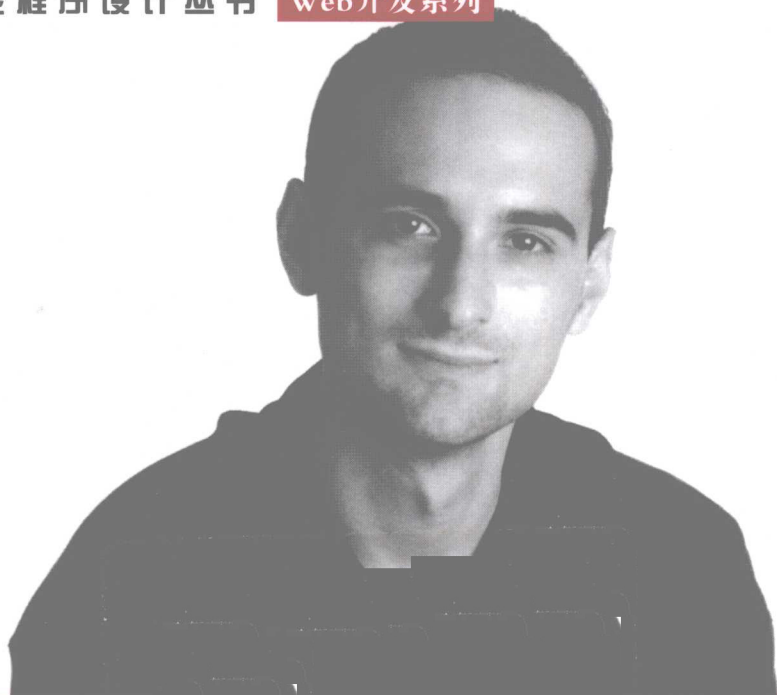


TURING

图灵程序设计丛书

Web开发系列

WILEY



Professional JavaScript for Web Developers **Second Edition**

JavaScript

高级程序设计 (第2版)

[美] Nicholas C. Zakas 著
李松峰 曹力 译

- JavaScript经典教程
- Amazon超级畅销书
- 前端开发人员必备



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Web开发系列



Professional JavaScript for Web Developers Second Edition

JavaScript

高级程序改订 (第2版)

[美] Nicholas C. Zakas 著
李松峰 曹力 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

JavaScript高级程序设计：第2版 / (美) 泽卡斯
(Zakas, N. C.) 著；李松峰，曹力译. — 北京：人民邮
电出版社，2010.7

(图灵程序设计丛书)

书名原文: Professional JavaScript for Web
Developers, 2nd Edition

ISBN 978-7-115-23095-9

I. ①J… II. ①泽… ②李… ③曹… III. ①
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第096984号

内 容 提 要

本书在上一版基础上进行了大幅度更新和修订，融入了近几年来 JavaScript 应用发展的最新成果，几乎涵盖了所有需要理解的重要概念和最新的 JavaScript 应用成果。从颇具深度的 JavaScript 语言基础到作用域(链)，从引用类型到面向对象编程，从极其灵活的匿名函数到闭包的内部机制，从浏览器对象模型(BOM)、文档对象模型(DOM)到基于事件的 Web 脚本设计，从 XML (E4X) 到 Ajax 及 JSON，从高级前端开发技术到前沿的客户端存储，从最佳编程实践到即将成为现实的 API，直至 JavaScript 未来的发展，全景式地展示了 JavaScript 高级程序设计的方方面面。

本书适合不同层次的 JavaScript/Web 开发人员阅读参考，也可作为高校相关专业课程的教材。

图灵程序设计丛书

JavaScript高级程序设计 (第2版)

-
- ◆ 著 [美] Nicholas C. Zakas
 - 译 李松峰 曹 力
 - 责任编辑 朱 巍
 - 执行编辑 毛倩倩

 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷

 - ◆ 开本: 800×1000 1/16
印张: 38.75
字数: 1092千字 2010年7月第2版
印数: 36 001 - 40 000册 2010年7月北京第1次印刷
著作权合同登记号 图字: 01-2009-5733 号

ISBN 978-7-115-23095-9

定价: 89.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original edition, entitled *Professional JavaScript for Web Developers 2nd Edition*, by Nicholas C. Zakas, ISBN 978-0-470-22780-0, published by John Wiley & Sons, Inc.

Copyright ©2009 by John Wiley & Sons, Inc., All rights reserved. This translation published under License.

Translation edition published by POSTS & TELECOM PRESS Copyright ©2010.

