

SAMS
**Teach
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务


Pearson

 异步图书
www.epubit.com

每章内容针对初学者精心设计，**1**小时轻松阅读学习，
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，
通过**实践**提高应用技能，巩固所学知识

JavaScript

入门经典（第7版）

[美]菲尔·巴拉德（Phil Ballard）著

李强译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



Pearson

JavaScript

入门经典 (第7版)

[美] 菲尔·巴拉德 (Phil Ballard) 著

李强 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

JavaScript入门经典 : 第7版 / (美) 菲尔·巴拉德
(Phil Ballard) 著 ; 李强译. — 北京 : 人民邮电出版
社, 2019.5(2019.8重印)
ISBN 978-7-115-50938-3

I. ①J… II. ①菲… ②李… III. ①JAVA语言—程序
设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第042414号

版权声明

Authorized translation from the English language edition, entitled JAVASCRIPT IN 24 HOURS, SAMS TEACH YOURSELF, 7th Edition by BALLARD, PHIL, published by Pearson Education, Inc., Copyright © 2019.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by POSTS AND TELECOM PRESS CO., LTD., Copyright © 2019.

本书中文简体版由 Pearson Education, Inc. 授权人民邮电出版社有限公司出版。未经出版者书面许可, 不得以任何方式或任何手段复制和抄袭本书内容。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。
版权所有, 侵权必究。

◆ 著 [美] 菲尔·巴拉德 (Phil Ballard)
译 李 强
责任编辑 吴晋瑜
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷

◆ 开本: 800×1000 1/16
印张: 21
字数: 515 千字 2019 年 5 月第 1 版
印数: 2 401—3 900 册 2019 年 8 月河北第 2 次印刷

著作权合同登记号 图字: 01-2018-7753 号

定价: 69.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

本书主要介绍 JavaScript 现有的主要特性，涵盖了 JavaScript 基础知识、编程方法、对象、HTML5 和 CSS3、JavaScript 库、高级特性、代码调试等内容。全书分为六个部分（包括 24 章和一个附录），从基本概念入手，逐步引申到按照当今 Web 标准编写 JavaScript 代码的最佳方式，非常利于初学者学习参考。

本书适合零编程经验的初学者学习，适合作为高等院校计算机专业的教学用书，也适合 Web 开发初学者、JavaScript 初级程序员以及培训机构的学员参考。

前言

在开始阅读本书之前，我们先来看一下本书的目标读者是谁，为什么要写这本书，它所采用的是什么样的体例，其内容是如何组织的，以及需要哪些工具来编写 JavaScript。

读者对象

对于想学习 JavaScript 的读者来说，你们很可能已经掌握了 HTML 和 Web 页面设计的基本知识，希望为网页添加一些更好的互动性。抑或你们目前正使用其他语言进行编程，想了解一下 JavaScript 能够提供哪些更多的功能。

如果对 HTML 没有任何了解，或是没有任何计算机编程经验，我们建议读者先了解一些 HTML 基本知识。鉴于 HTML 非常易于理解，读者不必成为 HTML 专家，就足以了解本书的 JavaScript 范例。

JavaScript 很适合作为学习编程技术的起点。读者在调试过程掌握的基本概念大多可以用于其他编程语言，比如 C、Java 或 PHP。

本书宗旨

JavaScript 最初的用途是相当有限的，它只具备基本的功能，对于浏览器的支持也很不稳定，所以只被视为“花哨的小技巧”。现在，随着浏览器对 W3C 标准的支持越来越好，对 JavaScript 的实现不断改善，JavaScript 已经成为一种常规的编程语言。

其他高级编程语言里的编程规则能够方便地应用于 JavaScript，比如面向对象编程方法有助于编写稳定、易读、易维护和易重用的代码。

所谓“低调”的编程技术和 DOM 脚本都致力于为 Web 页面增加更好的互动，同时保持 HTML 简单易读，并且能够轻松地与代码分离。

本书着力介绍 JavaScript 目前主要特性及基本技巧，从基本概念开始，逐步介绍按照当今 Web 标准编写 JavaScript 代码的最佳方式。

很多编程教程会给出复杂的代码示例和练习，结果让那些没有编程经验的程序员知难而退。为了最大限度地讲清楚且让读者容易上手，本书所给出的示例都是为了阐释每一章的关键知识点，而使用尽可能少和尽可能容易的代码。

