

TURING

图灵程序设计

由 Foxit PDF Editor 编辑

版权所有 (c) by Foxit Software Company, 2003 - 2009

仅用于评估。



Professional JavaScript for Web Developers 3rd Edition

JavaScript 高级程序设计 (第3版)

[美] Nicholas C. Zakas 著
李松峰 曹力 译



人民邮电出版社
POSTS & TELECOM PRESS

1995年, Brendan Eich创造了JavaScript。

2005年, 席卷全球的“Ajax热”激发了全世界Web开发人员学习JavaScript的热情。与此同时, 本书第1版诞生。这一版的中文版狂销4万册, 被誉为“最具深度的JavaScript经典”, 奠定了其不可替代的权威地位。

2005年到2009年, 前端开发社区在实践中充分检验了这门语言的各种实现和扩展, JavaScript从被戏谑的“玩具语言”一跃成为软件业举足轻重的通用编程语言。2009年1月本书第2版应运而生, 凝聚作者和社区专家多年宝贵经验的这一技术名著再次得到读者认可和褒扬, 中文版销量达到2万册。

2009年到2011年, ECMAScript 5和HTML5在标准之争中双双胜出, 使大量专有实现和客户端扩展正式进入规范, 同时也为这门语言增添了很多适应未来发展的新特性。2012年初本书第3版面世, 中文版也紧随其后。第3版除增加5章全新内容外, 其他章节也有较大幅度的增补和修订, 新内容篇幅约占三分之一。

作为JavaScript技术经典名著, 本书承继了之前版本全面深入、贴近实战的特点, 在详细讲解了JavaScript语言的核心之后, 条分缕析地为读者展示了现有规范及实现为开发Web应用提供的各种支持和特性。

本书主要包括:

- ◆ 对JavaScript实现各个组成部分的详尽解读;
- ◆ 对JavaScript面向对象编程的全方位阐述;
- ◆ 对DOM、BOM及浏览器事件模型的透彻剖析;
- ◆ Web应用基本数据格式JSON、XML及其存取;
- ◆ Ajax、Comet服务器端通信和基于File API的拖放式文件上传;
- ◆ ECMAScript 5定义的最新核心语言特性;
- ◆ HTML 5涵盖的表单、媒体、Canvas (包括WebGL);
- ◆ Selectors、Web Workers、地理定位及跨文档传递消息等最新API;
- ◆ 离线应用及客户端存储 (包括IndexedDB);
- ◆ 维护、性能、部署相关的最佳开发实践;
- ◆ 新兴API及ECMAScript Harmony展望。

本书适合有一定编程经验的Web应用开发人员阅读, 也可作为高校及社会实用技术培训相关专业课程的教材。



WILEY
www.wiley.com

图灵社区: www.ituring.com.cn
反馈/投稿/推荐信箱: contact@turingbook.com
热线: (010)51095186转604

分类建议 计算机/程序设计

人民邮电出版社网址: www.ptpress.com.cn



ISBN 978-7-115-27579-0



9 787115 275790 >

ISBN 978-7-115-27579-0

定价: 99.00元

由 Foxit PDF Editor 编辑
版权所有 (c) by Foxit Software Company, 2003 - 2009
仅用于评估。

TURING

图灵社区设计丛书



Professional JavaScript for Web Developers 3rd Edition

JavaScript 高级程序设计 (第3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 (C I P) 数据

JavaScript 高级程序设计 : 第3版 / (美) 泽卡斯
(Zakas, N.C.) 著 ; 李松峰, 曹力译. -- 北京 : 人民邮
电出版社, 2012.3 (2012.5重印)

(图灵程序设计丛书)

书名原文: Professional JavaScript for Web
Developers, 3rd Edition
ISBN 978-7-115-27579-0

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(2012)第027879号

内 容 提 要

本书是 JavaScript 超级畅销书的最新版。ECMAScript 5 和 HTML5 在标准之争中双双胜出,使大量专有实现和客户端扩展正式进入规范,同时也为 JavaScript 增添了很多适应未来发展的新特性。本书这一版除增加 5 章全新内容外,其他章节也有较大幅度的增补和修订,新内容篇幅约占三分之一。全书从 JavaScript 语言实现的各个组成部分——语言核心、DOM、BOM、事件模型讲起,深入浅出地探讨了面向对象编程、Ajax 与 Comet 服务器端通信,HTML5 表单、媒体、Canvas (包括 WebGL) 及 Web Workers、地理定位、跨文档传递消息、客户端存储 (包括 IndexedDB) 等新 API,还介绍了离线应用和与维护、性能、部署相关的最佳开发实践。本书附录展望了未来的 API 和 ECMAScript Harmony 规范。

本书适合有一定编程经验的 Web 应用开发人员阅读,也可作为高校及社会实用技术培训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3版)

-
- ◆ 著 [美] Nicholas C. Zakas
 - 译 李松峰 曹 力
 - 责任编辑 朱 巍
 - ◆ 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 电子邮件 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: 46.75
字数: 1092千字 2012年3月第1版
印数: 6001~12000册 2012年5月北京第2次印刷

著作权合同登记号 图字: 01-2012-1175 号

ISBN 978-7-115-27579-0

定价: 99.00元

读者服务热线: (010)51095186转604 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 3rd Edition*, by Nicholas C. Zakas, ISBN 978-1-118-02669-4, published by John Wiley & Sons, Inc.

