

TURING

图灵程序设计丛书

Web 开发系列

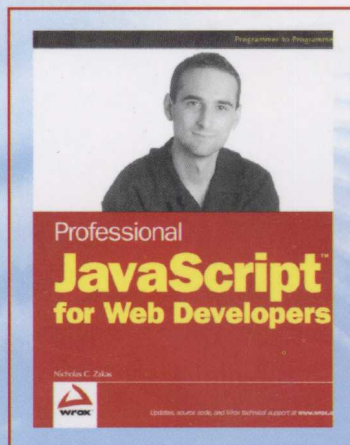


Professional JavaScript
for Web Developers

JavaScript 高级程序设计

Nicholas C. Zakas 著
曹力 张欣 等译

- JavaScript 最新经典教程
- Amazon 超级畅销书
- Ajax 程序员必备



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

JavaScript高级程序设计

Professional JavaScript for Web Developers

Nicholas C. Zakas 著

曹力 张欣 等译

 **人民邮电出版社**
POSTS & TELECOM PRESS

版 权 声 明

Original edition, entitled *Professional JavaScript for Web Developers* by Nicholas C.Zakas, by Wiley Publishing, Inc. Copyright © 2005 by Wiley Publishing, Inc.

All rights reserved. This translation published under license.

The Wrox Brand trade dress is a trademark of Wiley Publishing, Inc. in the United States and/or other countries. Used by permission.

Translation edition published by POSTS & TELECOM PRESS Copyright © 2006 .

本书简体中文版由 Wiley Publishing, Inc.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

Wrox 商标是 Wiley 出版公司在美国及其他国家使用的商标，经许可后才能使用。

版权所有，侵权必究。

译者序

JavaScript 是赋予网页活力与交互性的主要手段之一，全世界每天都有无数网页在依靠 JavaScript 完成各种关键任务。随着 Web 2.0 和 Ajax 进入主流，JavaScript 已经被推到了舞台中心，使用它来开发更大更复杂的程序势在必行，更多开发人员和 Web 设计师需要熟练掌握 JavaScript。人们发现，由于 JavaScript 同时具有面向对象、过程和函数型语言三类语言的特性，将灵活性与强大功能融于一身，要想真正掌握到能够开发复杂程序的程度，其实并不容易。而 JavaScript 方面完备的开发工具和好书的缺乏，更使这种情况雪上加霜。

让人高兴的是，人民邮电出版社及时地引进了本书，弥补了这一空白。书中世界知名的 JavaScript 专家，用通俗易懂的语言，将 JavaScript 各种相关的技术娓娓道来。在简要讲述了 JavaScript 的语言核心之后，很快转向高级主题，紧贴 Web 开发者目前面对的各种问题。更难得的是，作者还涵盖了当今各个流行浏览器的区别，并帮助读者解决这些问题。

本书主要针对有一定开发经验的读者。刚学 JavaScript 的朋友，可以按部就班成为高手，而已经是高手的朋友，则可以将本书作为参考手册。初学 JavaScript 的朋友，可以选择人民邮电出版社即将出版的最佳入门书：Tom Negrino 的《JavaScript 基础教程》（英文名 *JavaScript for the World Wide Web*）。此外，Jeremy Keith 的《JavaScript DOM 编程艺术》（英文名 *DOM Scripting*，中文版人民邮电出版社即将出版）是目前 Amazon 上排名最高的 JavaScript 图书，书中将如何结合 JavaScript 和 DOM 创造各种绚丽的动态效果发挥得淋漓尽致。需要参考书的朋友，可以考虑 David Flanagan 的《JavaScript 权威指南》，但国内中文版不是最新版本，许多内容已经过时。此外，Danny Goodman 的《JavaScript Bible》是一部类似的大部头著作，但是口碑不如前者，国内尚无译本，需要指出的是，市场上有一本名为《JavaScript 编程宝典》的书，封面装帧完全模仿此书而且价格不菲，其实与 Goodman 完全没有关系，实际上是国人欺世之作，请大家不要上当。