本书约定

本书全部代码范例都是用 HTML5 编写的，并且符合 ECMAScript JavaScript 6。

除了每个课程里的正文之外，书中还有一些标记为“说明”“提示”和“注意”的方框。

NOTE

说明：这里的内容给出了额外的解释，帮助读者理解正文和示例。

TIP

提示：这里的内容给出了额外的技巧、提示，帮助读者更轻松地进行编程。

CAUTION

注意：这里的内容帮助读者用相关的信息来避免常见的陷阱。

实践

每一章至少有一部分内容用以指导读者尝试自己完成脚本，帮助读者建立编写 JavaScript 脚本的信心。

问答、测验和练习

每一章的最后都有这三部分内容：

- “问答”——主要用于解答课程中最常遇到的问题；
- “测验”——用于检验读者对课程内容的掌握情况；
- “练习”——根据课程的内容提供一些让读者进一步深入学习的建议。

本书结构

本书正文分为 6 个部分，内容的难度逐步提高。

➤ 第一部分：JavaScript 基础

这部分是 JavaScript 语言的基础知识，介绍了用常用函数编写简单脚本的方法。这部分内容主要针对之前缺少或没有编程知识以及没有 JavaScript 知识的读者编写。

➤ 第二部分：JavaScript 编程

这部分介绍了 JavaScript 的数据类型，例如数值、字符串和数组；此外，还介绍了更复杂的编程范型，比如事件处理、循环控制和定时器等。

➤ 第三部分：理解 JavaScript 对象

这部分着重介绍了如何创建和操作对象，包括遍历和编辑属于 DOM（文档对象模型）的对象。

➤ 第四部分：用 JavaScript 操作 Web 页面

这部分较为深入地介绍了 JavaScript 如何与 HTML（包括 HTML5）和 CSS（包括最新的 CSS3 规范）交互。

➤ 第五部分：与 JavaScript 工具相关的高级技术

这部分介绍了一些专门的编程技术，包括 cookie、正则表达式、闭包和模块的使用。

➤ 第六部分：专业技能

这一部分介绍了专业 JavaScript 开发的知识，例如良好的编程习惯、JavaScript 的调试等。

必要工具

编写 JavaScript 并不需要昂贵和复杂的工具，如集成开发环境（IDE）、编译器或调试器。

本书的范例代码都可以利用像 Windows 记事本这样的文本编辑软件生成。每个操作系统都会提供至少一款这样的软件，而且互联网上还有大量免费或廉价的类似软件。

说明：

附录列出的 JavaScript 开发工具和资源都可以方便地获得。

为了查看代码的运行情况，我们需要一个 Web 浏览器，比如 Microsoft Edge、Mozilla Firefox、Opera、Safari 或 Google Chrome。建议使用浏览器的最新稳定版本。特别是，最好不要使用逐渐被废弃的 Microsoft 的 Internet Explorer，而是使用更加符合标准的 Edge 浏览器，或者使用 Chrome 或 Firefox 来替代它。

本书绝大多数范例代码在运行时并不需要连接互联网，只要把源代码保存到计算机上，然后用浏览器打开它们就可以了。例外的情况是关于 cookie 和 Ajax 的章节，这些代码需要一个 Web 连接（或者是局域网上的一个 Web 服务连接）和一些 Web 空间来上传代码。对于尝试过 HTML 编码的读者来说，都应该具备上述配置；即使没有这些配置，使用业余级别的 Web 主机账户就可以满足要求，而这些都是很便宜的。