Copyright ©2012 by John Wiley & Sons, Inc., All rights reserved. This translation published under License.

Translation edition published by POSTS & TELECOM PRESS Copyright ©2012.

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。

序

20 多年的职业生涯，我也长出了白发。回首往事，曾经对我的职业道路产生过重要影响的技术和人历历在目。如果让我只说一种技术，一种对我产生了最大正面影响的技术，那么就是 JavaScript。说实话，我也并非一直都是 JavaScript 的信徒。跟许多人一样，我以前也把它当作一门玩具语言，认为它只能用来做一些旋转的横幅广告，或者在页面中添加一些有意思的交互效果作为装饰。我原来是做服务器端开发的，我们都对这种玩具语言不感冒，该死的！可是，后来 Ajax 出现了。

永远也忘不了当时无孔不入的 Ajax，大家都认为它是一种非常酷、非常新，同时极具创造性的技术。我也开始了解它，阅读相关资料。知道这门曾被嗤之以鼻的玩具语言如今被每一位专业 Web 开发人员津津乐道之后，我感到很震惊。突然，我的看法就转变了。随着探索 Ajax 的继续深入，我认识到 JavaScript 的强大威力，急切地想了解它能提供的所有“法宝”。于是，我全身心地投入到学习 JavaScript 之中，不仅努力学习这门语言，还加入了 jQuery 项目团队，专门从事客户端开发。我的日子过得很爽。

对 JavaScript 了解得越深，接触的开发人员就越多，其中不乏今天在我眼里依然是巨星和导师级的人物。尼古拉斯·泽卡斯（本书作者）就是这样一位开发人员。我一直记得在读书第 2 版时心中油然而生的喜悦之情，虽然我也有多年的积累，但仍然从中学到了很多新东西。这本书实实在在、深入浅出，读来就好像尼古拉斯对不同层次的读者都了如指掌，所以他的风格才那么贴切自然。对于技术书来说，这是非常突出的一个特色。多数作者都想靠坚深的技术给人留下印象，但这本书不同。所以，它很快就成为了我案头必备的书，我也会向那些有志全面掌握 JavaScript 的开发人员推荐这本书。我希望每个人对这本书都能有跟我一样的体会，认识到它的价值所在。

后来，在一次 jQuery 大会上，我荣幸地见到了尼古拉斯本人。站在我面前的是一位世界顶级的 JavaScript 开发人员，而且正负责世界上最重要的一个 Web 站点（雅虎）。尼古拉斯是我见过的最随和的人之一。真的，见到他的时候我有一种追星族的幻觉。但他就是那么一个活生生的人，一个想帮助开发人员成就梦想的人。不仅他的书改变了我对 JavaScript 的认识，而且尼古拉斯这个人，也让我愿意接近，愿意了解。

听说尼古拉斯要请我作序，我激动得不知道说什么才好。在此，我代表大牛来为本书暖场。这个序也是他本人有多么令人景仰的一个明证。不过，更重要的是，这也给了我一个机会，让我能跟大家分享自己为什么觉得这本书如此重要。我看过很多 JavaScript 图书，的确也有很多令人叹服的佳作。但在我看来，这本书为读者成为全方位的 JavaScript 高手提供了“一揽子方案”。

这本书从介绍表达式和变量声明开始，平滑地过渡到了闭包、面向对象开发等高级主题。与那些把大量篇幅花在讲解背景知识上的书，以及那些让人感觉好像是要使用 JavaScript 开发导弹制导系统的书相比，这本书让人感觉细致周到、亲切自然。这是一本写给“普通人”的书，它能让你编写出引以为荣的代码，构建出令人叫绝的网站。

雷·邦戈（Rey Bango）

微软公司高级布道师，jQuery 项目团队核心成员

献给我的父母，是他们永远给我支持和鼓励。

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儿都不为过。

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。

本书读者对象

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。

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（以下脚注如无特殊说明，均为译者注）

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。本书还探讨了 HTML5、Selectors API 和 File API 等一系列新 API。

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。

本书结构

本书共 25 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“函数表达式”，集中介绍了 JavaScript 中最为强大的一个特性——函数表达式。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“DOM 扩展”，介绍了其他 API 以及浏览器本身为 DOM 添加的各种功能。涉及内容包括 Selectors API、Element Traversal API 和 HTML5 扩展。

