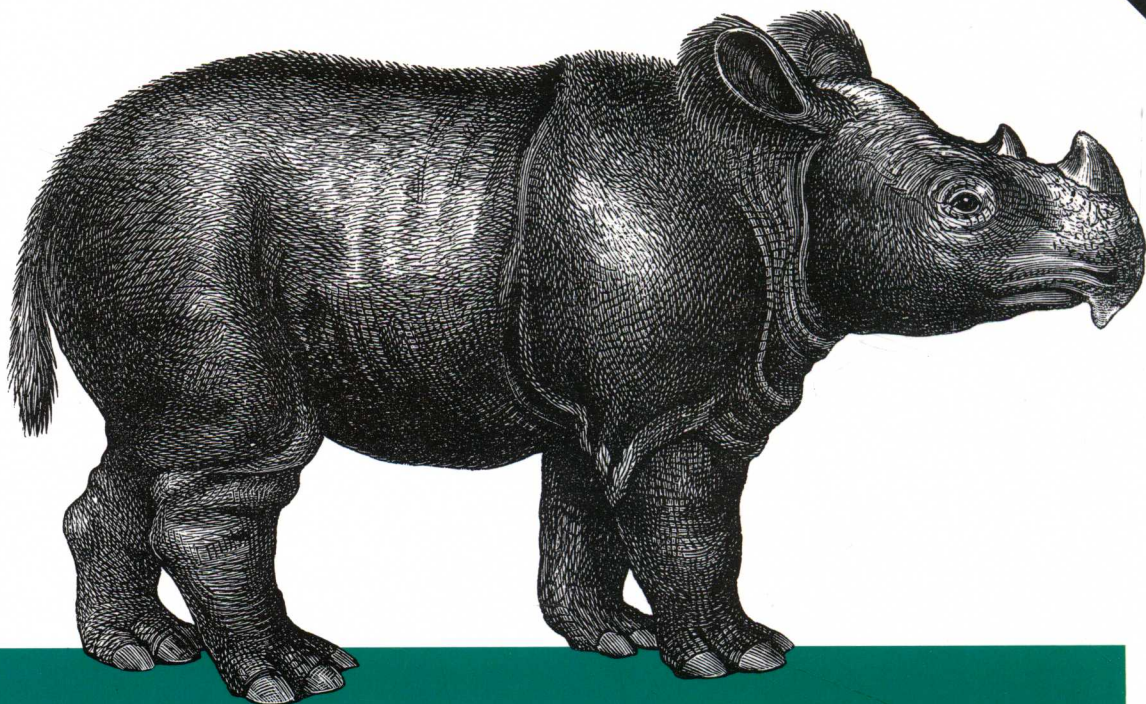


O'REILLY®

异步图书
www.epubit.com.cn

第3版



JavaScript 学习指南

Learning JavaScript 让网页变得栩栩如生的艺术

[美] Ethan Brown 著
娄佳 袁慎建 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

JavaScript 学习指南 (第 3 版)

[美] Ethan Brown 著

娄 佳 译
袁慎建

人 民 邮 电 出 版 社

北 京

图书在版编目 (C I P) 数据

JavaScript学习指南：第3版 / (美) 布朗
(Ethan Brown) 著；娄佳，袁慎建译. — 北京：人民
邮电出版社，2017.7
ISBN 978-7-115-45632-8

I. ①J… II. ①布… ②娄… ③袁… III. ①JAVA语
言—程序设计—指南 IV. ①TP312.8-62

中国版本图书馆CIP数据核字(2017)第130679号

版 权 声 明

© PT Press 2017

Authorized translation of the English edition of High Performance Python, First Edition © 2014 SitePoint
Pty. Ltd. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all
rights to publish and sell the same.

All rights reserved.

