

O'REILLY®

TURING

图灵程序设计丛书

深入理解

JavaScript

特性

Practical Modern JavaScript

Dive into ES6 and the Future of JavaScript

[阿根廷] 尼古拉斯·贝瓦夸 著

李松峰 刘冰晶 高峰 译

黄小璐 欧雪雯 审校



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



图灵程序设计丛书

深入理解JavaScript特性

Practical Modern JavaScript

[阿根廷] 尼古拉斯·贝瓦夸 著

李松峰 刘冰晶 高峰 译

黄小璐 欧雪雯 审校

Beijing • Boston • Farnham • Sebastopol • Tokyo



O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

深入理解JavaScript特性 / (阿根廷) 尼古拉斯·贝瓦夸著 ; 李松峰, 刘冰晶, 高峰译. — 北京 : 人民邮电出版社, 2019.5

(图灵程序设计丛书)

ISBN 978-7-115-51040-2

I. ①深… II. ①尼… ②李… ③刘… ④高… 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059529号

内 容 提 要

本书旨在让读者轻松学习 JavaScript 的新进展, 包括 ES6 及后续更新。书中提供了大量实用示例, 以循序渐进的方式讲解了异步函数、对象解构、动态导入及异步生成器等内容。本书不仅介绍了箭头函数、解构、模板字面量以及其他语法方面的新元素, 还全面展示了 ES6 引入的流程控制机制, 以及如何高效地简化自己的代码。本书的讨论还涉及 ES6 内置的新集合类型、使用代理控制属性访问、ES6 中内置 API 的改进、CommonJS 与 ECMAScript 模块的互用性等方面。

本书适合前端开发者、编程爱好者和拥有 JavaScript 实际开发经验的程序员阅读参考。

-
- ◆ 著 [阿根廷] 尼古拉斯·贝瓦夸
 - 译 李松峰 刘冰晶 高 峰
 - 审 校 黄小璐 欧雪雯
 - 责任编辑 温 雪
 - 责任印制 周昇亮

- ◆ 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>
- 北京鑫正大印刷有限公司印刷

- ◆ 开本: 800×1000 1/16
 - 印张: 16.25
 - 字数: 384千字 2019年5月第1版
 - 印数: 1—3 500册 2019年5月北京第1次印刷
 - 著作权合同登记号 图字: 01-2018-8087号
-

定价: 79.00元

读者服务热线: (010)51095183转600 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

译者介绍

李松峰

360前端开发资深专家、前端TC委员、W3C AC代表，奇舞团Web字体服务“奇字库”作者。

刘冰晶

毕业于北京邮电大学，现为奇舞团前端开发工程师，专注于数据可视化以及前端动画领域。

高峰

硕士，毕业于中国科学技术大学软件学院。现为奇虎360前端开发工程师。

审校介绍

黄小璐

毕业于华中科技大学计算机学院。现为奇虎360软件开发工程师。参与翻译了《高性能HTML5》《移动Web手册》《大型JavaScript应用最佳实践指南》《Web开发权威指南》等书。

欧雪雯

前端开发工程师，多年从事科技、前端技术领域业余翻译工作。

站在巨人的肩膀上

Standing on Shoulders of Giants



iTuring.cn

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务还是面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列非凡想法（真希望当初我也想到了）建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

本书赞誉

“本书全面介绍了 ES6 新特性的语法和语义，有助于你大幅度提升代码的表达能力。作者把这些特性融入简单易懂的示例中，帮你快速上手。”

——Kent C. Dodds, PayPal 前端工程师, TC39 成员

“作者对重新定义现代 JavaScript 的诸多特性进行了深入彻底又贴合实际的解读。”

——Rod Vagg, require.io 技术顾问, Node.js TSC 成员

“原本复杂难懂的技术，被作者巧妙地转化为通俗易懂的文字和示例代码。”

——Mathias Bynens, Google 前端工程师, TC39 成员

“真正掌握 JavaScript 很难，2015 版规范又增加了很多新特性。本书化整为零，分别介绍了这些特性的使用场景、概念模型和最佳实践，而且示例很简单，因此掌握起来就容易多了。”

——Jordan Harband, Airbnb 软件工程师, TC39 成员

“ES6 为这门语言新增了很多特性，即使是富有经验的开发者也要花很多时间去学习掌握。在此过程中，我们需要一个向导，而本书是明智之选。”

——Ingvar Stepanyan, CloudFlare 前端工程师

“1995 年，我在发明 JavaScript 的时候，并没有想到它日后能成为互联网上使用最广的编程语言。本书与我循序渐进和直观明了的教学理念不谋而合。建议你好好读一读，从中发现对自己有用的东西，进而真正拥抱 JavaScript，最终致力于为所有人开发更好的 Web 应用。”