本书第 1~5 章由张欣翻译，第 6~15 章由曹力翻译，第 16~20 章由王霄翻译，全书由张欣统稿、润色及审校。我们深深地感谢我们的家人和朋友。在翻译过程中，他们给予了我们莫大的关心、支持和帮助。限于我们的水平，译文中的疏漏和错误在所难免，请广大读者批评指正。

译者

2006 年 7 月

前 言

虽然，服务器端的 Netscape Enterprise Server 和 Active Server Pages (ASP) 都曾经支持 JavaScript，但它主要还是 Web 浏览器使用的客户端脚本语言。目前，JavaScript 主要用于帮助开发者与网页和 Web 浏览器窗口本身进行交互。

不太严格地说，JavaScript 是基于 Java 的¹，而 Java 是一种面向对象程序设计语言，因能够以嵌入式 applet 的形式用于 Web 而流行起来。虽然 JavaScript 的语法和程序设计方法都与 Java 相似，但它并非 Java 的简化版本。相反，JavaScript 是一种独立的语言，它存在于全世界所有的 Web 浏览器中，能够增强用户与 Web 站点和 Web 应用程序之间的交互。

本书从最早期 Netscape 浏览器中的 JavaScript 开始讲起，直到当前它对 XML 和 Web 服务的具体支持。你将学到如何扩展这种语言，以使它适应特殊的需求；学到如何在没有 Java 或隐藏框架这些媒介的情况下，创建无缝的客户机—服务器通信。简而言之，你将学到如何应用 JavaScript 解决 Web 开发者面对的各种问题。

本书内容

本书针对开发人员介绍 JavaScript，包括很多高级的、重要的特性。

本书开始部分探讨了 JavaScript 的起源及其迄今为止的发展。之后详细介绍了构成 JavaScript 实现的各个组件，着重介绍了 ECMAScript 和 DOM（文档对象模型）这样的标准。此外还讨论了在不同 Web 浏览器中使用的 JavaScript 实现的不同。

基于上述讨论，本书开始介绍 JavaScript 的基本概念，包括面向对象程序设计、继承以及它在各种标记语言（如 HTML）中的使用。在探讨了浏览器检测技术，并介绍了在 JavaScript 中如何使用正则表达式后，本书对事件和事件处理进行了深度考察。之后，应用所有这些知识来创建动态的用户界面。

本书最后一部分讨论在 Web 应用程序中部署 JavaScript 解决方案的相关问题，包括错误处理、调试、安全性、优化/模糊化、XML 和 Web 服务。

读者对象

本书针对以下三类读者群：

- 熟悉面向对象程序设计方法的有经验的开发人员。他们由于 JavaScript 与传统的 OO 语言（如 Java 和 C++）相关，所以想学习它。

1. 从语言特性来说，JavaScript 与 Java 并不相同，但 JavaScript 开发过程中的确是以 Java 为模仿对象的，请参考 JavaScript 之父 Brendan Eich 的回忆文章。——编者注

- 尝试提高 Web 站点和 Web 应用程序可用性的 Web 应用程序开发人员。
- 想更好地理解 JavaScript 语言的初学者。

此外，本书也适用熟悉下列相关技术的读者：

- XML
- XSLT
- Java
- Web Services
- HTML
- CSS

本书不适合没有计算机科学基础背景的初学者，也不适合只想在 Web 站点添加一些简单用户交互特性的读者。这些读者应该阅读人民邮电出版社即将出版的《JavaScript 基础教程》。

环境配置

要运行本书中的示例，需要下列软件：

- Windows 2000、Windows Server 2003、Windows XP 或 Mac OS X；
- IE 5.5 或更高版本（Windows）、Mozilla 1.0 或更高版本（所有平台）、Opera 7.5 或更高版本（所有平台）、Safari 1.2 或更高版本（Mac OS X）。

可以从图灵网站 www.turingbook.com 或 <http://www.wrox.com> 的本书配套网页下载书中示例的完整源代码。

本书结构

第 1 章 JavaScript 是什么

这一章解释了 JavaScript 的起源，包括它是怎样产生的，如何发展的以及现状如何。介绍到的概念包括 JavaScript 与 ECMAScript、DOM（文档对象模型）和 BOM（浏览器对象模型）之间的关系。此外还讨论了与 ECMA 和 W3C 的有关标准。