第 12 章“DOM2 和 DOM3”，在前两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 13 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 14 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 15 章“使用 Canvas 绘图”，讨论了<canvas>标签以及如何通过它来动态绘图。不仅涵盖 2D 上下文，也将讨论 WebGL (3D) 上下文，可以为创建动画和游戏夯实基础。

第 16 章“HTML5 脚本编程”，介绍了 HTML5 规定的 JavaScript API，涉及跨文档传递消息、拖放 API 和以编程方式控制<audio>和<video>元素，以及管理历史状态。

第 17 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 18 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 19 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 20 章“JSON”，介绍了作为 XML 替代格式的 JSON，包含浏览器原生支持的 JSON 解析和序列化，以及使用 JSON 时要注意的安全问题。

第 21 章“Ajax 与 Comet”，讲解了常用的 Ajax 技术，包括使用 XMLHttpRequest 对象及 CORS (Cross-Origin Resource Sharing, 跨来源资源共享) API 实现跨域 Ajax 通信。这一章展示了浏览器在实现与支持方面存在的差异，同时也给出了一些使用建议。

第 22 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和使用 ECMAScript 5 创建防篡改对象。

第 23 章“离线应用与客户端存储”，讨论了如何检测应用离线以及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 Web Storage 和 IndexedDB。

第 24 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 25 章“新兴的 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 Web 计时和文件 API。

使用示例

要运行本书中的示例，需要安装下列软件：

- Windows XP、Windows 7 或 Mac OS X；
- Internet Explorer 6 及更高版本、Firefox 2 及更高版本、Opera 9 及更高的版本、Chrome、Safari 2 及更高版本。

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）^①。

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。



这种带警告图标的方框样式，表示与上下文相关的重要的、需要牢记的内容。



这种带钢笔图标的方框样式，表示与上下文相关的说明、提示、技巧、窍门和背景知识。

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粗  
var obj = new Object(); // 加粗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。登录该站点后，先找到本书（通过搜索或者图书列表），打开本书页

^① 读者也可以在图灵社区（<http://www.it-ebooks.com/>）本书的页面中免费注册下载。

面后，单击其中的 Download Code 链接，就可以下载本书的源代码了^①。对于包含在下载文件中的源代码，书中会添加以下图标：



本书代码示例旁边会附有文件名，从中可以找到对应的代码片段。文件名的格式如下：

代码片段所在的文件名



由于很多书的书名看起来类似，所以更好的方式是通过书的 ISBN 来搜索它。本书原版的 ISBN 是 978-1-118-02669-4。

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^②

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。

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；

^① 翻译本书时，[wrox.com](http://www.wrox.com) 中下载本书代码的短地址为：<http://tinyurl.com/projs-3rd-code>。

^② 您也可以登录图灵社区（<http://www.it-ebooks.cn/>），在本书页面中提交您发现的错误。

- (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

虽然作者的名字被印在了封面上, 但一个人是不可能完成这本书的, 我想感谢与出版本书有关的一些人。

首先, 感谢 John Wiley & Sons 继续给我写作的机会。当时, 出版本书第 1 版时, 他们是唯一愿意承担风险的一家出版社。对此, 我将永远铭记于心。

感谢 John Wiley & Sons 的编辑人员, 特别是 Kevin Kent 和 John Peloquin, 他们卓有成效的工作使我保持了坦诚直率的风格, 也解决了我在写作期间不断变更内容的问题。

还要感谢对本书草稿给出反馈意见的所有人: Rob Friesel、Sergey Ilinsky、Dan Kielp、Peter-Paul Koch、Jeremy McPeak、Alex Petrescu、Dmitry Soshnikov 和 Juriy “Kangax” Zaytsev。你们的宝贵意见让我自己都为本书感到骄傲。

我想特别感谢 Brendan Eich, 感谢他纠正了第 1 章中有关 JavaScript 历史的细节问题。

最后, 当然也是非常重要的, 感谢 Rey Bango 为本书作序。很高兴在与 Rey 通过网络认识几年之后, 终于在 2010 年有缘相见。他是这个行业里真正出色的人, 我非常荣幸能请到他为本书作序。

关于技术编辑

John Peloquin 是一位有十多年 JavaScript 经验的前端工程师, 开发过各种规模的应用。John 拥有加州大学伯克利分校的数学学士学位, 目前在一家致力于卫生保健的创业公司担任开发主管。在编辑本书之前, John 编辑过 Jeremy McPeak 的 *JavaScript 24-Hour Trainer* (Wiley, 2010)。编写代码和收集勘误之余, John 经常沉迷于数学、哲学和魔术。