本书中文简体字版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可，对本书
任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

-
- ◆ 著 [美]Ethan Brown
 - 译 娄 佳 袁慎建
 - 责任编辑 陈冀康
 - 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>
 - 三河市海波印务有限公司印刷
 - ◆ 开本：800×1000 1/16
 - 印张：20.75
 - 字数：385 千字 2017 年 7 月第 1 版
 - 印数：1—2 400 册 2017 年 7 月河北第 1 次印刷
- 著作权合同登记号 图字：01-2016-7578 号
-

定价：59.00 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

JavaScript 是目前 Web 开发领域非常流行的一种编程语言，得到众多 IT 从业人员和编程爱好者的关注。现在市面上的 JavaScript 图书多数基于 ES5，而本书重点关注 ES6。ES6 是语言的新标准，是目前业界超级活跃的计算机语言。

本书共分为 22 章，在 ES6 的基础上，本书系统地介绍了 JavaScript 的基本语法、语言特性、相关工具、基本对象、技术以及推动现代 JavaScript 开发的范例等方面的知识。其内容由浅及深，从变量、控制流、函数等内容到异步编程、正则表达式等。

本书适合具备一定 JavaScript 基础知识的读者阅读，也适合从事程序设计工作并想要深入探索 JavaScript 语言的读者阅读。

前言

虽然这已经是我在 JavaScript 技术领域的第二本书了，但对于 JavaScript 专家和布道师这个角色，我仍然觉得有些意外。和很多开发人员一样，我对 JavaScript 持有强烈的偏见，这种偏见一直持续到 2012 年。即便现在我的思想已经发生了转变，我的迷惑仍然存在。

我对 JavaScript 持有偏见的缘由很简单：我认为 JavaScript 是一种“玩具”语言（我并没有真正地好好学习过它，因此不是完全了解我所评价的东西），它经常被那些危险、粗心、未经培训的业余程序员所使用。这些现象都是有事实依据的。ES6 开发速度很快，即使是它的发明人 Brendan Eich 后来也承认有些东西在一开始并没有考虑周全，而当他意识到的时候，已经有太多人指望他能够对这些问题做出有效的改进（然而，又有哪个语言没有这方面的困扰呢）。第二个原因是，JavaScript 确实让编程突然变得更加容易上手。这不仅是因为浏览器的普及，更是因为它的高回报/投入比，人们只要付出一点点努力，就可以收获 JavaScript 为他们的网站所创造的巨大价值。人们可以通过试错，通过阅读彼此的代码，甚至在很多情况下，通过模仿那些缺乏理解的不良代码来学习这门编程语言。

我很庆幸自己在学习了足够多的 JavaScript 知识后，意识到它不仅不是一种玩具语言，而且拥有极其雄厚的基础，强大、灵活、具有表现力。我也很庆幸自己能够毫无芥蒂地拥抱 JavaScript 所带来的简易特性。毫无疑问，我对业余爱好者没有任何敌意：每个人都会找到自己的起点，而编程是一项有益的技能，以软件开发作为职业则会带来更多好处。

对于新手和业余的开发人员，我会说：成为一个业余开发人员并不是一件羞耻的事情。但如果一直停留在业余水平，那么就有点不合适了（如果你决定把编程作为职业）。如果你想练习编程，那么就付诸行动。搜集所有你能找到的资源，学习任何你可以学到的东西。保持谦虚、开放的心态，最重要的是，学会质疑。质疑一切权威，质疑任何一个经验丰富的开发人员，多问几句“为什么？”

多数情况下，我都尽量尝试让本书的内容符合 JavaScript 的“事实”，但是依然不可避免地会有不同的意见。当我提出不同意见时，按照原有的意思理解它们就行。

我非常欢迎不同的看法，我也鼓励读者去寻求其他有经验的开发人员的意见。

如今正是学习 JavaScript 最好的时代。互联网正在逐渐脱离蹒跚学步（从技术的角度上）的阶段，网站开发领域也不再处于 5~10 年前那种令人不解的拓荒阶段。一些类似 HTML5 或 ES6 之类的标准使得学习网站开发变得越来越简单，开发出高质量的网站应用也不再是一件困难的事情。Node.js 使 JavaScript 的应用范围不再局限于浏览器，现在的 JavaScript 已经可以用来开发系统脚本、桌面应用程序、网站后端，甚至是嵌入式应用。可以肯定的是，我从 20 世纪 80 年代中期开始编程以来，从未觉得编程是一件如此有趣的事。