——Brendan Eich, JavaScript 之父

序

1995 年，我在 Netscape 发明 JavaScript 的时候，并没有想到它日后能成为互联网上使用最广的编程语言。我只知道自己时间有限，要尽快写出一个“最小可发布的”版本。正因为如此，我将它写得尽量可扩展，自全局对象以下都可以修改，甚至连基础级的元对象协议钩子（如与 Java 方法同名的 toString 和 valueOf）也可以修改。

抛开其不断演进和越来越流行不说，JavaScript 始终受益于一种递进的、务实的教学方法，即“要事先行”。我认为这与匆忙设计和故意为之的可扩展性有着必然的联系。我重度使用了函数和对象这两个核心元素。因此，程序员可以将它们作为通用构造，花式地构建各种工具，最终造出一把锋利的“瑞士军刀”。而对学生来说，这意味着他们必须了解各种工具都能解决哪些任务，以及如何恰如其分地运用这些工具。

我感觉 Netscape 就像一阵旋风，相信 1995 年的那批同事也都有同感。那一年，Marc Andreessen 在路演时反复强调“Netscape+Java 会灭掉 Windows”，借与微软竞争而迅速实现了 IPO。Java 是正统编程语言、老大哥、“蝙蝠侠”，而 JavaScript 只是一个小兄弟、“助手罗宾”、小小的脚本语言。

但在编写第一版（代号 Mocha）时我就知道，JavaScript（而不是 Java）会与 Netscape 浏览器以及我同时创造的文档对象模型深度绑定。Netscape 与 Sun 或者浏览器与 JVM 的界限是无法跨越的，Java 只能作为插件嵌入。

因此我当时就有一种预感，JavaScript 要么会随时间的推移慢慢走向成功，要么会因为某些原因迅速消亡。记得我的好友兼室友 Jeff Weinstein 问我后 20 年干什么时，我说：“要么开发 JavaScript，要么完蛋。”那时，我就感觉好像欠了 JavaScript 用户一屁股债。这种感觉源于我在极短时间和管理层严令“看起来要像 Java”的双重限制下，选择了“双面瑞士军刀”这种设计。

“模块化 JavaScript” (Modular JavaScript Series) 这套书与我循序渐进和直观明了的教学理念不谋而合，先从浅显易懂的代码示例讲起，逐步扩展到设计模式，再到完整的基于模块的应用构建。这套书匠心独运，专门探讨了有关测试的最佳实践和部署 JavaScript 应用的高超技术。说这套书是 O'Reilly 出版公司“JavaScript 图书”这顶王冠上又一颗璀璨的明珠一点也不为过。

看到尼古拉斯为此付出的努力，我感到非常欣慰，因为这本书一看就能给 JavaScript 程序员耳目一新的感觉。我第一次见到尼古拉斯是在巴黎的一次晚宴上，当时对他有了一点了解，之后就是线上交流。他写的东西特别实用，而且能够体谅 JavaScript 初学者，同时也非常幽默。这也促使我审阅了这本书的草稿。相信最终付梓时，拿到手的成品会更容易阅读，也更有意思。建议你好好读一读这本书，从中发现对自己有用的东西，进而真正拥抱 JavaScript，最终致力于为所有人开发更好的 Web 应用。

Brendan Eich

JavaScript 之父，Brave 软件公司 CEO 和联合创始人

前言

时间倒回 1998 年，那时我在学校用 FrontPage 做网页。如果当时有人告诉我，将来我会以 Web 开发为业，我一定会笑出声来。之后数年，JavaScript 一直向前发展，很难想象 Web 离开 JavaScript 是否还能像现在这样繁荣。本书将一点一点地勾勒出一幅完整的现代 JavaScript 图景。

读者对象

本书主要写给 Web（前端）开发者、编程爱好者和拥有 JavaScript 实际开发经验的程序员。这些人以及想要进一步了解 JavaScript 语言的任何人都可以通过阅读本书受益。

为什么编写本书

本书旨在让你轻松学习 JavaScript 的最新进展，包括 ES6 及后续更新。ES6 是这门语言一个里程碑式的版本，它几乎与简化的规范制定流程同时发布。当时我就围绕 ES6 的不同特性写了不少博文，受到了很多读者的喜爱。关于 ES6 的书也不少，但这些书与我心目中的关于 ES6 及其未来的书略有不同。本书在详细介绍 ES6 的新特性时，不会陷入规范、实现细节，也不会探讨那些几乎不可能遇到的边界情形（如果这些情况出现，肯定是要在网上查找相关资料的）。