目 录

| | | | |
|-----------------------------------|----|---------------------------------|----|
| 第 1 章 JavaScript 简介 | 1 | 3.4.2 Undefined 类型 | 24 |
| 1.1 JavaScript 简史 | 1 | 3.4.3 Null 类型 | 25 |
| 1.2 JavaScript 实现 | 2 | 3.4.4 Boolean 类型 | 26 |
| 1.2.1 ECMAScript | 3 | 3.4.5 Number 类型 | 27 |
| 1.2.2 文档对象模型 (DOM) | 5 | 3.4.6 String 类型 | 32 |
| 1.2.3 浏览器对象模型 (BOM) | 8 | 3.4.7 Object 类型 | 35 |
| 1.3 JavaScript 版本 | 8 | 3.5 操作符 | 36 |
| 1.4 小结 | 9 | 3.5.1 一元操作符 | 36 |
| 第 2 章 在 HTML 中使用 JavaScript | 10 | 3.5.2 位操作符 | 39 |
| 2.1 <script>元素 | 10 | 3.5.3 布尔操作符 | 44 |
| 2.1.1 标签的位置 | 12 | 3.5.4 乘性操作符 | 47 |
| 2.1.2 延迟脚本 | 13 | 3.5.5 加性操作符 | 48 |
| 2.1.3 异步脚本 | 13 | 3.5.6 关系操作符 | 50 |
| 2.1.4 在 XHTML 中的用法 | 14 | 3.5.7 相等操作符 | 51 |
| 2.1.5 不推荐使用的语法 | 16 | 3.5.8 条件操作符 | 53 |
| 2.2 嵌入代码与外部文件 | 16 | 3.5.9 赋值操作符 | 53 |
| 2.3 文档模式 | 16 | 3.5.10 逗号操作符 | 54 |
| 2.4 <noscript>元素 | 18 | 3.6 语句 | 54 |
| 2.5 小结 | 18 | 3.6.1 if 语句 | 54 |
| 第 3 章 基本概念 | 19 | 3.6.2 do-while 语句 | 55 |
| 3.1 语法 | 19 | 3.6.3 while 语句 | 55 |
| 3.1.1 区分大小写 | 19 | 3.6.4 for 语句 | 56 |
| 3.1.2 标识符 | 19 | 3.6.5 for-in 语句 | 57 |
| 3.1.3 注释 | 20 | 3.6.6 label 语句 | 58 |
| 3.1.4 严格模式 | 20 | 3.6.7 break 和 continue 语句 | 58 |
| 3.1.5 语句 | 20 | 3.6.8 with 语句 | 60 |
| 3.2 关键字和保留字 | 21 | 3.6.9 switch 语句 | 60 |
| 3.3 变量 | 22 | 3.7 函数 | 62 |
| 3.4 数据类型 | 23 | 3.7.1 理解参数 | 64 |
| 3.4.1 typeof 操作符 | 23 | 3.7.2 没有重载 | 66 |
| | | 3.8 小结 | 67 |

| | | | |
|---------------------|-----|-----------------------|-----|
| 第 4 章 变量、作用域和内存问题 | 68 | 5.5.3 作为值的函数 | 112 |
| 4.1 基本类型和引用类型的值 | 68 | 5.5.4 函数内部属性 | 113 |
| 4.1.1 动态的属性 | 68 | 5.5.5 函数属性和方法 | 116 |
| 4.1.2 复制变量值 | 69 | 5.6 基本包装类型 | 118 |
| 4.1.3 传递参数 | 70 | 5.6.1 Boolean 类型 | 120 |
| 4.1.4 检测类型 | 72 | 5.6.2 Number 类型 | 120 |
| 4.2 执行环境及作用域 | 73 | 5.6.3 String 类型 | 122 |
| 4.2.1 延长作用域链 | 75 | 5.7 单体内置对象 | 130 |
| 4.2.2 没有块级作用域 | 76 | 5.7.1 Global 对象 | 131 |
| 4.3 垃圾收集 | 78 | 5.7.2 Math 对象 | 134 |
| 4.3.1 标记清除 | 78 | 5.8 小结 | 137 |
| 4.3.2 引用计数 | 79 | 第 6 章 面向对象的程序设计 | 138 |
| 4.3.3 性能问题 | 80 | 6.1 理解对象 | 138 |
| 4.3.4 管理内存 | 81 | 6.1.1 属性类型 | 139 |
| 4.4 小结 | 81 | 6.1.2 定义多个属性 | 142 |
| 第 5 章 引用类型 | 83 | 6.1.3 读取属性的特性 | 143 |
| 5.1 Object 类型 | 83 | 6.2 创建对象 | 144 |
| 5.2 Array 类型 | 86 | 6.2.1 工厂模式 | 144 |
| 5.2.1 检测数组 | 88 | 6.2.2 构造函数模式 | 144 |
| 5.2.2 转换方法 | 89 | 6.2.3 原型模式 | 147 |
| 5.2.3 栈方法 | 90 | 6.2.4 组合使用构造函数模式和原型模式 | 159 |
| 5.2.4 队列方法 | 91 | 6.2.5 动态原型模式 | 159 |
| 5.2.5 重排序方法 | 92 | 6.2.6 寄生构造函数模式 | 160 |
| 5.2.6 操作方法 | 94 | 6.2.7 稳妥构造函数模式 | 161 |
| 5.2.7 位置方法 | 95 | 6.3 继承 | 162 |
| 5.2.8 迭代方法 | 96 | 6.3.1 原型链 | 162 |
| 5.2.9 缩小方法 | 97 | 6.3.2 借用构造函数 | 167 |
| 5.3 Date 类型 | 98 | 6.3.3 组合继承 | 168 |
| 5.3.1 继承的方法 | 100 | 6.3.4 原型式继承 | 169 |
| 5.3.2 日期格式化方法 | 101 | 6.3.5 寄生式继承 | 171 |
| 5.3.3 日期/时间组件方法 | 102 | 6.3.6 寄生组合式继承 | 172 |
| 5.4 RegExp 类型 | 103 | 6.4 小结 | 174 |
| 5.4.1 RegExp 实例属性 | 105 | 第 7 章 函数表达式 | 175 |
| 5.4.2 RegExp 实例方法 | 106 | 7.1 递归 | 177 |
| 5.4.3 RegExp 构造函数属性 | 107 | 7.2 闭包 | 178 |
| 5.4.4 模式的局限性 | 109 | 7.2.1 闭包与变量 | 181 |
| 5.5 Function 类型 | 110 | 7.2.2 关于 this 对象 | 182 |
| 5.5.1 没有重载 (深入理解) | 111 | 7.2.3 内存泄漏 | 183 |
| 5.5.2 函数声明与函数表达式 | 111 | | |