JavaScript 简史

Brendan Eich 于 1995 年开发了 JavaScript，他曾就职于网景通信公司（Netscape Communications Corporation）。最早的 JavaScript 是在很短时间内开发出来的，很多批判它的人都说它的开发缺乏远见。然而，Brendan Eich 并不是一个浅尝辄止的人：他有着非常扎实的计算机科学基础，对 JavaScript 有着成熟和颇具先见之明的想法。从很多方面来说，这些想法是超越时代的，主流开发者们花了 15 年的时间才逐渐理解了这门语言的先进性。

JavaScript 最早的名字是 Mocha，在 1995 年网景领航员浏览器（Netscape Navigator）的一次发布中被正式命名为 JavaScript 之前，它曾经短期使用了 LiveScript 这个名字。“JavaScript”中的单词“Java”并不是巧合，但是却有点费解：撇开通用的语法传统不说，相比 Java，JavaScript 更类似于 Self（一种基于原型的语言，20 世纪 80 年代中期由 Xerox PARC 所开发）和 Scheme（一种深受 Lisp 和 ALGOL 影响的语言，20 世纪 70 年代由 Guy Steele 和 Gerald Sussman 所开发）。Eich 精通 Self 和 Scheme，他在开发 JavaScript 时应用了这两种语言中的一些具有前瞻性思考的范例。JavaScript 这个名字的由来一部分是出于市场需求，当时的 Java 语言非常流行，所以将它们关联起来^①。

1996 年 11 月，网景通信公司声明他们已经把 JavaScript 提交到 Ecma。Ecma 是一家私有的、国际化非营利性标准化组织，它在技术和通信行业有着重大影响。Ecma 国际发布了第一版 ECMA-26 规格，其本质就是 JavaScript。

Ecma 规格（一种叫作 ECMAScript 的语言）和 JavaScript 的区别更多体现在学术方面。从技术上说，JavaScript 是 ECMA 的一种实现，但在实际应用中，JavaScript 和 ECMAScript 是可以互换的。

① 在 2014 年的一次公开采访中，Eich 对“憎恨 JavaScript”的 Sun Microsystems 公司嗤之以鼻。

ECMAScript 的上一个主版本是 5.1（通常也称为 ES5），发布于 2011 年 6 月。市面上现存的老到不支持 ECMAScript 5.1 的“非正规”浏览器已经不足个位数，可以很有把握地说，ECMAScript 5.1 已经是现今的网络通用语言了。

ECMAScript 6（ES6）作为本书的重点，由 Ecma 国际发布于 2015 年 6 月。在正式发布之前，它被叫作“Harmony”，所以你可能会听到有人把 ES6 称作“Harmony”“ES6 Harmony”“ES6”“ES2015”以及“ECMAScript 2015”。在本书中，我们将其统称为 ES6。

ES6

如果说 ES5 是目前的网络通用语言的话，有心的读者可能会奇怪为什么本书要重点关注 ES6 呢。

ES6 代表了 JavaScript 语言的一个重大进步，ES5 的一些缺陷也在 ES6 中得到了改进。你一定会发现，在工作中，ES6 是一门更加令人愉快和强大的语言（从 ES5 开始入门则会很享受）。同时，感谢转换器，你现在可以编写 ES6 代码，然后把它们转化成“浏览器可识别的”ES5 代码。

随着 ES6 的最终发布，浏览器对它的支持也会越来越稳定，总有一天，转换器将从历史的舞台褪去（我不会预测这件事什么时候会发生，即使是粗略地估计）。

毫无疑问，ES6 代表了 JavaScript 开发的未来，你现在花些时间来学，就是为未来做准备，我们现在使用转换器，可以保留代码的可移植性。

然而，并不是每个开发人员都希望编写 ES6 代码。很可能你正在一个非常庞大的基于 ES5 的代码库上工作，将这些代码转换成 ES6 的成本非常高，一些开发人员仅仅是不想为此花费额外的精力。