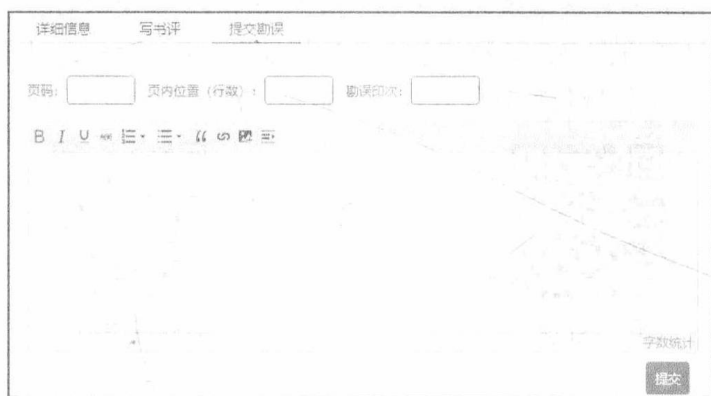
资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎读者将发现的问题反馈给我们，帮助我们提升图书的质量。

读者在发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对读者提交的勘误进行审核，确认并接受后，我们将赠予读者异步社区的 100 积分（积分可用于在异步社区兑换优惠券、样书或奖品）。



扫码关注本书

扫描下方二维码，读者可在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果读者对本书有任何疑问或建议，请发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果读者有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问

www.epubit.com/selfpublish/submission 即可)。

如果学校、培训机构或企业想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果读者在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目 录

| | |
|---|----|
| 第一部分 JavaScript 基础 | |
| 第 1 章 JavaScript 简介 | 3 |
| 1.1 Web 脚本编程基础 | 3 |
| 1.1.1 服务器端与客户端编程 | 4 |
| 1.1.2 JavaScript 简介 | 4 |
| 1.1.3 JavaScript 起源 | 5 |
| 1.1.4 浏览器的竞争 | 5 |
| 1.1.5 <script> 标签 | 6 |
| 1.1.6 DOM 简介 | 7 |
| 1.1.7 W3C 和标准兼容 | 7 |
| 1.1.8 window 和 document 对象 | 7 |
| 1.1.9 对象表示法 | 8 |
| 1.1.10 与用户交互 | 9 |
| 1.1.11 document.write() | 10 |
| 1.1.12 读取 document 对象的 属性 | 12 |
| 1.2 小结 | 12 |
| 1.3 问答 | 12 |
| 1.4 作业 | 13 |
| 1.4.1 测验 | 13 |
| 1.4.2 答案 | 14 |
| 1.5 练习 | 14 |
| 第 2 章 创建简单的脚本 | 15 |
| 2.1 在 Web 页面里添加 JavaScript | 15 |
| 2.2 编写 JavaScript 语句 | 17 |
| 2.3 变量 | 18 |
| 2.4 操作符 | 19 |
| 2.4.1 算术操作符 | 20 |
| 2.4.2 操作符优先级 | 21 |
| 2.4.3 对字符串使用 操作符 “+” | 21 |
| 2.5 捕获鼠标事件 | 23 |
| 2.5.1 onClick 事件处理器 | 23 |
| 2.5.2 onMouseOver 和 onMouseOut 事件 处理器 | 24 |
| 2.6 小结 | 26 |
| 2.7 问答 | 26 |
| 2.8 作业 | 27 |
| 2.8.1 测验 | 27 |
| 2.8.2 答案 | 27 |
| 2.9 练习 | 28 |
| 第 3 章 使用函数 | 29 |
| 3.1 基本语法 | 30 |
| 3.2 调用函数 | 30 |
| 3.2.1 把 JavaScript 代码放置到 页面的 <head> 区域 | 31 |
| 3.2.2 给函数传递参数 | 32 |
| 3.2.3 多参数 | 32 |
| 3.2.4 从函数返回值 | 34 |
| 3.2.5 匿名函数 | 35 |
| 3.3 小结 | 36 |