本书不贪大求全，而是专注于循序渐进的学习过程，确保先学内容可以成为后学内容的基础，避免你为寻找某个定义翻来翻去。本书配备了大量实用示例，不仅涉及 ES6，还涉及 2015 年 6 月（ES6 规范定稿时间）以后的其他变化，包括异步函数、对象解构、动态导入、Promise#finally 和异步生成器。

本书的目标是确保你能够顺利继续学习本系列图书的后续其他分册。在这第一本书的基础上，我们将接着探讨模块化设计、测试和部署，到时就不必再细究代码示例中用到的语法

特性了。这种递进和模块化的学习方式体现在了整个系列中，包括每一本书、每一章，甚至每一节。

本书内容

第 1 章简单介绍了 JavaScript 及其标准化过程，回顾其发展，概述其现状，并展望其未来。这一章也会简单介绍 Babel 和 ESLint 这两个现代 JavaScript 开发必备的工具。

第 2 章介绍了 ES6 中最基础的部分，包括箭头函数、解构、let 和 const、模板字面量以及其他语法方面的新东西。

第 3 章讨论了用于声明对象原型的 class 语法、名为 Symbol 的新原始类型，以及 Object 对象的新方法。

第 4 章全面展示了 ES6 引入的流程控制机制。先从 Promise 开始，然后是迭代器、生成器和异步函数，这一章会详尽讨论每个主题并配有丰富的实例，揭示所有这些特性之间的协同关系。这一章不只是为了让你学会使用它们，更希望你能真正理解它们的最佳用途，从而简化自己的代码。

第 5 章介绍了 ES6 内置的新集合类型 Set 和 Map，Set 用于创建包含唯一值的集合，Map 用于创建对象映射。这一章也会给出实际的应用实例。

第 6 章展示了新的 Proxy 和 Reflect 内置特性，不仅会介绍如何使用代理，还会探讨为什么使用它们时要谨慎。

第 7 章主要讨论了 ES6 中的其他改进，特别是与 Array、Math、数值、字符串、Unicode 和正则表达式相关的改进。

第 8 章讨论了原生 JavaScript 模块，并简单介绍了其诞生的历史，最后详细介绍了其语法。

第 9 章的标题是“实用建议”，这与其他编程语言方面的图书有所不同。我没有将自己的这些建议放到各个章节，而是将它们汇编到一章。你将在这一章中看到什么时候该用哪种变量声明方式或字面量引号，异步代码流的控制，以及使用类和代理到底好不好之类的建议和思考。

如果你已经比较熟悉 ES6 了，那么建议你认真阅读第 4 章，其中关于流程控制的部分非常有价值。第 7 章和第 8 章也是必读的，因为其中的内容很难在别处看到。最后一章无疑会激发你的思考：在模块化 JavaScript 这个新领域中，什么样的做法合适，什么样的做法不可取？无论是否同意我的观点，你都会从中受益颇多。

排版约定

本书使用了下列排版约定。

- **黑体**

表示新术语和重点强调的内容。

- 等宽字体 (constant width)

表示程序片段，以及正文中出现的程序元素，比如变量、函数名、数据库、数据类型、环境变量、语句和关键字等。



该图标表示提示或建议。



该图标表示警告或警示。

O'Reilly Safari



Safari（前身为 Safari Books Online）是一个会员制的培训和参考平台，面向企业、政府、教育从业者和个人。

Safari 用户可访问 O'Reilly Media、Harvard Business Review、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、Course Technology 等 250 多家出版社的上千种图书、培训视频、学习路径、交互式教程和精选播放列表。

如需了解更多信息，请访问 <http://oreilly.com/safari>。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息¹。本书的网站地址是：<http://www.oreilly.com/catalog/0636920047124>。

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：

<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：

<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：

<http://www.youtube.com/oreillymedia>

致谢

人类做的任何事都是以其他人的成果为基础的，本书也不例外。感谢 O'Reilly 公司的编辑。首先是 Nan Barber，她对我创作本书给予了极大的支持。还有 Ally MacDonald，她启发我以模块化方式来写这套书，而且帮助我克服了项目初期的困难。

本书的技术审校阵容可谓豪华。其中很多人都来自 TC39，也就是制定 JavaScript 标准的委员会，他们抽出宝贵的时间进行审阅，使本书的内容有了保证。Mathias Bynens（曾任职于 Opera）帮忙校对了本书中与 Unicode 标准相关的所有内容，确保了书中代码示例的高度一致。Kent C. Dodds（TC39 成员，PayPal）创造性地进行了视频评审，指出了一些问