| | | | |
|---------------------------|-----|---------------------------------------|-----|
| 7.3 模仿块级作用域 | 184 | 10.1.1 Node 类型 | 248 |
| 7.4 私有变量 | 186 | 10.1.2 Document 类型 | 253 |
| 7.4.1 静态私有变量 | 188 | 10.1.3 Element 类型 | 261 |
| 7.4.2 模块模式 | 189 | 10.1.4 Text 类型 | 270 |
| 7.4.3 增强的模块模式 | 191 | 10.1.5 Comment 类型 | 273 |
| 7.5 小结 | 192 | 10.1.6 CDATASection 类型 | 274 |
| 第 8 章 BOM | 193 | 10.1.7 DocumentType 类型 | 274 |
| 8.1 window 对象 | 193 | 10.1.8 DocumentFragment 类型 | 275 |
| 8.1.1 全局作用域 | 193 | 10.1.9 Attr 类型 | 276 |
| 8.1.2 窗口关系及框架 | 194 | 10.2 DOM 操作技术 | 277 |
| 8.1.3 窗口位置 | 197 | 10.2.1 动态脚本 | 277 |
| 8.1.4 窗口大小 | 198 | 10.2.2 动态样式 | 279 |
| 8.1.5 导航和打开窗口 | 199 | 10.2.3 操作表格 | 281 |
| 8.1.6 间歇调用和超时调用 | 203 | 10.2.4 使用 NodeList | 283 |
| 8.1.7 系统对话框 | 205 | 10.3 小结 | 284 |
| 8.2 location 对象 | 207 | 第 11 章 DOM 扩展 | 286 |
| 8.2.1 查询字符串参数 | 207 | 11.1 选择符 API | 286 |
| 8.2.2 位置操作 | 208 | 11.1.1 querySelector() 方法 | 286 |
| 8.3 navigator 对象 | 210 | 11.1.2 querySelectorAll() 方法 | 287 |
| 8.3.1 检测插件 | 211 | 11.1.3 matchesSelector() 方法 | 288 |
| 8.3.2 注册处理程序 | 213 | 11.2 元素遍历 | 288 |
| 8.4 screen 对象 | 214 | 11.3 HTML5 | 289 |
| 8.5 history 对象 | 215 | 11.3.1 与类相关的扩充 | 289 |
| 8.6 小结 | 216 | 11.3.2 焦点管理 | 291 |
| 第 9 章 客户端检测 | 217 | 11.3.3 HTMLDocument 的变化 | 292 |
| 9.1 能力检测 | 217 | 11.3.4 字符集属性 | 293 |
| 9.1.1 更可靠的能力检测 | 218 | 11.3.5 自定义数据属性 | 293 |
| 9.1.2 能力检测, 不是浏览器检测 | 220 | 11.3.6 插入标记 | 294 |
| 9.2 怪癖检测 | 220 | 11.3.7 scrollIntoView() 方法 | 298 |
| 9.3 用户代理检测 | 221 | 11.4 专有扩展 | 298 |
| 9.3.1 用户代理字符串的历史 | 222 | 11.4.1 文档模式 | 298 |
| 9.3.2 用户代理字符串检测技术 | 228 | 11.4.2 children 属性 | 299 |
| 9.3.3 完整的代码 | 242 | 11.4.3 contains() 方法 | 300 |
| 9.3.4 使用方法 | 245 | 11.4.4 插入文本 | 301 |
| 9.4 小结 | 246 | 11.4.5 滚动 | 303 |
| 第 10 章 DOM | 247 | 11.5 小结 | 304 |
| 10.1 节点层次 | 247 | | |