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书简体中文版由John Wiley & Sons, Inc.授权人民邮电出版社独家出版。

本书封底贴有John Wiley & Sons, Inc.激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

译者序

儒家经典《大学》将“格物致知”奉为做学问、养身性的最高境界。

尼古拉斯·扎卡斯重新修订的这个最新版本，为各层次的 JavaScript 爱好者和 Web 前端开发人员提供了一条“格物致知”的捷径。

新版本的原书不仅篇幅由原来的 600 多页增加到 800 页，而且几乎全部更新、重写了上一版的内容，删除了上一版中与今天的职业需求无关的主题，新增了大量比上一版更有价值、更能反映 JavaScript 最新发展成果的内容。从颇具深度的 JavaScript 语言基础到作用域（链），从 JavaScript 引用类型到面向对象编程，从极其灵活的匿名函数到闭包的内部机制，从浏览器对象模型（BOM）到客户端检测，从文档对象模型（DOM）到基于事件的 Web 脚本编程，从错误处理到前端调试，从 XML（E4X）到 Ajax 及 JSON，从高级前端开发技术到前沿的客户端存储，从最佳编程实践到展望即将成为现实的 API，直至 JavaScript 未来的发展。全书基本上囊括了 JavaScript 技术的各个方面，几乎涉及了 Web 前端开发的所有热门话题。值得一提的是，本书还涵盖了当前最受开发人员关注的 HTML5 和移动设备（如 iPhone）开发的内容。可以预见，本书一定会成为 Web 前端开发人员不可多得的经典之作。

需要提请读者注意的是，本书第 22 章讲到了 JavaScript 未来的变化，里面大部分讲的是 ECMAScript 4/JavaScript 2，而 ECMAScript 4 已经被放弃了，新标准是 ECMAScript 5。请读者参考 <http://www.ecmascript.org/>。感谢周涛（Snandy）指出此问题。

本书文前和第 1 章至第 17 章由李松峰翻译，第 18 章至第 22 章及附录由曹力翻译。武卫东老师审读斧正了序的翻译，责任编辑朱巍为本书早日出版多方协调，执行编辑毛倩倩发现了译稿中多处错译和漏译，排校负责人董秋霞、谢凌老师严把三道排校的质量关，谢廷晟全面审校了第 1 章至第 17 章，吴玺喆（George Wing）、吴生辉（千年一梦）、周裕波、梁超（LC）、张树恒（Shuzai）、罗永德（lyongde）也审校了前 17 章的部分内容，为确保本书翻译质量起到了重要作用，在此对上述老师和同学致以深深的谢意。

译者

2010 年 5 月

序

从诞生至今的大部分时间里，恐惧、咒骂、蔑视和误解一直与 JavaScript 如影随形。JavaScript 刚刚问世那几年，很多“严肃的程序员”都认为它不够严肃。

相比之下，.COM 泡沫时期加入 Web 开发行列的许多文科生，则普遍觉得 JavaScript 深不可测、晦涩难懂。就算那些耐力和韧性俱佳者能够把 JavaScript 琢磨得很透，但仍然摆脱不掉竞争中的各种浏览器提供的不同实现给他们带来的麻烦。凡此种种，最终导致粗制滥造的脚本越来越多。另一方面，拜 Web 前端代码的无比开放性所赐，各种坏习惯不断从一个站点被粘贴进另一个站点的源代码中。那些实现活该臭名昭著，可是，JavaScript 这门语言也因此被严重拖累，背上了不该有的坏名声。

2001 年前后（随着 Internet Explorer 6 的发布），浏览器实现已经大为改进，Web 开发实践也开始得到改善，呈现出了二者水乳交融的局面。作为 Ajax 核心的 XMLHttpRequest 对象正慢慢地为人们所认识，一种新的桌面风格的用户交互模式出现在浏览器中。允许 JavaScript 操作 Web 文档结构和内容的 DOM API 已经定型。而 CSS，不管人们如何曲解或者无视它，也无论浏览器开发商怎样丧心病狂地实现它，都已经成长得足够茁壮，它的美妙和反应敏捷令它能够与新的 Web 交互能力配合无间。最终，JavaScript 一扫颓势，变得令人惊诧、让人兴奋、使人敬畏。想想 2004 年第一次使用 Google Maps 时的情景吧，那种感觉你或许还记忆犹新。

Google Maps 是新兴应用程序的典型代表。这类浏览器编程与后端编程并重的应用程序，不禁令人对 Web 浏览器窗口中那块“画布”的未来浮想联翩。（除 Google Maps 之外，早在 2003 年就基于网页邮件客户端提供类似 Outlook 功能的 Oddpost，也是这类应用程序的一个了不起的先驱。）随着这类应用程序如雨后春笋般大量涌现，以及支持它们的浏览器的市场份额不断攀升，一个 Web 应用全面复兴的时代真的到来了。“Web 2.0”诞生了，Ajax 也成了“IT”技术。Web 似乎在一夜之间脱胎换骨，重新激发了人们的兴趣。而 JavaScript 作为唯一的 Web 编程语言，也变得更令人神往了。

令人神往，但用好它却不简单。JavaScript 以及在 DOM 和 BOM 中为其定义的 API 不一致的实现，给跨浏览器编程造成了比原本大得多的困难。前端设计行业还远未成熟。大学教学计划并没有（至今仍没有）做出相应的调整，以满足相关的培训要求。