| | | | | | |
|--------------|--------------------------|-----------|---------------------------|---|-----------|
| 3.4 | 问答 | 36 | 5.5.5 | 取整 | 56 |
| 3.5 | 作业 | 36 | 5.5.6 | 获得最大值和最小值 | 56 |
| 3.5.1 | 测验 | 36 | 5.5.7 | 生成随机数 | 56 |
| 3.5.2 | 答案 | 37 | 5.5.8 | 数学常数 | 57 |
| 3.6 | 练习 | 37 | 5.5.9 | 关键字 with | 57 |
| 第 4 章 | 函数的更多知识 | 38 | 5.6 | 小结 | 59 |
| 4.1 | 变量作用域 | 38 | 5.7 | 问答 | 60 |
| 4.1.1 | 使用 this 关键字 | 40 | 5.8 | 作业 | 60 |
| 4.1.2 | 使用 let 和 const | 41 | 5.8.1 | 测验 | 60 |
| 4.1.3 | 用 const 关键字声明 变量 | 42 | 5.8.2 | 答案 | 61 |
| 4.2 | 箭头函数 | 44 | 5.9 | 练习 | 61 |
| 4.3 | 设置默认参数 | 45 | 第二部分 JavaScript 编程 | | |
| 4.4 | 小结 | 45 | 第 6 章 | 数字和字符串 | 65 |
| 4.5 | 问答 | 46 | 6.1 | 数值 | 65 |
| 4.6 | 作业 | 46 | 6.1.1 | 整数 | 66 |
| 4.6.1 | 测验 | 46 | 6.1.2 | 浮点数 | 66 |
| 4.6.2 | 答案 | 47 | 6.1.3 | 十六进制数、二进制数和 八进制数 | 66 |
| 4.7 | 练习 | 47 | 6.2 | 全局方法 | 67 |
| 第 5 章 | DOM 对象和内置对象 | 48 | 6.2.1 | toString() | 67 |
| 5.1 | 与用户交互 | 48 | 6.2.2 | toFixed() | 68 |
| 5.1.1 | alert() | 49 | 6.2.3 | toExponential() | 68 |
| 5.1.2 | confirm() | 49 | 6.3 | Number 对象 | 68 |
| 5.1.3 | prompt() | 49 | 6.3.1 | Number.isNaN() | 69 |
| 5.2 | 根据 id 选择元素 | 50 | 6.3.2 | Number.isInteger() | 69 |
| 5.3 | 访问浏览器历史记录 | 51 | 6.3.3 | Number.parseFloat()和 Number.parseInt() | 69 |
| 5.4 | 使用 location 对象 | 51 | 6.3.4 | 无穷大 | 70 |
| 5.4.1 | 用 location 对象导航 | 52 | 6.4 | Number()函数 | 70 |
| 5.4.2 | 刷新页面 | 52 | 6.5 | 布尔值 | 71 |
| 5.4.3 | 获取浏览器信息: navigator 对象 | 53 | 6.6 | null (空) 和 undefined (未定义) | 72 |
| 5.5 | 日期和时间 | 54 | 6.7 | 小结 | 73 |
| 5.5.1 | 创建具有当前日期和时间的 Date 对象 | 54 | 6.8 | 问答 | 73 |
| 5.5.2 | 创建具有指定日期和时间的 Date 对象 | 55 | 6.9 | 作业 | 73 |
| 5.5.3 | 设置和编辑日期与时间 | 55 | 6.9.1 | 测验 | 73 |
| 5.5.4 | 利用 Math 对象简化 运算 | 55 | 6.9.2 | 答案 | 74 |
| | | | 6.10 | 练习 | 74 |
| | | | 第 7 章 | 操作字符串 | 75 |