注 1：读者可访问本书图灵社区页面（<http://www.ituring.com.cn/book/2452>）下载示例代码并提交中文版勘误。

——编者注

题，进行了质量把关。Jordan Harband（TC39 成员，Airbnb）对书中讨论的很多 JavaScript 特性给出了技术意见。Alex Russell（TC39 成员，Google）帮助厘清了 JavaScript 的历史及有关标准主体的内容。Ingvar Stepanyan（Cloudflare）也是一位从代码中挑错的高手，帮我纠正了与规范底层细节相关的错误。Brian Terlson（TC39 编辑，Microsoft）也就 TC39 的时间安排和相关细节提供了解答。Rod Vagg（Node.js 项目）对本书代码示例的风格统一和更贴近主题提供了帮助。

Brendan Eich（TC39 成员，Brave CEO）分享了关于 JavaScript 早期和 TC39 的很多见闻，这为第一章的内容打下了基础。如果没有他，也就不会有你手中这本书。

最后，我要感谢我的妻子 Marianela，感谢她在我撰写本书期间做出的牺牲和对我的宽容。Marian，没有你，我不可能完成本书！

电子书

扫描如下二维码，即可购买本书电子版。



目录

序	xi
前言	xiii
第 1 章 ECMAScript 和 JavaScript 的未来	1
1.1 JavaScript 标准简史	1
1.2 持续迭代的 ECMAScript	3
1.3 浏览器支持和辅助工具	5
1.3.1 Babel 转译器简介	5
1.3.2 使用 ESLint 提高代码质量和一致性	9
1.4 ES6 特性	11
1.5 JavaScript 的未来	12
第 2 章 ES6 基础	14
2.1 对象字面量	14
2.1.1 属性值简写	14
2.1.2 可计算属性名	15
2.1.3 方法定义	17
2.2 箭头函数	18
2.2.1 词法作用域	19
2.2.2 箭头函数的写法	20
2.2.3 优点和用例	21
2.3 解构	22
2.3.1 对象的解构	22
2.3.2 数组的解构	25
2.3.3 函数参数的默认值	25

2.3.4	函数参数的解构	26
2.3.5	解构的用例	28
2.4	剩余参数和扩展运算符	29
2.4.1	剩余参数	29
2.4.2	扩展运算符	30
2.5	模板字面量	32
2.5.1	字符串插值	32
2.5.2	多行模板字面量	33
2.5.3	标签模板	35
2.6	let 和 const 声明	37
2.6.1	块级作用域和 let 声明	38
2.6.2	暂时性死区	39
2.6.3	const 声明	41
2.6.4	const 和 let 的优势	42
第 3 章 类、符号、对象和装饰器		44
3.1	类	44
3.1.1	使用类	44
3.1.2	类的属性和方法	47
3.1.3	类的继承	49
3.2	符号	51
3.2.1	本地符号	51
3.2.2	符号的实际用法	53
3.2.3	全局符号注册表	55
3.2.4	众所周知的符号	57
3.3	对象的内置改进	58
3.3.1	使用 Object.assign 扩展对象	58
3.3.2	使用 Object.is 进行对象比较	61
3.3.3	Object.setPrototypeOf	62
3.4	装饰器	63
3.4.1	初识 JavaScript 装饰器	63
3.4.2	装饰器叠加及不变性提醒	64
3.4.3	用例: C# 中的特性	64
3.4.4	在 JavaScript 中装饰属性	66
第 4 章 迭代与流程控制		68
4.1	Promise	68
4.1.1	快速理解 Promise	68

4.1.2	Promise 的延续与连缀	72
4.1.3	创建 Promise	76
4.1.4	Promise 的状态	78
4.1.5	Promise#finally 提案	79
4.1.6	Promise.all 和 Promise.race	81
4.2	迭代器协议与可迭代协议	83
4.2.1	迭代的原理	83
4.2.2	无穷序列	85
4.2.3	迭代对象以生成键/值对	88
4.2.4	打造多功能播放列表	90
4.3	生成器函数与生成器对象	93
4.3.1	生成器基础	93
4.3.2	手工迭代生成器	96
4.3.3	将生成器混入可迭代对象	97
4.3.4	使用生成器遍历树	99
4.3.5	传递生成器函数	101
4.3.6	处理异步流	102
4.3.7	在生成器上抛出错误	103
4.3.8	代替生成器返回	104
4.3.9	基于生成器的异步 I/O	105
4.4	异步函数	109
4.4.1	各种异步代码	109
4.4.2	使用 async/await	111
4.4.3	并发异步流	112
4.4.4	错误处理	113
4.4.5	深入理解异步函数	114
4.5	异步迭代	119
4.5.1	异步迭代器	119
4.5.2	异步生成器	120
第 5 章	巧妙使用 ES 中的集合	122
5.1	使用 ES6 map	124
5.1.1	初识 ES6 map	124
5.1.2	散列映射和 DOM 元素	128
5.2	理解和使用 WeakMap	130
5.3	ES6 中的 Set	131
5.4	ES6 WeakSet	133