除了第 1 章以外，本书将重点讲解 ES6，而非 ES5。我会在恰当的时机指出 ES6 和 ES5 的不同之处，但是不会出现对照代码的例子，或者在更适合使用 ES6 的时候出现大量关于“ES5 用法”的讨论。如果你恰好是那种无论如何都要坚持使用 ES5 的开发人员，那么本书可能不适合你（即使这样我还是希望未来的某一天你可以开始使用 ES6）。

从编纂的角度来说，选择 ES6 作为本书的重点这一决定非常谨慎。ES6 里程碑式的进步使得维持其清晰的教学框架变得非常困难。简而言之，如果一本书试图同时涵盖关于 ES5 和 ES6 的内容，很可能会顾此失彼。

本书的目标读者

本书主要针对那些有一定编程经验的读者（即使只听过一些介绍编程的课程，或者在线课程）。如果你刚接触编程，本书将会对你非常有帮助，不过你可能需要一些介绍性的文章或者课程作为补充。

那些已经有一些 JavaScript 经验（尤其是只有 ES5 的经验）的读者，也会在本书中找到实用且全面的重要语言概念。

从其他编程语言转过来的开发人员对本书的内容应该会有一见如故的感觉。

本书将全面涵盖关于语言特性、相关工具、技术，以及推动现代 JavaScript 开发的范例等方面的知识。因此，本书中的素材也将相应地从简明易懂（如变量、控制流、函数）到复杂深奥（如异步编程、正则表达式）变化。根据你的经验情况，你可能会发现阅读其中某些章节颇具挑战性。毫无疑问，初学者将需要反复阅读其中某些章节。

本书的遗憾之处

本书不是关于 JavaScript 或其相关类库的索引大全。Mozilla 开发者网络（Mozilla Developer Network, MDN）维护了一个非常出色、全面、实时更新，并且免费的在线 JavaScript 索引（<https://developer.mozilla.org/en-US/docs/Web/JavaScript>），本书将会自由地引用以上类库。如果你更喜欢实体书，David Flanagan 的《JavaScript 权威指南》则更加全面（尽管它在本书编写的时候还未涵盖 ES6 的内容）。

本书的排版约定

本书将使用以下印刷相关约定：

斜体

表示新的术语、网址、邮件地址、文件名及文件扩展名。

等宽字体

用于代码列表，包括段落内引用代码元素的地方，比如，变量或者函数名、数据库、数据类型、环境变量、语句及关键字。

等宽加粗

表示命令行或者其他应该由用户输入的文本。

等宽斜体

表示应由用户输入或者由上下文决定的值。



此标志表示小窍门或者建议。



此标志表示一般性的注解。



此标志表示警告或警示。

Safari® Books Online

 Safari Books Online 是一个按需数字图书馆，它提供了来自全球各地技术和商业领域优秀作者的专业书籍和视频。

专业技术人员、软件开发人员、网页设计人员，以及商业和创意专业人士等将 Safari Books Online 用作其首选的内容数据库，进行搜索、解决问题、学习和认证培训。

Safari Books Online 为企业、政府、教育机构和个人提供一系列计划和定价。

会员有权在这个完全可搜索的内容数据库中访问来自 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett 和 Course Technology 等上百家出版商的上千本书籍、培训影片，以及正式出版前的手稿。了解更多有关 Safari Books Online 的信息，请访问我们的官方网站。

我们的联系方式

请通过以下方式把对本书的评价和问题提交给出版商：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
电话: 800-998-9938 (美国或者加拿大)
707-829-0515 (国际或者本地)
707-829-0104 (传真)

关于本书的勘误、示例和其他信息, 请访问官方页面。

关于本书的意见, 建议或技术问题, 请发邮件到: bookquestions@oreilly.com。

了解更多有关我们的图书、课程、会议的信息以及最新动态, 请访问我们的官方推特。也可以在 Youtube 上观看。

致谢

能够有机会为 O'Reilly 写书是一个巨大的荣耀, 我欠 Simon St.Laurent 一个大人情, 因为他在我身上看到了这种潜能, 并且带我走向写书这条路。Meg Foley 是我的编辑, 他一直在支持我, 鼓励我, 对我有前所未有的帮助 (我给你寄了一件非常绿的 T 恤衫, Meg!)。每一本 O'Reilly 出版的书都是一个团队努力的成果, 我的文字编辑 Rachel Monaghan, 制作编辑 Kristen Brown, 以及校对员 Jasmine Kwityn, 他们在跟我合作中都非常迅速, 全面, 富有见解: 谢谢每个人的努力!