| | | | |
|-----------------------------|-----|---------------------------------------|-----|
| 第 12 章 DOM2 和 DOM3 | 305 | 13.4.9 触摸与手势事件 | 399 |
| 12.1 DOM 变化 | 305 | 13.5 内存和性能 | 402 |
| 12.1.1 针对 XML 命名空间的变化 | 306 | 13.5.1 事件委托 | 402 |
| 12.1.2 其他方面的变化 | 309 | 13.5.2 移除事件处理程序 | 404 |
| 12.2 样式 | 312 | 13.6 模拟事件 | 405 |
| 12.2.1 访问元素的样式 | 313 | 13.6.1 DOM 中的事件模拟 | 405 |
| 12.2.2 操作样式表 | 317 | 13.6.2 IE 中的事件模拟 | 410 |
| 12.2.3 元素大小 | 320 | 13.7 小结 | 411 |
| 12.3 遍历 | 326 | 第 14 章 表单脚本 | 412 |
| 12.3.1 NodeIterator | 328 | 14.1 表单的基础知识 | 412 |
| 12.3.2 TreeWalker | 330 | 14.1.1 提交表单 | 413 |
| 12.4 范围 | 332 | 14.1.2 重置表单 | 414 |
| 12.4.1 DOM 中的范围 | 332 | 14.1.3 表单字段 | 414 |
| 12.4.2 IE8 及更早版本中的范围 | 340 | 14.2 文本框脚本 | 419 |
| 12.5 小结 | 343 | 14.2.1 选择文本 | 420 |
| 第 13 章 事件 | 345 | 14.2.2 过滤输入 | 423 |
| 13.1 事件流 | 345 | 14.2.3 自动切换焦点 | 426 |
| 13.1.1 事件冒泡 | 346 | 14.2.4 HTML5 约束验证 API | 427 |
| 13.1.2 事件捕获 | 346 | 14.3 选择框脚本 | 431 |
| 13.1.3 DOM 事件流 | 347 | 14.3.1 选择选项 | 432 |
| 13.2 事件处理程序 | 348 | 14.3.2 添加选项 | 434 |
| 13.2.1 HTML 事件处理程序 | 348 | 14.3.3 移除选项 | 435 |
| 13.2.2 DOM0 级事件处理程序 | 350 | 14.3.4 移动和重排选项 | 435 |
| 13.2.3 DOM2 级事件处理程序 | 351 | 14.4 表单序列化 | 436 |
| 13.2.4 IE 事件处理程序 | 352 | 14.5 富文本编辑 | 438 |
| 13.2.5 跨浏览器的事件处理程序 | 353 | 14.5.1 使用 contentEditable 属性 | 438 |
| 13.3 事件对象 | 355 | 14.5.2 操作富文本 | 439 |
| 13.3.1 DOM 中的事件对象 | 355 | 14.5.3 富文本选区 | 441 |
| 13.3.2 IE 中的事件对象 | 358 | 14.5.4 表单与富文本 | 443 |
| 13.3.3 跨浏览器的事件对象 | 360 | 14.6 小结 | 443 |
| 13.4 事件类型 | 362 | 第 15 章 使用 Canvas 绘图 | 445 |
| 13.4.1 UI 事件 | 362 | 15.1 基本用法 | 445 |
| 13.4.2 焦点事件 | 367 | 15.2 2D 上下文 | 446 |
| 13.4.3 鼠标与滚轮事件 | 368 | 15.2.1 填充和描边 | 446 |
| 13.4.4 键盘与文本事件 | 379 | 15.2.2 绘制矩形 | 447 |
| 13.4.5 复合事件 | 384 | 15.2.3 绘制路径 | 449 |
| 13.4.6 变动事件 | 385 | 15.2.4 绘制文本 | 451 |
| 13.4.7 HTML5 事件 | 388 | 15.2.5 变换 | 453 |
| 13.4.8 设备事件 | 395 | | |