到 2004 年底，JavaScript 无疑已经成为最重要的编程语言。但从学术角度看，它依然不具备进入一类学科资格。Web 虽然已经翻开了新的一页，但在培养足够的知识全面、训练有素的人才方面，我们依然面临着严峻的挑战。

为此，很多技术作者挺身而出，撰写了不少有关 JavaScript 的图书。几年来，这类书虽然也出了不下几十本，但总体来说仍然不尽如人意。其中有的在推销与落伍的浏览器有关的技术，有的在卖弄容易剪贴但却不好扩展和维护的技术。让人想不通的是，许多 JavaScript 图书让人觉得作者好像并不真正喜欢 JavaScript，或者他们不认为读者应该喜欢它，再或者他们根本不相信读者能够完全理解

JavaScript。

2005年，Nicholas C. Zakas 这本书的第一版面世，为前端工程领域奉献了一本真正的好书。当时，我和雅虎的同事们正在创建 YUI (Yahoo! User Interface Library, Yahoo! 用户界面库)，打算将其作为公司前端工程的基础，同时也借以推广我们这门新学问的最佳实践。每到周五，我们就聚到一间教室里讨论前端工程，也向大家讲解 JavaScript、CSS 以及在浏览器中创建 Web 应用程序的知识。我们从已出版的高级 JavaScript 及 DOM 脚本编程方面的图书中认真挑选了几本，想让新工程师通过它们掌握如何构建耐用、基于标准且容易维护的 Web 应用程序。Zakas 的书一出版，马上就被选为我们的 JavaScript 内部培训课本。

从那时起我们就一直使用他的书。我们一致认为这本书太有用了，于是就跟 Zakas 商量，让他加入雅虎帮我们建立公司的前端工程社区。

Zakas 在书中传达的理念与众不同——JavaScript 既需要严肃认真地对待，但也是完全可以理解和掌握的。如果你是个程序员，这本书会告诉你 JavaScript 与各种编程语言的关系，以及如何运用你已经习以为常的各种编程模式。你可以理解 JavaScript 的继承机制及其固有的动态特性（虽不合传统，但却十分自由十分强大），可以从 Zakas 这位尊重和理解 JavaScript 的同道那里学会欣赏 JavaScript 这门语言。

如果你曾经是一名文科生，在网络泡沫时期步入了这个行业，至今也没有转行，而且想要弥补自己在 JavaScript 方面的不足，你会发现 Zakas 是一位难得的良师益友。他可以帮你实现从“会做”到“做好”的转变。他能让你真正地理解这门严肃的学问。最重要的是，他不会让你先入为主地产生对这门语言应该理解多深的想法。相反，通过他严肃、耐心、通俗易懂的讲解，你自然而然地会对这门语言有同样深刻的认识。

本书是经过扩展、更新和改进后的第二版，删除了上一版中与今天的职业需求无关的主题，并用我们在 2005 年至 2008 年学习的新知识更新了剩余的内容。这几年是 JavaScript 发展的重要时期，Zakas 则始终位于最前沿孜孜不倦地学习这些新知识。他这些年一直在致力于建造新一代最流行的 Web 个人门户 (My Yahoo!)，以及开发 Web 上最受欢迎的站点 (Yahoo! 主页) 的新版本。Zakas 以他作为老师和作者的独特视角，筛选出由这些超复杂、超大型应用程序磨砺出的经验，并将这些经验融入到了本书的字里行间。

他给出的解决方案远远超出了一本好书的范畴，只有每天都与代码同呼吸共命运的人，才有可能与读者分享如此具有实用价值的知识。

说实话，本书新版的面世对我和各位读者而言真是个莫大的喜讯。因为它的内容比上一版更有价值、更能反映 JavaScript 最新的发展成果，因此也更加令人不可错过。

埃里克·米拉利亚 (Eric Miraglia)
YUI 高级技术经理 哲学博士
于加利福尼亚州森尼维耳市

前 言

从驱动全球商业、贸易及管理领域不计其数的复杂应用程序的角度来看，说 JavaScript 已经成为当今世界上最流行的编程语言一点都不为过。

JavaScript 是基于 Java 的一种非常松散的面向对象语言，也是 Web 开发中极受欢迎的一门语言。JavaScript，尽管它的语法和编程风格与 Java 都很相似，但它却不是 Java 的“轻量级”版本。JavaScript 是一种全新的动态语言，它植根于全球数亿网民都在使用的 Web 浏览器之中，致力于增强网站和 Web 应用程序的交互性。

在本书中，我们将对 JavaScript 追根溯源，从它在最早的 Netscape 浏览器中诞生谈起，一直谈到今天的它对 DOM 和 Ajax 的强大支持。读者将通过本书掌握如何运用和扩展这门语言，从而更好地满足自己的需求，以及如何实现客户端与服务器的无缝通信，而又不必求助于 Java 或隐藏的网页框架（frame 元素）。一言以蔽之，本书将教会你在面对各种常见的 Web 开发问题时，如何拿出自己的 JavaScript 解决方案。