第 2 章 ECMAScript 基础

这一章分析了 JavaScript 所基于的核心技术——ECMAScript。从变量和函数的声明到原始值与引用值的使用和理解，本章描述了编写 JavaScript 代码必需的基础语法和概念。

第 3 章 对象基础

这一章的重点是用 JavaScript 进行面向对象的程序设计（OOP）的基础，涵盖的主题包括用各种方法定义定制的对象、创建对象实例以及了解 JavaScript 和 Java 中 OOP 的异同。

第 4 章 继承

这一章继续探讨 JavaScript 中的 OOP，描述继承机制的工作机理，其中讨论了各种实现继承的方法，并且还比较了它们与 Java 中继承性的异同。

第 5 章 浏览器中的 JavaScript

这一章解释了如何把 JavaScript 嵌入到用各种语言（如 HTML、SVG 和 XUL）编写的网页中。此外还介绍了 BOM（浏览器对象模型）以及它的各种对象和接口。

第 6 章 DOM 基础

这一章介绍了在 JavaScript 中实现的 DOM，包括专门适用于 Web 开发者的 DOM 概念。之后，将这些概念应用在使用 HTML、SVG 和 XUL 编写的示例中。

第 7 章 正则表达式

这一章的重点是 JavaScript 实现的正则表达式，这是进行数据验证和字符串操作的强有力工具。本章探讨了正则表达式的起源、语法以及它在各种程序设计语言中的用法。本章的结尾探讨了正则表达式在 JavaScript 实现中的异同。

第 8 章 检测浏览器和操作系统

这一章解释了编写能在各种 Web 浏览器上运行的 JavaScript 脚本的重要性。它讨论了两种检测浏览器的方法，即对象/特性检测法和 user-agent 字符串检测法，并逐一列出了每种方法的优点和缺点。

第 9 章 事件

这一章讨论了 JavaScript 中最重要的概念之一——事件。事件是把 JavaScript 和任何用标记语言编写的 Web 用户界面连接在一起的主要方法。本章介绍了事件处理的各种方法和事件流的概念（包括冒泡和捕获）。

第 10 章 高级 DOM 技术

这一章介绍了一些较高级的 DOM 特性，包括范围和样式表操作。本章给出了一个例子，用于说明何时以及如何使用这些技术，此外还讨论了如何实现跨浏览器的支持。

第 11 章 表单和数据完整性

这一章讨论了使用表单时数据验证的重要性。在介绍处理验证的方法时，还应用了前面介绍

的概念，如正则表达式、事件和 DOM 操作。

第 12 章 表格排序

这一章应用前面介绍的多种特性来实现客户端的动态表格排序。其中包括用 JavaScript 进行排序的深度讨论，以及如何用事件、DOM 操作和比较运算符开发各种 Web 浏览器都能使用的通用表格排序协议。

第 13 章 拖放

这一章解释了拖放的概念以及它们在 JavaScript 和 Web 浏览器中的应用。其中讨论了系统拖放和模拟拖放的概念，章末创建了一个能跨浏览器使用的标准拖放界面。

第 14 章 错误处理

这一章通过讨论 try...catch 语句和 onerror 事件处理函数的用法来介绍 JavaScript 中事件处理的概念。另一个主题是用 throw 语句创建定制的错误消息以及 JavaScript 调试器的用法。

第 15 章 JavaScript 中的 XML

这一章介绍了 JavaScript 用于读取和操作 XML（可扩展标记语言）数据的特性，解释了各种 Web 浏览器中的支持和对象的不同，还为跨浏览器编码提供了建议。此外，本章还介绍了如何用 XSLT 语言转换客户端的 XML 数据。

第 16 章 客户端与服务器端的通信

这一章探讨了 JavaScript 与服务器通信的各种方法。这些方法包括使用 cookie 和基于 JavaScript 的 HTTP 请求。此外，这一章还解释了如何在不使用隐藏框架的情况下实现 GET 和 POST HTTP 请求。