| | | | |
|-------------------------------------|-----------|------------------------------------|------------|
| 7.1 字符串..... | 75 | 9.3 event 对象..... | 98 |
| 7.1.1 length 属性..... | 76 | 9.3.1 阻止默认行为..... | 99 |
| 7.1.2 转义序列..... | 76 | 9.3.2 事件冒泡和捕获..... | 102 |
| 7.1.3 字符串方法..... | 77 | 9.3.3 关闭冒泡和捕获..... | 102 |
| 7.2 模板字符串..... | 80 | 9.4 小结..... | 103 |
| 7.3 小结..... | 81 | 9.5 问答..... | 103 |
| 7.4 问答..... | 81 | 9.6 作业..... | 104 |
| 7.5 作业..... | 81 | 9.6.1 测验..... | 104 |
| 7.5.1 测验..... | 82 | 9.6.2 答案..... | 104 |
| 7.5.2 答案..... | 82 | 9.7 练习..... | 105 |
| 7.6 练习..... | 83 | 第 10 章 程序控制..... | 106 |
| 第 8 章 数组..... | 84 | 10.1 条件语句..... | 106 |
| 8.1 数组..... | 84 | 10.1.1 if()语句..... | 106 |
| 8.1.1 创建新数组..... | 85 | 10.1.2 比较操作符..... | 107 |
| 8.1.2 初始化数组..... | 85 | 10.1.3 测试相等性..... | 109 |
| 8.1.3 数组的方法..... | 85 | 10.1.4 if 进阶..... | 110 |
| 8.2 如何遍历数组..... | 89 | 10.1.5 测试多个条件..... | 110 |
| 8.2.1 使用 forEach()..... | 89 | 10.1.6 switch 语句..... | 111 |
| 8.2.2 使用 map()..... | 90 | 10.1.7 逻辑操作符..... | 111 |
| 8.2.3 使用 for-of 访问数组..... | 90 | 10.2 循环和控制结构..... | 112 |
| 8.3 三点表示法..... | 91 | 10.2.1 while..... | 112 |
| 8.3.1 组合多个数组..... | 91 | 10.2.2 do...while..... | 113 |
| 8.3.2 用参数的数组来调用 函数..... | 91 | 10.2.3 for..... | 113 |
| 8.3.3 收集数据..... | 91 | 10.2.4 用 break 跳出循环..... | 113 |
| 8.4 小结..... | 92 | 10.2.5 用 for...in 在对象集里 循环..... | 114 |
| 8.5 问答..... | 92 | 10.3 设置和使用定时器..... | 115 |
| 8.6 作业..... | 92 | 10.3.1 setTimeout()..... | 115 |
| 8.6.1 测验..... | 92 | 10.3.2 setInterval()..... | 116 |
| 8.6.2 答案..... | 93 | 10.4 小结..... | 116 |
| 8.7 练习..... | 93 | 10.5 问答..... | 116 |
| 第 9 章 用 JavaScript 处理事件..... | 94 | 10.6 作业..... | 116 |
| 9.1 事件的类型..... | 94 | 10.6.1 测验..... | 116 |
| 9.2 事件处理器..... | 96 | 10.6.2 答案..... | 117 |
| 9.2.1 内联事件处理器..... | 96 | 10.7 练习..... | 117 |
| 9.2.2 作为 DOM 对象的属性的 事件处理器..... | 96 | 第三部分 理解 JavaScript 对象 | |
| 9.2.3 使用 addEventListener()..... | 97 | 第 11 章 面向对象编程..... | 121 |
| 9.2.4 添加多个监听器..... | 98 | 11.1 什么是面向对象编程..... | 121 |

| | | | |
|-----------------------------------|------------|------------------------------------|------------|
| 11.2 创建对象 | 122 | 13.4.3 编辑元素属性 | 159 |
| 11.2.1 创建直接实例 | 123 | 13.4.4 动态加载 JavaScript 文件 | 159 |
| 11.2.2 使用关键字 this | 123 | 13.5 小结 | 164 |
| 11.2.3 匿名函数 | 125 | 13.6 问答 | 164 |
| 11.2.4 使用构造函数 | 126 | 13.7 作业 | 165 |
| 11.3 用 prototype 扩展和继承对象 | 128 | 13.7.1 测验 | 165 |
| 11.3.1 扩展对象 | 128 | 13.7.2 答案 | 165 |
| 11.3.2 继承对象 | 129 | 13.8 练习 | 166 |
| 11.4 封装 | 132 | 第 14 章 JSON 简介 | 167 |
| 11.5 小结 | 132 | 14.1 JSON 是什么 | 167 |
| 11.6 问答 | 132 | 14.2 访问 JSON 数据 | 168 |
| 11.7 作业 | 133 | 14.2.1 使用 eval() | 169 |
| 11.7.1 测验 | 133 | 14.2.2 使用直接浏览器 JSON 支持 | 169 |
| 11.7.2 答案 | 133 | 14.2.3 使用 JSON.parse() | 170 |
| 11.8 练习 | 134 | 14.3 JSON 的数据序列化 | 170 |
| 第 12 章 对象进阶 | 135 | 14.4 JSON 数据类型 | 172 |
| 12.1 类 | 135 | 14.5 模拟关联数组 | 172 |
| 12.1.1 使用 getter 和 setter | 137 | 14.6 使用 JSON 创建对象 | 173 |
| 12.1.2 Symbol | 138 | 14.6.1 属性 | 173 |
| 12.2 对象继承 | 139 | 14.6.2 方法 | 174 |
| 12.3 使用功能检测 | 140 | 14.6.3 数组 | 174 |
| 12.4 小结 | 142 | 14.6.4 对象 | 175 |
| 12.5 问答 | 143 | 14.7 JSON 安全性 | 177 |
| 12.6 作业 | 143 | 14.8 小结 | 177 |
| 12.6.1 测验 | 143 | 14.9 问答 | 177 |
| 12.6.2 答案 | 144 | 14.10 作业 | 177 |
| 12.7 练习 | 144 | 14.10.1 测验 | 177 |
| 第 13 章 DOM 脚本编程 | 145 | 14.10.2 答案 | 178 |
| 13.1 DOM 节点 | 146 | 14.11 练习 | 178 |
| 13.1.1 节点类型 | 147 | 第四部分 用 JavaScript 操作 Web 页面 | |
| 13.1.2 使用 nodeValue | 150 | 第 15 章 HTML 与 JavaScript 编程 | 181 |
| 13.1.3 使用 nodeName | 150 | 15.1 HTML5 的新标签 | 182 |
| 13.2 用 getElementByTagName() 选择元素 | 151 | 15.2 一些重要的新元素 | 182 |
| 13.3 读取元素的属性 | 152 | 15.2.1 用 <video> 回放视频 | 182 |
| 13.4 Mozilla 的 DOM 查看器 | 153 | 15.2.2 用 canPlayType() 测试可用的格式 | 183 |
| 13.4.1 创建新节点 | 154 | | |
| 13.4.2 操作子节点 | 155 | | |