本书内容

本书提供了 JavaScript 开发人员必须掌握的内容，全面涵盖了 JavaScript 的各种高级、有用的特性。

本书首先介绍了 JavaScript 的起源及其发展现状，随后讨论了构成 JavaScript 实现的各个组成部分，重点讲解了 ECMAScript 和 DOM 标准。此外，还对不同 Web 浏览器的 JavaScript 实现之间存在的差异，给出了相应的说明。

在此基础上，本书从讲解 JavaScript 的基本概念入手，探讨了 JavaScript 面向对象程序设计和继承的方式，以及如何在 HTML 等标记语言中使用它。在深入剖析了事件和事件处理之后，又解释了各种浏览器检测技术和在 JavaScript 中使用正则表达式的技巧。然后又通过综合运用前面学到的知识，展示了创建动态用户界面的方方面面。

本书最后一部分专门讨论了高级主题，涉及性能和内存优化、最佳实践以及对 JavaScript 未来的展望。

本书读者

本书将下列三类人员作为目标读者：

- (1) 熟悉面向对象编程、经验丰富而又打算学习 JavaScript 的开发人员，JavaScript 毕竟与 Java、C++ 等传统 OO 语言存在着诸多联系；
- (2) 有意提升自己网站和 Web 应用程序易用性的 Web 开发人员；
- (3) 希望全面深入地理解这门语言的初级 JavaScript 开发人员。

此外，本书也适合熟悉下列相关技术的读者阅读：

- (1) Java
- (2) PHP
- (3) ASP.NET
- (4) HTML
- (5) CSS
- (6) XML

本书不适合没有计算机基础知识的初学者，也不适合只想为网站添加简单交互功能的读者。建议这些朋友学习阅读 *Beginning JavaScript, 3rd Edition* (Wiley, 2007) 一书^①。

使用示例

要运行本书中的示例，需要安装下列软件：

- Windows 2000、Windows Server 2003、Windows XP、Vista 或 Mac OS X；
- Internet Explorer 6 以及更高的版本、Opera 9 以及更高的版本、Chrome 0.2 以及更高的版本、Safari 2 以及更高的版本。

完整的示例源代码可以从 <http://www.wrox.com/> 中下载（下载步骤见“源代码”一节）^②。

本书结构

本书共 22 章，各章简介如下。

第 1 章“JavaScript 简介”，讲述了 JavaScript 的起源：因何而生，如何发展，现状如何。涉及的概念主要有 JavaScript 与 ECMAScript 之间的关系、DOM (Document Object Model, 文档对象模型)、BOM (Browser Object Model, 浏览器对象模型)。此外，还将讨论 ECMA (European Computer Manufacturer's Association, 欧洲计算机制造商协会) 和 W3C (World Wide Web Consortium, 万维网联盟) 制定的一些相关标准。

第 2 章“在 HTML 中使用 JavaScript”，介绍了如何在 HTML 中使用 JavaScript 创建动态网页。这一章不仅展示了在网页中嵌入 JavaScript 的各种方式，还讨论了 JavaScript 内容类型 (content-type) 及其与 <script> 元素的关系。

第 3 章“基本概念”，讨论了 JavaScript 语言的基本概念，包括语法和流控制语句。这一章也分析了 JavaScript 与其他基于 C 的语言在语法上的相同和不同之处，还介绍了与内置操作符有关的类型转换问题。

第 4 章“变量、作用域和内存问题”，探讨了 JavaScript 如何处理其松散类型的变量。这一章还讨论了原始值和引用值之间的差别，以及与变量有关的执行环境的相应内容。最后，通过介绍 JavaScript 的垃圾收集机制，解释了变量在退出作用域时释放其内存的问题。

第 5 章“引用类型”，详尽介绍了 JavaScript 内置的所有引用类型，如 Object 和 Array。这一章对 ECMA-262 规范中描述的每一种引用类型既做了理论上的阐释，又从浏览器实现的角度给出了介绍。

第 6 章“面向对象的程序设计”，讲述了在 JavaScript 中如何实现面向对象的程序设计。由于 JavaScript 没有类的概念，因此这一章从对象创建和继承的层面上展示了一些流行的技术。此外，这

① 本书中文版《JavaScript 入门经典（第 3 版）》已经由清华大学出版社出版。——译者注

② 读者也可以在图灵公司网站 (<http://www.turingbook.com/>) 本书的页面中免费注册下载。——译者注

一章还讲解了函数原型的概念，并对函数原型与整个面向对象方法的关系进行了探讨。

第 7 章“匿名函数”，集中介绍了 JavaScript 中最为强大的一个特性——匿名函数。相关的内容涉及闭包、this 对象的角色、模块模式和创建私有对象成员等。

