 对象的属性 448
  - 25.4 对象的方法 449
  - 25.5 对象事件处理程序 450
  - 25.6 对象模型概述 451
  - 25.7 基本对象模型 452
  - 25.8 附加图像的基本对象模型 452
  - 25.9 仅用于Navigator 4的扩展 453
    - 25.9.1 事件捕获模型 453
    - 25.9.2 层 453
  - 25.10 Internet Explorer 4+扩展 454
    - 25.10.1 HTML元素对象 454
    - 25.10.2 元素包含层次结构 454
    - 25.10.3 层叠样式表 455
    - 25.10.4 事件冒泡 456
  - 25.11 Internet Explorer 5+扩展 456
  - 25.12 W3C DOM 457
    - 25.12.1 DOM层 457
    - 25.12.2 规范中恒定不变的部分 458
    - 25.12.3 W3C DOM不具备的特性 458
    - 25.12.4 新的HTML惯例 459
    - 25.12.5 新DOM概念 459
    - 25.12.6 W3C DOM的静态HTML对象 467
    - 25.12.7 双向事件模型 469
  - 25.13 脚本编程的发展趋势 470
    - 25.13.1 将内容与脚本分离 470
    - 25.13.2 尽量使用W3C DOM 471
    - 25.13.3 处理事件 471
  - 25.14 标准兼容模式(DOCTYPE切换) 472
  - 25.15 小结 473
- 第26章 通用HTML元素对象 475
- 第27章 window对象和frame对象 657
  - 27.1 window对象术语 657
  - 27.2 框架 658
    - 27.2.1 创建框架 658

## &lt;&lt;JavaScript宝典&gt;&gt;

- 27.2.2 框架对象模型 658
- 27.2.3 引用框架 660
- 27.2.4 top和parent 660
- 27.2.5 防止在其他Web站点的框架中显示自己的页面 660
- 27.2.6 确认页面载入框架集 661
- 27.2.7 从有框架转换为无框架 661
- 27.2.8 继承性和封装性 661
- 27.2.9 框架的同步 662
- 27.2.10 空白框架 662
- 27.2.11 查看框架源代码 663
- 27.2.12 框架和frame元素对象 663
- 27.3 window对象属性 663
  - 27.3.1 语法 665
  - 27.3.2 关于window对象 665
  - 27.3.3 属性 667
  - 27.3.4 方法 700
  - 27.3.5 事件处理程序 754
- 27.4 frame元素对象 759
  - 27.4.1 语法 759
  - 27.4.2 关于frame对象 759
  - 27.4.3 属性 760
- 27.5 frameset元素对象 765
  - 27.5.1 语法 765
  - 27.5.2 关于frameset对象 766
  - 27.5.3 属性 767
- 27.6 iframe元素对象 771
  - 27.6.1 语法 771
  - 27.6.2 关于iframe对象 772
  - 27.6.3 属性 772
- 27.7 popup对象 776
  - 27.7.1 语法 776
  - 27.7.2 关于popup对象 777
  - 27.7.3 属性 777
  - 27.7.4 方法 778
- 第28章 location对象和history对象 781
  - 28.1 location对象 781
    - 28.1.1 语法 782
    - 28.1.2 关于location对象 782
    - 28.1.3 属性 784
    - 28.1.4 方法 795
  - 28.2 history对象 798
    - 28.2.1 语法 798
    - 28.2.2 关于history对象 798
    - 28.2.3 属性 799
    - 28.2.4 方法 800

## &lt;&lt;JavaScript宝典&gt;&gt;

- 第29章 document对象和body对象 805
  - 29.1 document对象 806
    - 29.1.1 语法 808
    - 29.1.2 关于document对象 808
    - 29.1.3 属性 809
    - 29.1.4 方法 848
    - 29.1.5 事件处理程序 870
  - 29.2 body元素对象 871
    - 29.2.1 语法 872
    - 29.2.2 关于body对象 872
    - 29.2.3 属性 873
    - 29.2.4 方法 877
    - 29.2.5 事件处理程序 879
  - 29.3 TreeWalker对象 879
    - 29.3.1 语法 879
    - 29.3.2 关于TreeWalker对象 879
    - 29.3.3 属性 880
    - 29.3.4 方法 881
- 第30章 link和anchor对象 883
- 第31章 image、area、map和canvas对象 891
  - 31.1 image和img元素对象 891
    - 31.1.1 语法 892
    - 31.1.2 关于image对象 893
    - 31.1.3 属性 894
    - 31.1.4 事件处理程序 908
  - 31.2 area元素对象 910
    - 31.2.1 语法 910
    - 31.2.2 关于area对象 911
    - 31.2.3 属性 912
  - 31.3 map元素对象 913
    - 31.3.1 语法 914
    - 31.3.2 关于map对象 914
    - 31.3.3 属性 914
  - 31.4 canvas元素对象 917
    - 31.4.1 语法 918
    - 31.4.2 关于canvas对象 918
    - 31.4.3 属性 921
    - 31.4.4 方法 923
- 第32章 event对象 927
  - 32.1 事件 927
    - 32.1.1 事件的内容和事件发生时间 928
    - 32.1.2 静态event对象 928
  - 32.2 事件传播 929
    - 32.2.1 仅用于NN4的事件传播 929

## &lt;&lt;JavaScript宝典&gt;&gt;

- 32.2.2 IE4+事件传播 931
- 32.2.3 W3C事件传播 935
- 32.3 引用事件对象 941
- 32.4 绑定事件 942
  - 32.4.1 使用标记特性绑定事件 942
  - 32.4.2 使用对象特性绑定事件 943
  - 32.4.3 使用IE附加功能绑定事件 944
  - 32.4.4 通过W3C监听器绑定事件 944
  - 32.4.5 跨浏览器的事件绑定解决方案 945
- 32.5 事件对象兼容性 946
- 32.6 事件模型详析 948
  - 32.6.1 以跨平台方式检查修改键 948
  - 32.6.2 以跨平台方式捕获按键 950
- 32.7 事件类型 951
  - 32.7.1 IE4+和NN6+/W3C中的事件类型 952
  - 32.7.2 语法 954
  - 32.7.3 关于event对象 955
  - 32.7.4 属性 955
- 32.8 NN6+/Moz的event对象 974
  - 32.8.1 语法 975
  - 32.8.2 关于event对象 975
  - 32.8.3 属性 976
  - 32.8.4 方法 994
- 附录A JavaScript和浏览器对象快速参考 997
- 附录B 本书配套光盘内容 1011

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>