| | | | |
|--|-----|-------------------------------|-----|
| 15.2.6 绘制图像 | 456 | 17.2.3 错误 (error) 事件 | 505 |
| 15.2.7 阴影 | 457 | 17.2.4 处理错误的策略 | 506 |
| 15.2.8 渐变 | 458 | 17.2.5 常见的错误类型 | 507 |
| 15.2.9 模式 | 460 | 17.2.6 区分致命错误和非致命 错误 | 510 |
| 15.2.10 使用图像数据 | 460 | 17.2.7 把错误记录到服务器 | 511 |
| 15.2.11 合成 | 462 | 17.3 调试技术 | 512 |
| 15.3 WebGL | 463 | 17.3.1 将消息记录到控制台 | 512 |
| 15.3.1 类型化数组 | 463 | 17.3.2 将消息记录到当前页面 | 515 |
| 15.3.2 WebGL 上下文 | 468 | 17.3.3 抛出错误 | 515 |
| 15.3.3 支持 | 478 | 17.4 常见的 IE 错误 | 516 |
| 15.4 小结 | 478 | 17.4.1 操作终止 | 516 |
| 第 16 章 HTML5 脚本编程 | 480 | 17.4.2 无效字符 | 518 |
| 16.1 跨文档消息传递 | 480 | 17.4.3 未找到成员 | 518 |
| 16.2 原生拖放 | 481 | 17.4.4 未知运行时错误 | 519 |
| 16.2.1 拖放事件 | 482 | 17.4.5 语法错误 | 519 |
| 16.2.2 自定义放置目标 | 482 | 17.4.6 系统无法找到指定资源 | 519 |
| 16.2.3 dataTransfer 对象 | 483 | 17.5 小结 | 520 |
| 16.2.4 dropEffect 与 effectAllowed | 484 | 第 18 章 JavaScript 与 XML | 521 |
| 16.2.5 可拖动 | 485 | 18.1 浏览器对 XML DOM 的支持 | 521 |
| 16.2.6 其他成员 | 485 | 18.1.1 DOM2 级核心 | 521 |
| 16.3 媒体元素 | 486 | 18.1.2 DOMParser 类型 | 522 |
| 16.3.1 属性 | 487 | 18.1.3 XMLSerializer 类型 | 523 |
| 16.3.2 事件 | 488 | 18.1.4 IE8 及之前版本中的 XML | 523 |
| 16.3.3 自定义媒体播放器 | 488 | 18.1.5 跨浏览器处理 XML | 527 |
| 16.3.4 检测编解码器的支持情况 | 489 | 18.2 浏览器对 XPath 的支持 | 529 |
| 16.3.5 Audio 类型 | 490 | 18.2.1 DOM3 级 XPath | 529 |
| 16.4 历史状态管理 | 491 | 18.2.2 IE 中的 XPath | 534 |
| 16.5 小结 | 492 | 18.2.3 跨浏览器使用 XPath | 535 |
| 第 17 章 错误处理与调试 | 493 | 18.3 浏览器对 XSLT 的支持 | 537 |
| 17.1 浏览器报告的错误 | 493 | 18.3.1 IE 中的 XSLT | 537 |
| 17.1.1 IE | 493 | 18.3.2 XSLTProcessor 类型 | 541 |
| 17.1.2 Firefox | 494 | 18.3.3 跨浏览器使用 XSLT | 543 |
| 17.1.3 Safari | 496 | 18.4 小结 | 544 |
| 17.1.4 Opera | 497 | 第 19 章 E4X | 546 |
| 17.1.5 Chrome | 498 | 19.1 E4X 的类型 | 546 |
| 17.2 错误处理 | 499 | 19.1.1 XML 类型 | 546 |
| 17.2.1 try-catch 语句 | 500 | 19.1.2 XMLList 类型 | 547 |
| 17.2.2 抛出错误 | 503 | 19.1.3 Namespace 类型 | 548 |