第 17 章 Web 服务

这一章介绍了如何用 JavaScript 提供 Web 服务，其中讨论了在 IE 和 Mozilla 中使用的不同方法，还为没有内置 Web 服务支持的浏览器提供了一种添加 Web 服务的基本解决方案。

第 18 章 与插件进行交互

这一章解释了 JavaScript 与各种浏览器插件（如 Java applet、SVG 文档和 ActiveX 控件）之间的通信方法。其他主题包括如何编写能与 JavaScript 一起使用的插件。

第 19 章 部署问题

这一章的重点是完成 JavaScript 编码后的操作。本章讨论了在把 JavaScript 解决方案部署到 Web 站点或 Web 应用程序之前要做的一些操作。其中的主题包括安全性、国际化、优化和知识产权保护。

第 20 章 JavaScript 的未来

这一章考查了 JavaScript 的未来，介绍了这种语言的发展方向。其中讨论了 ECMAScript 的 ECMAScript 4 和 XML。

本书约定

为了帮助读者更充分地利用本书，方便阅读，我们在本书中采用如下约定：

在这样的矩形框中给出的内容都是重要的、不应忘记的信息，它与周围的内容直接相关。

对当前讨论的主题可能有一些提示、技巧和旁注，这些都将用楷体显示。

正文中还包括以下样式：

- 在初次介绍重要术语时，使用楷体突出强调；
- 用 Ctrl+A 这样的形式说明键盘按键；
- 正文中的文件名、URL 和代码使用 Courier 字体显示；
- 代码有两种形式：代码示例中，新出现的代码或重要代码用灰色背景突出显示；对当前讨论不太重要的代码或者是前面已经出现过的代码不用灰色背景强调。

```
In code examples we highlight new and important code with a gray background.
```

```
The gray highlighting is not used for code that's less important in the present context or has been shown before.
```

源代码

在使用本书中的例子时，你可以手工键入所有代码，也可以直接使用本书在网上随附的源代码文件。本书中用到的所有源代码文件都可以从 www.turingbook.com 或 www.wrox.com 下载。访问 Wrox 网站时，只要找到本书的英文版书名 (*Professional JavaScript for Web Developers*) [可以使用搜索 (Search) 框，也可以使用某个书目列表]，并点击该书详细信息网页上的下载代码 (Download Code) 链接，就可以得到本书的所有源代码。

由于会有许多书名字雷同，最佳的方法是利用 ISBN 搜索，本书的英文版 ISBN 是 0-7645-7908-8。

下载代码之后,你只需用最习惯的压缩工具解压就可以了。此外,还可以在 Wrox 的主下载页面 <http://www.wrox.com/dynamic/books/download.aspx> 处找到本书及其他 Wrox 出版的书的配套代码。

勘误表

我们一直努力确保代码或正文中没有错误。不过,是人都会犯错误。如果你发现了我们出版的书中的错误,不论是拼写错误还是代码错误,都请告知我们,我们将非常感谢。这样能节省其他读者的时间,同时还能帮助我们提高内容的准确性。

在 <http://www.wrox.com> 处,用 Search 框或名字列表找到本书的名字,然后在本书的主页面上点击 Book Errata 链接,可以找到本书的勘误表。在这个页面上可以找到由 Wrox 的编辑发布的、已经发现的所有的错误。在 www.wrox.com/misc-pages/booklist.shtml 处可以找到 Wrox 出版的所有书的列表,其中有每本书的勘误表的链接。

如果在 Book Errata 页面上没有找到你发现的错误,请访问 www.wrox.com/contact/techsupport.shtml 页面,填写其中的表单,把你发现的错误发送给我们。我们将检查你提交的信息,如果正确,就会把它发布在本书的勘误表页面上,并在本书以后的版本中纠正这一错误。

p2p.wrox.com

要与作者或其他人讨论有关问题,请加入 P2P 论坛 (p2p.wrox.com)。这个论坛是一个基于 Web 的系统,你可以在此发表有关 Wrox 图书和相关技术的消息,并与其他读者和技术用户交流。论坛针对你感兴趣的主题提供订购功能,论坛新发布相关消息时,会通过电子邮件通知你。Wrox 作者、编辑、其他行业专家以及其他读者也会造访这些论坛。