第 8 章“BOM”，介绍 BOM (Browser Object Model, 浏览器对象模型)，即负责处理与浏览器自身有关的交互操作的对象集合。这一章全面介绍了每一个 BOM 对象，包括 window、document、location、navigator 和 screen。

第 9 章“客户端检测”，讨论了检测客户端机器及其支持特性的各种手段，包括特性检测及用户代理字符串检测的不同技术。这一章还就每种手段的优缺点及适用情形给出了详细说明。

第 10 章“DOM”，介绍 DOM (Document Object Model, 文档对象模型)，即 DOM1 规定的 JavaScript 中的 DOM 对象。这一章也简要介绍了 XML 及其与 DOM 的关系，为深入探讨所有 DOM 规范及其定义的操作网页的方式奠定了基础。

第 11 章“DOM2 和 DOM3”，在上一章的基础上继续探讨了 DOM2 和 DOM3 中新增的 DOM 属性、方法和对象。这一章还讨论了 IE 与其他浏览器的兼容性问题。

第 12 章“事件”，解释了 JavaScript 中事件的本质，对遗留机制的支持，以及 DOM 对事件机制的重新定义。这一章讨论了多种设备，包括 Wii 和 iPhone。

第 13 章“表单脚本”，讲述如何使用 JavaScript 增强表单的交互性，突破浏览器的局限性。这一章的讨论主要围绕单个表单元素如文本框、选择框，以及围绕数据验证和操作展开。

第 14 章“错误处理与调试”，讨论浏览器如何处理 JavaScript 代码错误，并展示了一些处理错误的方式。这一章针对每种浏览器分别讨论了相应的调试工具和技术，还给出了简化调试工作的建议。

第 15 章“JavaScript 与 XML”，展示了 JavaScript 中用于读取和操作 XML (eXtensible Markup Language, 可扩展标记语言) 的特性。这一章分析了不同浏览器提供的 XML 支持和对象的差异，给出了编写跨浏览器代码的简易方法。此外，这一章还介绍了用于在客户端转换 XML 数据的 XSLT (eXtensible Stylesheet Language Transformations, 可扩展样式表语言转换) 技术。

第 16 章“E4X”，讨论了 E4X (ECMAScript for XML, ECMAScript 中的 XML 扩展)；设计 E4X 的出发点是简化 XML 处理任务。这一章探讨了在处理 XML 时，使用 E4X 与使用 DOM 相比有哪些优势。

第 17 章“Ajax 与 JSON”，讲解了常用的 Ajax 技术，包括使用 XMLHttpRequest 对象及使用 IE 的 XDomainRequest 对象实现跨域 Ajax 通信。这一章展示了浏览器在实现与支持 Ajax 方面存在的差异，同时也给出了一些使用建议。

第 18 章“高级技巧”，深入讲解了一些 JavaScript 中较复杂的模式，包括函数科里化 (currying)、部分函数应用和动态函数。这一章还讨论了如何创建自定义的事件框架，以便为自定义对象添加简单的事件处理能力。

第 19 章“客户端存储”，讨论了在客户端机器中存储数据的各种技术。先从受到最广泛支持的特性——cookie 谈起，继而介绍了新兴的客户端存储技术，如 DOM 存储。

第 20 章“最佳实践”，探讨了在企业级环境中使用 JavaScript 的各种方式。其中，着眼于提高可维护性的内容包括编码技巧、格式化和通用编程实践。这一章还介绍了改善代码执行性能及速度优化的一些技术。最后讨论了部署问题，包括如何创建构建过程。

第 21 章“未来的 API”，介绍了为增强浏览器中的 JavaScript 而创建的新 API。虽然这些 API 还没有得到完整或全面的支持，但它们已经崭露头角，有些浏览器也已经部分地实现了这些 API。这一章的内容主要是选择符 API 和 HTML5。

第 22 章“JavaScript 的未来”，展望 JavaScript 的未来，把握 JavaScript 的发展方向。这一章讨论的内容包括 ECMAScript 3.1、ECMAScript 4 和 ECMAScript Harmony。

排版约定

为了让读者更好地理解本书内容，同时把握住全书的重点，本书将采用以下排版约定。

这种带边框的样式表示与上下文相关的重要的、需要牢记的内容。

与上下文相关的说明、提示、技巧、窍门和背景知识以楷体形式印刷。

正文中的样式说明如下。

- (1) 新术语及重要的词汇在首次出现时使用加粗字体以示强调；
- (2) 表示键盘命令组合的方式是 Ctrl+A；
- (3) 正文中的文件名、URL 及代码使用等宽字体，如 `persistence.properties`；
- (4) 代码有两种样式：

```
var obj = new Object(); // 大多数示例代码都没有加背景
var obj = new Object(); // 添加灰色背景的代码表示在上下文中特别重要
```

源代码

在学习本书示例代码时，可以手工敲入所有代码，也可以使用随书的源代码文件。本书所有源代码都可以到 www.wrox.com 中下载^①。登录该站点后，先找到本书（通过搜索或者图书列表^②），打开本书页面后，单击其中的 Download Code 链接，就可以下载本书的源代码了。