6 目 录

| | | | | | |
|--------|--------------------------|-----|--------|----------------------|-----|
| 19.1.4 | QName 类型 | 549 | 21.4.3 | Preflighted Requests | 584 |
| 19.2 | 一般用法 | 550 | 21.4.4 | 带凭据的请求 | 585 |
| 19.2.1 | 访问特性 | 551 | 21.4.5 | 跨浏览器的 CORS | 585 |
| 19.2.2 | 其他节点类型 | 552 | 21.5 | 其他跨域技术 | 586 |
| 19.2.3 | 查询 | 553 | 21.5.1 | 图像 Ping | 586 |
| 19.2.4 | 构建和操作 XML | 555 | 21.5.2 | JSONP | 587 |
| 19.2.5 | 解析和序列化 | 557 | 21.5.3 | Comet | 588 |
| 19.2.6 | 命名空间 | 558 | 21.5.4 | 服务器发送事件 | 590 |
| 19.3 | 其他变化 | 559 | 21.5.5 | Web Sockets | 591 |
| 19.4 | 全面启用 E4X | 560 | 21.5.6 | SSE 与 Web Sockets | 593 |
| 19.5 | 小结 | 561 | 21.6 | 安全 | 593 |
| 第 20 章 | JSON | 562 | 21.7 | 小结 | 594 |
| 20.1 | 语法 | 562 | 第 22 章 | 高级技巧 | 596 |
| 20.1.1 | 简单值 | 562 | 22.1 | 高级函数 | 596 |
| 20.1.2 | 对象 | 563 | 22.1.1 | 安全的类型检测 | 596 |
| 20.1.3 | 数组 | 564 | 22.1.2 | 作用域安全的构造函数 | 597 |
| 20.2 | 解析与序列化 | 565 | 22.1.3 | 惰性载入函数 | 600 |
| 20.2.1 | JSON 对象 | 565 | 22.1.4 | 函数绑定 | 602 |
| 20.2.2 | 序列化选项 | 566 | 22.1.5 | 函数柯里化 | 604 |
| 20.2.3 | 解析选项 | 569 | 22.2 | 防篡改对象 | 606 |
| 20.3 | 小结 | 570 | 22.2.1 | 不可扩展对象 | 606 |
| 第 21 章 | Ajax 与 Comet | 571 | 22.2.2 | 密封的对象 | 607 |
| 21.1 | XMLHttpRequest 对象 | 571 | 22.2.3 | 冻结的对象 | 608 |
| 21.1.1 | XHR 的用法 | 573 | 22.3 | 高级定时器 | 609 |
| 21.1.2 | HTTP 头部信息 | 575 | 22.3.1 | 重复的定时器 | 610 |
| 21.1.3 | GET 请求 | 576 | 22.3.2 | Yielding Processes | 612 |
| 21.1.4 | POST 请求 | 577 | 22.3.3 | 函数节流 | 614 |
| 21.2 | XMLHttpRequest 2 级 | 578 | 22.4 | 自定义事件 | 616 |
| 21.2.1 | FormData | 578 | 22.5 | 拖放 | 618 |
| 21.2.2 | 超时设定 | 579 | 22.5.1 | 修缮拖动功能 | 620 |
| 21.2.3 | overrideMimeType() 方法 | 580 | 22.5.2 | 添加自定义事件 | 622 |
| 21.3 | 进度事件 | 580 | 22.6 | 小结 | 624 |
| 21.3.1 | load 事件 | 580 | 第 23 章 | 离线应用与客户端存储 | 626 |
| 21.3.2 | progress 事件 | 581 | 23.1 | 离线检测 | 626 |
| 21.4 | 跨源资源共享 | 582 | 23.2 | 应用缓存 | 627 |
| 21.4.1 | IE 对 CORS 的实现 | 582 | 23.3 | 数据存储 | 628 |
| 21.4.2 | 其他浏览器对 CORS 的 实现 | 584 | 23.3.1 | Cookie | 629 |
| | | | 23.3.2 | IE 用户数据 | 637 |

| | | | | | |
|--------|--------------------------|-----|--------|---|-----|
| 23.3.3 | Web 存储机制 | 638 | 25.1.4 | webkitRequestAnimationFrame 与 msRequestAnimationFrame | 685 |
| 23.3.4 | IndexedDB | 643 | 25.2 | Page Visibility API | 686 |
| 23.4 | 小结 | 654 | 25.3 | Geolocation API | 687 |
| 第 24 章 | 最佳实践 | 656 | 25.4 | File API | 689 |
| 24.1 | 可维护性 | 656 | 25.4.1 | FileReader 类型 | 690 |
| 24.1.1 | 什么是可维护的代码 | 656 | 25.4.2 | 读取部分内容 | 692 |
| 24.1.2 | 代码约定 | 657 | 25.4.3 | 对象 URL | 693 |
| 24.1.3 | 松散耦合 | 659 | 25.4.4 | 读取拖放的文件 | 694 |
| 24.1.4 | 编程实践 | 662 | 25.4.5 | 使用 XHR 上传文件 | 695 |
| 24.2 | 性能 | 666 | 25.5 | Web 计时 | 696 |
| 24.2.1 | 注意作用域 | 666 | 25.6 | Web Workers | 697 |
| 24.2.2 | 选择正确方法 | 667 | 25.6.1 | 使用 Worker | 697 |
| 24.2.3 | 最小化语句数 | 672 | 25.6.2 | Worker 全局作用域 | 698 |
| 24.2.4 | 优化 DOM 交互 | 673 | 25.6.3 | 包含其他脚本 | 699 |
| 24.3 | 部署 | 676 | 25.6.4 | Web Workers 的未来 | 700 |
| 24.3.1 | 构建过程 | 676 | 25.7 | 小结 | 700 |
| 24.3.2 | 验证 | 677 | 附录 A | ECMAScript Harmony | 701 |
| 24.3.3 | 压缩 | 679 | 附录 B | 严格模式 | 717 |
| 24.4 | 小结 | 681 | 附录 C | JavaScript 库 | 723 |
| 第 25 章 | 新兴的 API | 682 | 附录 D | JavaScript 工具 | 727 |
| 25.1 | requestAnimationFrame() | 682 | | | |
| 25.1.1 | 早期动画循环 | 682 | | | |
| 25.1.2 | 循环间隔的问题 | 683 | | | |
| 25.1.3 | mozRequestAnimationFrame | 683 | | | |