| | | | | | |
|---------------|-------------------------|------------|---------------|-------------------------------|------------|
| 15.2.3 | 控制回放 | 184 | 17.3.3 | 多背景图像 | 213 |
| 15.2.4 | 用<audio>标签播放声音 | 184 | 17.4 | CSS3 渐变 | 214 |
| 15.2.5 | 用<canvas>在页面上绘图 | 185 | 17.4.1 | 线性渐变 | 214 |
| 15.3 | 拖放 | 187 | 17.4.2 | 放射渐变 | 215 |
| 15.4 | 本地存储 | 190 | 17.5 | CSS3 文本效果 | 216 |
| 15.5 | 操作本地文件 | 190 | 17.5.1 | 文本阴影 | 216 |
| 15.6 | 小结 | 192 | 17.5.2 | 单词换行 | 217 |
| 15.7 | 问答 | 192 | 17.6 | CSS3 过渡、变换和动画 | 217 |
| 15.8 | 作业 | 193 | 17.7 | 在 JavaScript 中引用 CSS3 属性 | 218 |
| 15.8.1 | 测验 | 193 | 17.7.1 | 将 CSS 属性名转换到 JavaScript 中 | 218 |
| 15.8.2 | 答案 | 193 | 17.7.2 | DOM getComputedStyle() 方法 | 218 |
| 15.9 | 练习 | 194 | 17.8 | 设置带有厂商前缀的 CSS3 属性 | 220 |
| 第 16 章 | JavaScript 和 CSS | 195 | 17.9 | 小结 | 221 |
| 16.1 | CSS 简介 | 196 | 17.10 | 问答 | 221 |
| 16.1.1 | 从内容分离样式 | 196 | 17.11 | 作业 | 222 |
| 16.1.2 | CSS 样式声明 | 196 | 17.11.1 | 测验题 | 222 |
| 16.1.3 | 在哪里保存样式声明 | 197 | 17.11.2 | 答案 | 222 |
| 16.2 | DOM 的 style 属性 | 198 | 17.12 | 练习 | 223 |
| 16.3 | 用 className 访问类 | 200 | 第五部分 | 与 JavaScript 工具相关的高级技术 | |
| 16.4 | DOM 的 styleSheets 对象 | 202 | 第 18 章 | 读取和写入 cookie | 227 |
| 16.5 | 小结 | 206 | 18.1 | 什么是 cookie | 227 |
| 16.6 | 问答 | 206 | 18.2 | 使用 document.cookie 属性 | 228 |
| 16.7 | 作业 | 207 | 18.3 | cookie 组成 | 229 |
| 16.7.1 | 测验 | 207 | 18.4 | 编写 cookie | 230 |
| 16.7.2 | 答案 | 207 | 18.5 | 编写一个函数来写 cookie | 230 |
| 16.8 | 练习 | 208 | 18.6 | 读取 cookie | 232 |
| 第 17 章 | CSS3 简介 | 209 | 18.7 | 删除 cookie | 233 |
| 17.1 | 特定厂商的属性和前缀 | 209 | 18.8 | 在一个 cookie 里设置多个值 | 235 |
| 17.2 | CSS3 边框 | 210 | 18.9 | 小结 | 236 |
| 17.2.1 | 创建边框阴影 | 211 | 18.10 | 问答 | 236 |
| 17.2.2 | 用 border-radius 属性实现圆角 | 211 | 18.11 | 作业 | 236 |
| 17.3 | CSS3 背景 | 212 | 18.11.1 | 测验 | 236 |
| 17.3.1 | background-size 属性 | 212 | 18.11.2 | 答案 | 237 |
| 17.3.2 | background-origin 属性 | 213 | 18.12 | 练习 | 237 |