在 <http://p2p.wrox.com> 上,你会看到许多论坛,这些论坛不仅可以帮助你阅读本书,还有助于你开发自己的应用程序。要想加入论坛,只需遵循以下几个步骤:

- (1) 访问 p2p.wrox.com, 并点击 Register (注册) 链接;
- (2) 阅读使用条文, 并点击 Agree (同意);
- (3) 填写加入论坛的必要信息, 如果想提供其他可选信息, 也可以相应填写, 点击 Submit (提交);
- (4) 你将收到一个电子邮件, 其中说明如何验证你的账户, 并完成加入过程。

如果只是阅读论坛中的消息, 无需加入 p2p。不过, 如果你想发布自己的消息, 就必须加入论坛。

一旦加入, 就可以发布新的消息了, 还可以对其他用户发布的消息做出响应。你在任何时刻都可以在 Web 上阅读消息。如果希望某个论坛能通过电子邮件向你发送新发布的消息, 请点击论坛列表中该论坛名旁边的 Subscribe to this Forum (订购此论坛) 图标。

要了解如何使用 Wrox p2p 的更多信息, 请阅读 p2p FAQ, 在此解释了这个论坛软件如何工

作的相关问题，另外还回答了与 p2p 和 Wrox 图书有关的许多常见问题。要阅读 FAQ，可以点击任何 p2p 页面上的 FAQ 链接。

本书索引可以从图灵网站的本书配套网页下载。索引中的页码为英文原书页码，与书中边栏的页码一致。

致谢

尽管本书的封面上只署了一个名字，但是完成本书的却不只一人。没有众人的帮助，本书是不可能完成的。

首先要感谢的是 Wiley 出版公司的每一个人，尤其是 Jim Minatel 和 Sharon Nash，他们为每位新作者提供了所有必需的指导和支持。

有很多人就一本好的 JavaScript 书应该具有什么内容为我提供了许多意见，包括 Keith Ciociola、Ken Fearnley、John Rajan 和 Douglas Swatski，向他们致谢。

特别感谢审阅本书初稿的每个人，他们是 Erik Arvidsson、Bradley Baumann、Guilherme Blanco、Douglas Crockford、Jean-Luc David、Emil A. Eklund、Brett Fielder、Jeremy McPeak 和 Micha Schopman。他们的每一条意见都使得本书更趋于完美。

感谢 Ed Bernard 和 Frances Bernard 医生，他们在本书编写过程中以及过去几年间，使我保持最好的健康状况。

最后，感谢我的家人——妈妈、爸爸和 Greg 还有我善解人意的女友 Emily。他们的爱和支持帮助我成就本书。

目 录

第 1 章 JavaScript 是什么	1	2.9.1 一元运算符	28
1.1 历史简述	1	2.9.2 位运算符	32
1.2 JavaScript 实现	2	2.9.3 Boolean 运算符	37
1.2.1 ECMAScript	3	2.9.4 乘性运算符	40
1.2.2 DOM	5	2.9.5 加性运算符	41
1.2.3 BOM	8	2.9.6 关系运算符	42
1.3 小结	8	2.9.7 等性运算符	43
第 2 章 ECMAScript 基础	9	2.9.8 条件运算符	45
2.1 语法	9	2.9.9 赋值运算符	45
2.2 变量	10	2.9.10 逗号运算符	46
2.3 关键字	12	2.10 语句	46
2.4 保留字	12	2.10.1 if 语句	46
2.5 原始值和引用值	13	2.10.2 迭代语句	47
2.6 原始类型	13	2.10.3 有标签的语句	48
2.6.1 typeof 运算符	14	2.10.4 break 语句和 continue	48
2.6.2 Undefined 类型	14	语句	48
2.6.3 Null 类型	15	2.10.5 with 语句	50
2.6.4 Boolean 类型	15	2.10.6 switch 语句	50
2.6.5 Number 类型	15	2.11 函数	51
2.6.6 String 类型	17	2.11.1 无重载	53
2.7 转换	18	2.11.2 arguments 对象	53
2.7.1 转换成字符串	18	2.11.3 Function 类	54
2.7.2 转换成数字	19	2.11.4 闭包	56
2.7.3 强制类型转换	20	2.12 小结	57
2.8 引用类型	22	第 3 章 对象基础	58
2.8.1 Object 类	22	3.1 面向对象术语	58
2.8.2 Boolean 类	23	3.1.1 面向对象语言的要求	58
2.8.3 Number 类	23	3.1.2 对象的构成	59
2.8.4 String 类	24	3.2 对象应用	59
2.8.5 instanceof 运算符	28	3.2.1 声明和实例化	59
2.9 运算符	28	3.2.2 对象引用	59
		3.2.3 对象废除	59