由于很多书的名称看起来类似，所以更好的方式是通过本书的 ISBN 来搜索它。本书的 ISBN 是 978-0-470-22780-0。

下载完代码后，请使用解压缩软件将其解压缩。此外，读者也可以登录 Wrox 代码下载主页 www.wrox.com/dynamic/books/download.aspx，查找并下载本书及其他 Wrox 图书的示例代码。

勘误信息^③

我们尽最大努力确保正文和代码没有错误。可是，金无足赤，错误在所难免。如果读者发现我们书中的任何错误，例如错别字或代码片段无法运行等，希望您能及时给我们反馈。您提交的勘误不仅能让其他读者受益，而且也能帮助我们进一步提高图书质量。

本书原版的勘误页面位于 www.wrox.com 中，登录该站点后可以通过搜索或查询图书列表找到本书页面，然后单击页面中的 Errata（勘误）链接。然后可以看到其他读者已经提交并由 Wrox 的编辑发布的勘误信息。另外，在 www.wrox.com/misc-pages/booklist.shtml 页面中也可以找到本书及勘误页面的链接。

① 本书作者提供了在线源代码，地址为 <http://tinyurl.com/pro-js-2nd-examples>。——译者注

② 本书英文版名称请参见本书封面。——译者注

③ 您也可以登录图灵公司网站 (<http://www.turingbook.com/>)，在本书页面中点击“提交勘误”链接，提交您发现的错误。——编者注

如果读者在本书勘误页面中没有发现“你的”错误，麻烦打开 www.wrox.com/contact/techsupport.shtml 页面，填写其中的表单并将错误发送给我们。我们会认真核对您提交的错误，如果错误确实存在，我们将把它补充到本书勘误页面中。同时，也将根据您提供的信息对本书后续版本加以改正。

p2p.wrox.com

如果您想与本书作者或者其他读者沟通，请加入 P2P 论坛 (p2p.wrox.com)。该论坛是基于 Web 的系统，您可以在其中发表与 Wrox 图书及相关技术有关的帖子，并同其他读者或者技术用户交流。论坛提供了一个订阅功能，您可以选择当发表您感兴趣的帖子时通过邮件通知您。Wrox 的作者、编辑、其他行业的专家以及与您正在读同一本书的读者都会出现在这个论坛中。

在 <http://p2p.wrox.com> 中，有很多论坛不仅对您理解本书有帮助，而且还会对开发应用程序有帮助。要加入这个论坛，请按下面几个步骤进行：

- (1) 登录到 p2p.wrox.com，单击 Register (注册) 链接；
- (2) 阅读使用条款并单击 Agree (同意)；
- (3) 完成必填信息和您愿意提供的可选信息，然后单击 Submit (提交)；
- (4) 随后，您会收到一封电子邮件，其中包含如何验证账号和完成注册过程的信息。

如果不加入 P2P 论坛，虽然也可以阅读其中的帖子，但却不能发表帖子，只有注册后才能发表。

在加入论坛后，既可以发表新帖子也可以回复其他用户的帖子。可以在任何时间上网浏览论坛中的帖子。如果希望将某个论坛中的新帖子通过电子邮件发送给您，请在论坛列表中单击与论坛名相关的 Subscribe to this Forum (订阅这个论坛) 图标。

如果想了解有关如何使用 Wrox P2P 的更多信息，请阅读包含论坛规则、P2P 及 Wrox 图书常见问题的 P2P FAQ；要阅读 FAQ，可以在任何 P2P 页面中单击 FAQ 链接。

致 谢

一本书的出版要融入很多人的心血。在此，我想感谢那些为本书做出过贡献的人。

首先，感谢 Wiley 公司所有人的支持：Jim Minatel 又一次给了我机会，Kevin Kent 帮我整理出了本书的写作提纲，Alexei Gorkov 这位世界最好的技术编辑让我做到了滴水不漏。

其次，感谢对本书样章提供反馈的朋友：David Serduke、Julian Turner、Pete Frueh、Chris Klaiber、Stoyan Stefanov、Ross Harmes 和 David Golightly。本书能有今天的规格和品质，离不开上述各位的早期反馈。