我的技术审稿人, Matt Inman、Shelley Powers、Nick Pinkham 和 Cody Lindley, 感谢你们敏锐的反馈, 绝妙的想法, 以及不遗余力地帮我润色这本书。可以说没有你们我就无法完成这本书。当然, 每个人的反馈都非常有帮助, 这里我想特别感谢一下 Matt: 他深厚的教育学背景为我提供了一个很有价值的教学视角, 他经常在反馈中使用的 Stephen Colbert 的图片帮我理清思路!

在此要特别感谢 Shelley Powers——本书上一个版本的作者, 不仅仅是因为她把这个头衔传递给我, 更因为她为本书提供了非常有经验的反馈, 从而让本书变得更好 (并且引发了一些激烈的讨论)。

我想感谢所有读过我上一本书 (《Node 与 Express 开发》) 的读者。如果没有你们的购买, 以及积极地评论, 我可能就没有机会写本书了。在此特别感谢提出反馈和修正建议的读者: 从你们的反馈中我学到了很多!

我很荣幸能在 Pop Art 工作, 感谢这里的每一位同事, 你们是我坚实的依靠。你们的支持令我自惭形秽, 你们的热情激励着我, 你们的专业精神和奉献精神是我每天

起床的动力。在这里特别感谢 Tom Paul：他坚定不移的原则，创新的经营理念，以及卓越的领导力时刻激励着我不仅要做好当下，还要放眼未来，不断进步。感谢 Steve Rosenbaum 创办 Pop Art，并在度过惊涛骇浪后将火炬成功地传递给 Tom。在我写作本书的过程中，Colwyn Fritze-Moor 和 Eric Buchmann 帮助处理了一些我的日常工作，感谢你们！感谢 Dylan Hallstrom 成为可靠的模范！感谢 Liz Tom 和 Sam Wilskey 加入 Pop Art 的团队！感谢 Carole Hardy、Nikki Brovold、Jennifer Erts、Randy Keener、Patrick Wu 和 Lisa Melogue 给我的所有支持。最后，感谢我的前辈 Tony Alferes、Paul Inman 和 Del Olds，从他们那里我学到了很多。

我对这本书以及编程语言的热情，是被 Dan Resler 博士点燃的，他是弗吉尼亚联邦大学的副教授。我在参加他的编译原理课程的时候缺乏兴趣，但是课程结束后却对形式语言理论产生了浓厚的兴趣。感谢您用自己的热情感染我，用您深邃的思想影响我。

感谢我所有在 PSU 中兼职 MBA 的朋友们，很高兴认识大家！特别感谢 Cathy、Amanda、Miska、Sahar、Paul S.、Cathy、John R.、Laurie、Joel、Tyler P.、Tyler S. 和 Jess：是你们让我的人生更加充实！

Pop Art 的同事们激励我变得杰出，点亮我的白天；朋友们激励我变得更有深度，照亮我的夜晚。Mark Booth：没有人比你更了解我，也没有人能像你这样获得我最深的信任。你的创造力和天赋让我感到惭愧，千万别把这本书拿到你面前当做炫耀的资本。Katy Roberts 像定期来临的潮水一样可靠，美丽的 Katy，感谢你深厚持久的善良和友谊。Sarah Lewis：我喜欢你的样子。Byron 和 Amber Clayton 是我真诚而忠实的朋友，他们总是能带给我欢乐。Lorraine，这么多年过去了，你总是能让我展现出最好的一面。在这里我想对 Kate Nahas 说：很高兴多年之后我们恢复了联系，我很期待为我们拥有在杜克大学的共同回忆而举杯。Desember：感谢你的信任，温暖和友谊。最后，感谢我的新朋友 Chris Onstad 和 Jessica Rowe：你们在过去两年中给我的生活带来如此多的欢笑，真不知道没有你们我该怎么办。