3.2.4 早绑定和晚绑定	60	5.1.5 隐藏还是不隐藏	113
3.3 对象的类型	60	5.1.6 <noscript/>标签	113
3.3.1 本地对象	60	5.1.7 XHTML 中的改变	114
3.3.2 内置对象	70	5.2 SVG 中的 JavaScript	116
3.3.3 宿主对象	75	5.2.1 SVG 基础	116
3.4 作用域	75	5.2.2 SVG 中的<script/>标签	117
3.4.1 公用、受保护和私有作用域	75	5.2.3 SVG 中的标签放置	118
3.4.2 静态作用域并非静态的	76	5.3 BOM	119
3.4.3 关键字 this	76	5.3.1 window 对象	119
3.5 定义类或对象	78	5.3.2 document 对象	130
3.5.1 工厂方式	78	5.3.3 location 对象	133
3.5.2 构造函数方式	80	5.3.4 navigator 对象	135
3.5.3 原型方式	80	5.3.5 screen 对象	136
3.5.4 混合的构造函数/原型方式	81	5.4 小结	137
3.5.5 动态原型方法	82	第 6 章 DOM 基础	138
3.5.6 混合工厂方式	83	6.1 什么是 DOM?	138
3.5.7 采用哪种方式	84	6.1.1 XML 简介	138
3.5.8 实例	84	6.1.2 针对 XML 的 API	141
3.6 修改对象	86	6.1.3 节点的层次	141
3.6.1 创建新方法	86	6.1.4 特定语言的 DOM	144
3.6.2 重定义已有方法	87	6.2 对 DOM 的支持	145
3.6.3 极晚绑定	88	6.3 使用 DOM	145
3.7 小结	88	6.3.1 访问相关的节点	145
第 4 章 继承	89	6.3.2 检测节点类型	146
4.1 继承机制实例	89	6.3.3 处理特性	147
4.2 继承机制的实现	90	6.3.4 访问指定节点	148
4.2.1 继承的方式	90	6.3.5 创建和操作节点	150
4.2.2 一个更实际的例子	96	6.4 HTML DOM 特征功能	155
4.3 其他继承方式	100	6.4.1 让特性像属性一样	155
4.3.1 zInherit	100	6.4.2 table 方法	156
4.3.2 xbObjects	104	6.5 遍历 DOM	158
4.4 小结	108	6.5.1 NodeIterator	158
第 5 章 浏览器中的 JavaScript	109	6.5.2 TreeWalker	163
5.1 HTML 中的 JavaScript	109	6.6 测试与 DOM 标准的一致性	165
5.1.1 <script/>标签	109	6.7 DOM Level 3	166
5.1.2 外部文件格式	110	6.8 小结	166
5.1.3 内嵌代码和外部文件	111	第 7 章 正则表达式	167
5.1.4 标签放置	111	7.1 正则表达式支持	167