最后，感谢 Eric Mirgalia 为本书倾情作序。Eric 给了我加入 Yahoo! 公司的理由，两年来我们之间的合作愉快而又融洽。

目 录

第 1 章 JavaScript 简介	1	3.4.2 Undefined 类型	21
1.1 JavaScript 简史	1	3.4.3 Null 类型	22
1.2 JavaScript 实现	2	3.4.4 Boolean 类型	22
1.2.1 ECMAScript	3	3.4.5 Number 类型	23
1.2.2 文档对象模型 (DOM)	5	3.4.6 String 类型	28
1.2.3 浏览器对象模型 (BOM)	7	3.4.7 Object 类型	30
1.3 JavaScript 版本	8	3.5 操作符	31
1.4 小结	8	3.5.1 一元操作符	31
第 2 章 在 HTML 中使用 JavaScript	9	3.5.2 位操作符	34
2.1 <script>元素	9	3.5.3 布尔操作符	38
2.1.1 标签的位置	11	3.5.4 乘性操作符	40
2.1.2 延迟脚本	11	3.5.5 加性操作符	42
2.1.3 在 XHTML 中的用法	12	3.5.6 关系操作符	43
2.1.4 不推荐使用的语法	13	3.5.7 相等操作符	44
2.1.5 嵌入代码与外部文件	13	3.5.8 条件操作符	46
2.2 文档模式	14	3.5.9 赋值操作符	46
2.3 <noscript>元素	15	3.5.10 逗号操作符	47
2.4 小结	16	3.6 语句	47
第 3 章 基本概念	17	3.6.1 if 语句	47
3.1 语法	17	3.6.2 do-while 语句	48
3.1.1 区分大小写	17	3.6.3 while 语句	48
3.1.2 标识符	17	3.6.4 for 语句	48
3.1.3 注释	18	3.6.5 for-in 语句	49
3.1.4 语句	18	3.6.6 label 语句	50
3.2 关键字和保留字	18	3.6.7 break 和 continue 语句	50
3.3 变量	19	3.6.8 with 语句	52
3.4 数据类型	20	3.6.9 switch 语句	52
3.4.1 typeof 操作符	20	3.7 函数	54
		3.7.1 理解参数	55

3.7.2 没有重载	57	5.5.5 函数属性和方法	97
3.8 小结	57	5.6 基本包装类型	99
第4章 变量、作用域和内存问题	59	5.6.1 Boolean 类型	100
4.1 基本类型和引用类型的值	59	5.6.2 Number 类型	101
4.1.1 动态属性	60	5.6.3 String 类型	102
4.1.2 复制变量值	60	5.7 内置对象	109
4.1.3 传递参数	62	5.7.1 Global 对象	109
4.1.4 检测类型	63	5.7.2 Math 对象	111
4.2 执行环境及作用域	64	5.8 小结	114
4.2.1 延长作用域链	66	第6章 面向对象的程序设计	115
4.2.2 没有块级作用域	66	6.1 创建对象	115
4.3 垃圾收集	69	6.1.1 工厂模式	115
4.3.1 标记清除	69	6.1.2 构造函数模式	116
4.3.2 引用计数	69	6.1.3 原型模式	119
4.3.3 性能问题	70	6.1.4 组合使用构造函数模式和原型模式	128
4.3.4 管理内存	71	6.1.5 动态原型模式	129
4.4 小结	72	6.1.6 寄生构造函数模式	129
第5章 引用类型	73	6.1.7 稳妥构造函数模式	130
5.1 Object 类型	73	6.2 继承	131
5.2 Array 类型	75	6.2.1 原型链	131
5.2.1 转换方法	77	6.2.2 借用构造函数	136
5.2.2 栈方法	78	6.2.3 组合继承	137
5.2.3 队列方法	79	6.2.4 原型式继承	138
5.2.4 重排序方法	80	6.2.5 寄生式继承	139
5.2.5 操作方法	81	6.2.6 寄生组合式继承	139
5.3 Date 类型	83	6.3 小结	142
5.3.1 继承的方法	84	第7章 匿名函数	143
5.3.2 日期格式化方法	85	7.1 递归	144
5.3.3 日期/时间组件方法	85	7.2 闭包	144
5.4 RegExp 类型	86	7.2.1 闭包与变量	147
5.4.1 RegExp 实例属性	88	7.2.2 关于 this 对象	148
5.4.2 RegExp 实例方法	89	7.2.3 内存泄漏	149
5.4.3 RegExp 构造函数属性	90	7.3 模仿块级作用域	150
5.4.4 模式的局限性	92	7.4 私有变量	152
5.5 Function 类型	92	7.4.1 静态私有变量	153
5.5.1 没有重载 (深入理解)	93	7.4.2 模块模式	154
5.5.2 函数声明与函数表达式	94	7.4.3 增强的模块模式	156
5.5.3 作为值的函数	94	7.5 小结	157
5.5.4 函数内部属性	96		