在这里我想对我的母亲 Ann 说：感谢您给予我坚定不移的支持、爱和耐心。我的父亲 Tom，使我一直保持好奇心、创新和奉献精神，如果没有他我可能只是一个可怜的工程帅（或者压根不是工程帅）。还有我的姐姐，Meris，她将永远是我生命中那个代表着忠诚和信念的不动点。

目录

第 1 章 第一个应用	1
1.1 从哪里开始	2
1.2 实用的工具	2
1.2.1 语法高亮	3
1.2.2 括号匹配	3
1.2.3 代码折叠	4
1.2.4 自动补全	4
1.3 关注注释	4
1.4 从这里开始	5
1.5 JavaScript 控制台	7
1.6 jQuery	8
1.7 绘制图元	8
1.8 自动执行重复任务	10
1.9 处理用户输入	11
1.10 Hello, World	12
第 2 章 JavaScript 开发中的常用工具	14
2.1 在当下编写 ES6 代码	14
2.2 ES6 的新特性	15
2.2.1 安装 Git	16
2.2.2 终端	16
2.2.3 项目根目录	17
2.2.4 版本控制: Git	17
2.2.5 包管理工具: npm	20
2.2.6 构建工具: Gulp 和 Grunt	22
2.2.7 项目结构	23
2.3 转换器	24
2.3.1 使用 Gulp 运行 Babel	24
2.3.2 格式审查工具	26
2.4 小结	29
第 3 章 字面量、变量、常量和数据类型	32

3.1	变量和常量	32
3.2	变量和常量：用哪个	33
3.3	标识符命名	34
3.4	字面量	35
3.5	基本类型和对象	36
3.6	数字	37
3.7	字符串	39
3.8	特殊字符	40
3.8.1	模板字符串	41
3.8.2	多行字符串	42
3.8.3	数字用作字符串	43
3.9	布尔型	43
3.10	符号	43
3.11	null 和 undefined	44
3.12	对象	44
3.13	Number, String 和 Boolean 对象	47
3.14	数组	47
3.15	对象和数组的拖尾逗号	49
3.16	日期	50
3.17	正则表达式	50
3.18	映射和集合	51
3.19	数据类型转换	51
3.19.1	转换成数字	51
3.19.2	转换成字符串	52
3.19.3	转换成布尔型	52
3.20	小结	53
第 4 章	控制流	54
4.1	控制流的底层	54
4.1.1	while 循环	57
4.1.2	块语句	58
4.1.3	空格	59
4.1.4	辅助方法	60
4.1.5	if else 语句	60
4.1.6	do...while 循环	62
4.1.7	for 循环	63

4.1.8	if 语句	64
4.1.9	最后的整合	65
4.2	JavaScript 中的控制语句	66
4.2.1	控制流异常	67
4.2.2	链式 if...else 语句	67
4.2.3	元语法	68
4.2.4	其他循环模式	69
4.2.5	switch 语句	70
4.2.6	for...in 循环	74
4.2.7	for...of 循环	74
4.3	实用的控制流模式	75
4.3.1	使用 continue 减少条件嵌套	75
4.3.2	使用 break 或 return 避免不必要的计算	75
4.3.3	在循环结束后使用索引的值	76
4.3.4	列表变动时索引递减	76
4.4	小结	77
第 5 章	表达式和运算符	78
5.1	运算符	79
5.2	算术运算符	80
5.3	运算符优先级	82
5.4	比较运算符	83
5.5	比较数字	85
5.6	字符串连接	86
5.7	逻辑运算符	87
5.8	与、或和非	88
5.8.1	短路求值	89
5.8.2	非布尔值的逻辑运算符	89
5.8.3	条件运算符	90
5.8.4	逗号运算符	91
5.9	分组运算符	91
5.9.1	位运算符	91
5.9.2	类型判断运算符	93
5.9.3	void 运算符	94
5.9.4	赋值运算符	94
5.10	解构赋值	95