7.1.1	使用 RegExp 对象	168	8.4.1	方法学	202
7.1.2	扩展的字符串方法	169	8.4.2	第一步	202
7.2	简单模式	170	8.4.3	检测 Opera	204
7.2.1	元字符	170	8.4.4	检测 Konqueror/Safari	206
7.2.2	使用特殊字符	170	8.4.5	检测 IE	208
7.2.3	字符类	172	8.4.6	检测 Mozilla	209
7.2.4	量词	174	8.4.7	检测 Netscape Communicator 4.x	210
7.3	复杂模式	177	8.5	平台/操作系统检测脚本	211
7.3.1	分组	177	8.5.1	方法学	211
7.3.2	反向引用	178	8.5.2	第一步	212
7.3.3	候选	179	8.5.3	检测 Windows 操作系统	212
7.3.4	非捕获性分组	180	8.5.4	检测 Macintosh 操作系统	214
7.3.5	前瞻	181	8.5.5	检测 Unix 操作系统	214
7.3.6	边界	182	8.6	全部脚本	215
7.3.7	多行模式	183	8.7	例子: 登录页面	219
7.4	理解 RegExp 对象	184	8.8	小结	224
7.4.1	实例属性	184	第 9 章	事件	225
7.4.2	静态属性	185	9.1	今天的事件	225
7.5	常用模式	186	9.2	事件流	226
7.5.1	验证日期	187	9.2.1	冒泡型事件	226
7.5.2	验证信用卡号	188	9.2.2	捕获型事件	227
7.5.3	验证电子邮件地址	192	9.2.3	DOM 事件流	228
7.6	小结	193	9.3	事件处理函数/监听函数	229
第 8 章	检测浏览器和操作系统	194	9.3.1	IE	230
8.1	navigator 对象	194	9.3.2	DOM	231
8.2	检测浏览器的方式	194	9.4	事件对象	232
8.2.1	对象/特征检测法	194	9.4.1	定位	233
8.2.2	user-agent 字符串检测法	195	9.4.2	属性/方法	233
8.3	user-agent 字符串简史	196	9.4.3	相似性	235
8.3.1	Netscape Navigator 3.0 与 IE3.0	196	9.4.4	区别	238
8.3.2	Netscape Communicator 4.0 与 IE 4.0	197	9.5	事件的类型	240
8.3.3	IE 5.0 及更高版本	198	9.5.1	鼠标事件	240
8.3.4	Mozilla	198	9.5.2	键盘事件	244
8.3.5	Opera	200	9.5.3	HTML 事件	246
8.3.6	Safari	201	9.5.4	变化事件	251
8.3.7	结语	201	9.6	跨平台的事件	252
8.4	浏览器检测脚本	201	9.6.1	EventUtil 对象	252
			9.6.2	添加/删除事件处理函数	252

9.6.3 格式化 event 对象	254	11.4.1 访问选项	309
9.6.4 获取事件对象	258	11.4.2 获取/更改选中项	309
9.6.5 示例	259	11.4.3 添加选项	310
9.7 小结	260	11.4.4 删除选项	311
第 10 章 高级 DOM 技术	261	11.4.5 移动选项	312
10.1 样式编程	261	11.4.6 重新排序选项	313
10.1.1 DOM 样式的方法	263	11.5 创建自动提示的文本框	313
10.1.2 自定义鼠标提示	264	11.5.1 匹配	314
10.1.3 可折叠区域	265	11.5.2 内部机制	314
10.1.4 访问样式表	266	11.6 小结	316
10.1.5 最终样式	270	第 12 章 表格排序	317
10.2 innerText 和 innerHTML	271	12.1 起点——数组	317
10.3 outerText 和 outerHTML	273	12.2 对单列的表格排序	319
10.4 范围	274	12.2.1 比较函数	320
10.4.1 DOM 中的范围	274	12.2.2 sortTable() 函数	320
10.4.2 IE 中的范围	284	12.3 对多列表格进行排序	323
10.4.3 范围在实际中的应用	288	12.3.1 比较函数生成器	323
10.5 小结	288	12.3.2 修改 sortTable() 方法	324
第 11 章 表单和数据完整性	289	12.3.3 逆序排列	325
11.1 表单基础	289	12.3.4 对不同的数据类型进行排序	327
11.2 对<form/>元素进行脚本编写	291	12.3.5 高级排序	330
11.2.1 获取表单的引用	291	12.4 小结	334
11.2.2 访问表单字段	291	第 13 章 拖放	335
11.2.3 表单字段的共性	292	13.1 系统拖放	335
11.2.4 聚焦于第一个字段	292	13.1.1 拖放事件	336
11.2.5 提交表单	293	13.1.2 数据传输对象dataTransfer	341
11.2.6 仅提交一次	294	13.1.3 dragDrop() 方法	345
11.2.7 重置表单	295	13.1.4 优点及缺点	346
11.3 文本框	295	13.2 模拟拖放	346
11.3.1 获取/更改文本框的值	296	13.2.1 代码	347
11.3.2 选择文本	297	13.2.2 创建放置目标	349
11.3.3 文本框事件	298	13.2.3 优点及缺点	352
11.3.4 自动选择文本	298	13.3 zDragDrop	352
11.3.5 自动切换到下一个	299	13.3.1 创建可拖动元素	352
11.3.6 限制 textarea 的字符数	300	13.3.2 创建放置目标	353
11.3.7 允许/阻止文本框中的字符	301	13.3.3 事件	353
11.3.8 使用上下按键操作数字文本	306	13.3.4 例子	354
11.4 列表框和组合框	308	13.4 小结	355