第8章 BOM	158	10.2.1 呈现模式	231
8.1 window 对象	158	10.2.2 滚动	232
8.1.1 全局作用域	158	10.2.3 children 属性	232
8.1.2 窗口关系及框架	159	10.2.4 contains()方法	233
8.1.3 窗口位置	161	10.2.5 操作内容	234
8.1.4 窗口大小	162	10.3 DOM 操作技术	239
8.1.5 导航和打开窗口	163	10.3.1 动态脚本	239
8.1.6 间歇调用和超时调用	166	10.3.2 动态样式	241
8.1.7 系统对话框	168	10.3.3 操作表格	242
8.2 location 对象	169	10.3.4 使用 NodeList	245
8.2.1 查询字符串参数	170	10.4 小结	245
8.2.2 位置操作	171	第11章 DOM2 和 DOM3	247
8.3 navigator 对象	172	11.1 DOM 变化	247
8.3.1 检测插件	173	11.1.1 针对 XML 命名空间的变化	248
8.3.2 注册处理程序	175	11.1.2 其他方面的变化	250
8.4 screen 对象	176	11.2 样式	254
8.5 history 对象	177	11.2.1 访问元素的样式	254
8.6 小结	178	11.2.2 操作样式表	258
第9章 客户端检测	179	11.2.3 元素大小	261
9.1 能力检测	179	11.3 遍历	267
9.2 怪癖检测	181	11.3.1 NodeIterator	268
9.3 用户代理检测	181	11.3.2 TreeWalker	271
9.3.1 用户代理字符串的历史	182	11.4 范围	272
9.3.2 用户代理字符串检测技术	187	11.4.1 DOM 中的范围	272
9.3.3 完整的代码	199	11.4.2 IE 中的范围	279
9.3.4 使用方法	202	11.5 小结	282
9.4 小结	202	第12章 事件	283
第10章 DOM	204	12.1 事件流	283
10.1 节点层次	204	12.1.1 事件冒泡	283
10.1.1 Node 类型	205	12.1.2 事件捕获	284
10.1.2 Document 类型	210	12.1.3 DOM 事件流	285
10.1.3 Element 类型	217	12.2 事件处理程序(或事件侦听器)	285
10.1.4 Text 类型	225	12.2.1 HTML 事件处理程序	286
10.1.5 Comment 类型	228	12.2.2 DOM0 级事件处理程序	286
10.1.6 CDATASection 类型	228	12.2.3 DOM2 级事件处理程序	287
10.1.7 DocumentType 类型	229	12.2.4 IE 事件处理程序	288
10.1.8 DocumentFragment 类型	229	12.2.5 跨浏览器的事件处理程序	289
10.1.9 Attr 类型	230	12.3 事件对象	290
10.2 DOM 扩展	231	12.3.1 DOM 中的事件对象	291

12.3.2	IE 中的事件对象	293	14.1.1	Internet Explorer	363
12.3.3	跨浏览器的事件对象	295	14.1.2	Firefox	364
12.4	事件类型	296	14.1.3	Safari	366
12.4.1	UI 事件	297	14.1.4	Opera	367
12.4.2	鼠标事件	297	14.1.5	Chrome	368
12.4.3	键盘事件	303	14.2	错误处理	369
12.4.4	HTML 事件	307	14.2.1	try-catch 语句	369
12.4.5	变动事件	312	14.2.2	抛出错误	372
12.4.6	专有事件	316	14.2.3	错误 (error) 事件	374
12.4.7	移动 Safari 支持的事件	325	14.3	错误处理策略	375
12.5	内存和性能	328	14.3.1	常见的错误类型	375
12.5.1	事件委托	328	14.3.2	区分致命错误和非致命错误	379
12.5.2	移除事件处理程序	330	14.3.3	把错误记录到服务器	380
12.6	模拟事件	331	14.4	调试技术	381
12.6.1	DOM 中的事件模拟	331	14.4.1	将消息记录到控制台	381
12.6.2	IE 中的事件模拟	335	14.4.2	将消息记录到当前页面	383
12.7	小结	336	14.4.3	抛出错误	384
第 13 章	表单脚本	337	14.5	常用的 IE 错误	384
13.1	表单	337	14.5.1	操作终止	384
13.1.1	提交表单	338	14.5.2	无效字符	386
13.1.2	重置表单	338	14.5.3	未找到成员	386
13.1.3	表单字段	339	14.5.4	未知运行时错误	387
13.2	文本框脚本	343	14.5.5	语法错误	387
13.2.1	选择文本	344	14.5.6	系统无法找到指定资源	387
13.2.2	过滤输入	347	14.6	调试工具	388
13.2.3	自动切换焦点	350	14.6.1	IE 中的调试器	388
13.3	选择框脚本	351	14.6.2	Firebug	392
13.3.1	选择选项	352	14.6.3	Drosera	396
13.3.2	添加选项	354	14.6.4	Opera 中的 JavaScript 调试器	398
13.3.3	移除选项	354	14.6.5	其他调试工具	401
13.3.4	移动和重排选项	355	14.7	小结	401
13.4	表单序列化	355	第 15 章	JavaScript 与 XML	402
13.5	富文本编辑	357	15.1	浏览器对 XML DOM 的支持	402
13.5.1	操作富文本	358	15.1.1	DOM2 级核心	402
13.5.2	富文本选区	359	15.1.2	DOMParser 类型	403
13.5.3	表单与富文本	361	15.1.3	XMLSerializer 类型	404
13.6	小结	361	15.1.4	DOM3 级加载和保存	404
第 14 章	错误处理与调试	363	15.1.5	IE 对 XML 的支持	409
14.1	浏览器错误报告	363			