| | | | |
|----------------------------|-----|-----------------------------|-----|
| 第 23 章 调试代码 | 287 | 24.4 jQuery 入门 | 302 |
| 23.1 调试简介 | 287 | 24.4.1 在页面里引用 jQuery | 302 |
| 23.1.1 错误类型 | 287 | 24.4.2 jQuery 的\$(document) | |
| 23.1.2 选择程序员的 | | .ready 处理器 | 303 |
| 编辑器 | 288 | 24.4.3 选择页面元素 | 303 |
| 23.1.3 用 alert()进行简单 | | 24.4.4 操作 HTML 内容 | 304 |
| 调试 | 288 | 24.4.5 显示和隐藏元素 | 305 |
| 23.2 更高级的调试 | 289 | 24.4.6 命令链 | 306 |
| 23.2.1 控制台 | 289 | 24.4.7 处理事件 | 306 |
| 23.2.2 分组消息 | 292 | 24.5 The jQuery UI | 307 |
| 23.2.3 用断点停止代码 | | 24.6 Ajax 简介 | 308 |
| 执行 | 293 | 24.6.1 Ajax 入门 | 309 |
| 23.2.4 条件性断点 | 297 | 24.6.2 XMLHttpRequest | |
| 23.2.5 从代码中启动 | | 对象 | 310 |
| 调试器 | 297 | 24.6.3 创建 request 对象 | 310 |
| 23.2.6 验证 JavaScript | 298 | 24.6.4 方法和属性 | 310 |
| 23.3 小结 | 298 | 24.6.5 与服务器通信 | 311 |
| 23.4 问答 | 298 | 24.6.6 在服务器端发生了 | |
| 23.5 作业 | 298 | 什么 | 311 |
| 23.5.1 测验 | 298 | 24.6.7 处理服务器响应 | 311 |
| 23.5.2 答案 | 299 | 24.6.8 还有更容易的方法， | |
| 23.6 练习 | 299 | 不是吗? | 312 |
| 第 24 章 继续深入学习 | 300 | 24.7 用 jQuery 实现 Ajax | 312 |
| 24.1 为什么要使用库 | 300 | 24.8 Node.js 简介 | 313 |
| 24.2 库能做什么 | 301 | 24.9 小结 | 314 |
| 24.3 一些常见的库和框架 | 301 | 24.10 问答 | 314 |
| 24.3.1 Prototype 框架 | 301 | 24.11 作业 | 314 |
| 24.3.2 Dojo | 301 | 24.11.1 测验 | 315 |
| 24.3.3 React | 302 | 24.11.2 答案 | 315 |
| 24.3.4 Node.js | 302 | 24.12 练习 | 315 |
| 24.3.5 jQuery | 302 | 附录 JavaScript 开发工具 | 317 |

第一部分

JavaScript 基础

- 第 1 章 JavaScript 简介
- 第 2 章 创建简单的脚本
- 第 3 章 使用函数
- 第 4 章 函数的更多知识
- 第 5 章 DOM 对象和内置对象