第 14 章 错误处理	356	16.1.1 cookie 的成分	416
14.1 错误处理的重要性	356	16.1.2 其他安全限制	417
14.2 错误和异常	357	16.1.3 JavaScript 中的 cookie	417
14.3 错误报告	358	16.1.4 服务器端的 cookie	419
14.3.1 IE (Windows)	358	16.1.5 在客户端与服务器端之间 传递 cookie	422
14.3.2 IE (MacOS)	359	16.2 隐藏框架	423
14.3.3 Mozilla (所有平台)	359	16.3 HTTP 请求	426
14.3.4 Safari (MacOS)	360	16.3.1 使用 HTTP 首部	428
14.3.5 Opera 7 (所有平台)	361	16.3.2 实现的复制品	429
14.4 处理错误	362	16.3.3 进行 GET 请求	430
14.4.1 onerror 事件处理函数	362	16.3.4 进行 POST 请求	430
14.4.2 try...catch 语句	365	16.4 LiveConnect 请求	431
14.5 调试技巧	370	16.4.1 进行 GET 请求	431
14.5.1 使用警告框	370	16.4.2 进行 POST 请求	433
14.5.2 使用 Java 控制台	371	16.5 智能 HTTP 请求	435
14.5.3 将消息写入 JavaScript 控制台 (仅限 Opera 7+)	372	16.5.1 get() 方法	435
14.5.4 抛出自定义错误	372	16.5.2 post() 方法	438
14.5.5 JavaScript 校验器	373	16.6 实际使用	439
14.6 调试器	374	16.7 小结	439
14.6.1 Microsoft Script Debugger	374	第 17 章 Web 服务	440
14.6.2 Venkman	376	17.1 Web 服务快速入门	440
14.7 小结	383	17.1.1 Web 服务是什么?	440
第 15 章 JavaScript 中的 XML	384	17.1.2 WSDL	441
15.1 浏览器中的 XML DOM 支持	384	17.2 IE 中的 Web 服务	443
15.1.1 IE 中的 XML DOM 支持	384	17.2.1 使用 WebService 组件	444
15.1.2 Mozilla 中 XML DOM 支持	388	17.2.2 WebService 组件例子	445
15.1.3 通用接口	393	17.3 Mozilla 中的 Web 服务	447
15.2 浏览器中的 XPath 支持	403	17.3.1 加强的特权	447
15.2.1 XPath 简介	403	17.3.2 使用 SOAP 方法	448
15.2.2 IE 中的 XPath 支持	404	17.3.3 使用 WSDL 代理	451
15.2.3 Mozilla 中的 XPath 支持	404	17.4 跨浏览器的方案	454
15.3 浏览器中的 XSLT 支持	408	17.4.1 WebService 对象	454
15.3.1 IE 中的 XSLT 支持	410	17.4.2 Temperature 服务	456
15.3.2 Mozilla 中 XSLT 支持	413	17.4.3 使用 TemperatureService 对象	458
15.4 小结	415	17.5 小结	458
第 16 章 客户端与服务器端的通信	416	第 18 章 与插件进行交互	459
16.1 cookie	416	18.1 为何使用插件	459
		18.2 流行的插件	460