5.11	对象和数组运算符	97
5.12	模板字符串中的表达式	97
5.13	表达式和控制流模式	97
5.13.1	将 if...else 语句转化成条件表达式	98
5.13.2	将 if 语句转化成短路求值的逻辑或 () 表达式	98
5.14	小结	98
第 6 章	函数	99
6.1	返回值	100
6.2	引用调用	100
6.3	函数参数	101
6.3.1	参数会让函数有所区别吗	103
6.3.2	解构参数	104
6.3.3	默认参数	105
6.4	函数作为对象属性	105
6.5	this 关键字	105
6.6	函数表达式和匿名函数	107
6.7	箭头符号	109
6.8	调用、请求和绑定	110
6.9	小结	112
第 7 章	作用域	113
7.1	作用域和存在	114
7.2	静态作用域与动态作用域	114
7.3	全局作用域	115
7.4	块作用域	117
7.5	变量屏蔽	117
7.6	函数、闭包和静态作用域	119
7.7	即时调用函数表达式	120
7.8	函数作用域和提升	121
7.9	函数提升	123
7.10	临时死区	124
7.11	严格模式	124
7.12	小结	125
第 8 章	数组及其处理	126
8.1	数组概览	126
8.2	操作数组内容	127

8.2.1	在起始和末尾添加或删除元素	128
8.2.2	在末尾添加多个元素	128
8.2.3	获取子数组	128
8.2.4	从任意位置添加或删除元素	129
8.2.5	数组内的分割和替换	129
8.2.6	用指定值填充数组	129
8.2.7	数组反转和排序	130
8.3	数组搜索	130
8.4	数组的基本操作: map 和 filter	133
8.5	数组魔法: reduce	135
8.6	数组方法, 已删除或者未定义的元素	138
8.7	字符串连接	138
8.8	小结	139
第 9 章	对象以及面向对象编程	141
9.1	属性枚举	141
9.1.1	for...in	142
9.1.2	Object.keys	142
9.2	面向对象编程	143
9.2.1	创建类和实例	143
9.2.2	动态属性	145
9.2.3	类即函数	146
9.2.4	原型	147
9.2.5	静态方法	148
9.2.6	继承	149
9.2.7	多态	151
9.2.8	枚举对象属性, 回顾	151
9.2.9	字符串表示	152
9.3	多继承、混合类和接口	153
9.4	小结	155
第 10 章	maps 和 sets	156
10.1	maps	156
10.2	Weak maps	158
10.3	sets	159
10.4	Weak sets	160
10.5	打破对象习惯	161

第 11 章	异常和错误处理	162
11.1	Error 对象	162
11.2	使用 try 和 catch 处理异常	163
11.3	抛出异常	164
11.4	异常处理和调用栈	165
11.5	try...catch... finally	166
11.6	让异常成为例外	167
第 12 章	迭代器和生成器	168
12.1	迭代协议	170
12.2	生成器	172
12.2.1	yield 表达式和双向交流	173
12.2.2	生成器和返回值	175
12.3	小结	175
第 13 章	函数和抽象思考的力量	176
13.1	函数作为子程序	176
13.2	函数作为有返回值的子程序	177
13.3	函数即……函数	178
13.4	那又如何	180
13.5	IIEFs 和异步代码	182
13.6	函数变量	184
13.6.1	数组中的函数	186
13.6.2	将函数传给函数	187
13.6.3	在函数中返回函数	188
13.7	递归	189
13.8	小结	190
第 14 章	异步编程	191
14.1	类比	192
14.2	回调	192
14.2.1	setInterval 和 clearInterval	193
14.2.2	scope 和异步执行	194
14.2.3	错误优先回调	195
14.2.4	回调地狱	196
14.3	promise	197
14.3.1	创建 promise	198
14.3.2	使用 promise	198