

O'REILLY®

深入挖掘JavaScript语言本质，简练形象地
解释抽象概念，打通JavaScript的任督二脉

TURING

图灵程序设计丛书

[美] KYLE SIMPSON 著
单业 姜南 译

你不知道的 JavaScript 中卷

TYPES & GRAMMAR
ASYNC & PERFORMANCE

YOU DON'T KNOW
JS



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

你不知道的JavaScript（中卷）

You Don't Know JavaScript: Types &
Grammar, Async & Performance



O'REILLY®

Beijing • Cambridge • Farnham • Köln • 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 (CIP) 数据

你不知道的JavaScript. 中卷 / (美) 辛普森
(Kyle Simpson) 著 ; 单业, 姜南译. — 北京 : 人民邮
电出版社, 2016. 8

(图灵程序设计丛书)

ISBN 978-7-115-43116-5

I. ①你… II. ①辛… ②单… ③姜… III. ①JAVA语
言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第174640号

内 容 提 要

JavaScript 这门语言简单易用, 很容易上手, 但其语言机制复杂微妙, 即使是经验丰富的 JavaScript 开发人员, 如果没有认真学习的话也无法真正理解。本套书直面当前 JavaScript 开发人员不求甚解的大趋势, 深入理解语言内部的机制, 全面介绍了 JavaScript 中常被他人误解和忽视的重要知识点。本书是其中卷, 主要介绍了类型、语法、异步和性能。

本书既适合 JavaScript 语言初学者了解其精髓, 又适合经验丰富的 JavaScript 开发人员深入学习。

-
- ◆ 著 [美] Kyle Simpson
 - 译 单业 姜南
 - 责任编辑 朱巍
 - 执行编辑 贺子娟 占亚娥
 - 责任印制 彭志环
 - ◆ 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>
 - 北京市昌平百善印刷厂印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 23.5
 - 字数: 555千字 2016年8月第1版
 - 印数: 1-4 000册 2016年8月北京第1次印刷
 - 著作权合同登记号 图字: 01-2016-4629号

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

版权声明

© 2015 by Kyle Simpson.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2016. Authorized translation of the English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2015。

简体中文版由人民邮电出版社出版，2016。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

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、在线服务或者面授课程，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（真希望当初我也想到了）非凡想法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

前言

JavaScript 从互联网萌芽时期开始就一直是实现交互体验的基本技术。虽然最初被用来实现闪烁的鼠标轨迹和烦人的弹出消息框，但在大约二十年以后，它在技术和功能方面都得到了很大的提升，几乎没有人再质疑它在互联网中的重要地位。

但是，作为一门编程语言，JavaScript 一直为人诟病，部分原因是其历史沿革，更重要的原因则是其设计理念。因为 JavaScript 这个名字，Brendan Eich 曾戏称它为“傻小弟”（相对于成熟的 Java 而言）。实际上，这个名字完全是政治和市场考量下的产物。两门语言之间千差万别，“JavaScript”之于“Java”就如同“Carnival”（嘉年华）之于“Car”（汽车）一样，两者之间并无半点关系。

JavaScript 在概念和语法风格上借鉴了其他编程语言，包括 C 风格的过程式编程和隐晦的 Scheme/Lisp 风格的函数式编程，这使得它能为不同背景的开发人员所接受，包括那些没有多少编程经验的人。用 JavaScript 编写一个“Hello World”程序非常简单。

JavaScript 可能是最容易上手的编程语言之一，但它的一些奇特之处使得它不像其他语言那样容易完全掌握。要想用 C 或者 C++ 开发一个完整的应用程序，开发者需要对该门语言有相当深入的了解。然而对于 JavaScript，即使我们用它开发了一个完整的系统也不见得就能深入理解它。

这门语言中有些复杂的概念隐藏得很深，却常常以一种看似简单的形式呈现。例如，将函数作为回调函数传递，这让 JavaScript 开发人员往往满足于使用这些现成便利的机制，而不愿去探究其中的原理。

JavaScript 是一门简单易用的语言，应用广泛，同时它的语言机制又十分复杂和微妙，即使经验丰富的开发人员也需要用心学习才能真正掌握。

JavaScript 的矛盾之处就在于此，它的阿喀琉斯之踵正是本书要解决的问题。因为无需深入理解就能用它来编程，所以人们常常放松对它的学习。

使命

在学习 JavaScript 的过程中，碰到令人抓狂的问题或挫折时，如果置之不理或不求甚解（就像有些人习惯做的那样），我们很快就会发现根本无从发挥这门语言的威力。

尽管这些被称为 JavaScript 的“精华”部分，但我恳请读者朋友们将其看作“容易的”“安全的”或者“不完整的”部分。

“你不知道的 JavaScript”系列丛书旨在介绍 JavaScript 的另一面，让你深入掌握 JavaScript 的全部，特别是那些难点。

JavaScript 开发人员常常满足于知半解，不愿更深入地了解其深层原因和运作方式，本书要解决的正是这个问题。我们会直面那些疑难困惑，绝不回避。

我个人不会仅仅满足于让代码运行起来而不明就里，你也应该这样。本书中，我会逐步介绍 JavaScript 中那些不太为人所知的地方，最终让你对这门语言有一个全面的了解。一旦掌握了这些知识，那些技巧、框架和时髦术语等都将不在话下。

本系列丛书全面深入地介绍了 JavaScript 中常为人误解和忽视的重要知识点，让你在读完之后不论从理论上还是实践上都能对这门语言有足够的信心。

目前你对 JavaScript 的了解可能都来自那些自身就一知半解的“专家”，而这仅仅是冰山一角。读完本系列丛书后，你将真正了解这门语言。现在就让我们踏上阅读寻知之旅吧。

小结

JavaScript 是一门优秀的语言。只学其中一部分内容很容易，但是要全面掌握则很难。开发人员遇到困难时往往将其归咎于语言本身，而不反省他们自己对语言的理解有多匮乏。本系列丛书旨在解决这个问题，使读者能够发自内心地喜欢上这门语言。



本书中的很多示例都假定你使用的是现代（以及未来）的 JavaScript 引擎环境，比如 ES6。有些代码在旧版本（ES6 之前）的引擎下可能不会像本书中描述的那样工作。

排版约定

本书使用了下列排版约定。

- 楷体
表示新术语。

- 等宽字体 (`constant width`)
表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。
- 加粗等宽字体 (`constant width bold`)
表示应该由用户输入的命令或其他文本。
- 等宽斜体 (`constant width italic`)
表示应该由用户输入的值或根据上下文确定的值替换的文本。



该图标表示提示或建议。



该图标表示一般笔记。



该图标表示警告或警示。

使用代码示例

补充材料（代码示例、练习等）可以从 <https://github.com/getify/You-Dont-Know-JS/tree/master/types%20&%20grammar> 和 [https://github.com/getify/You-Dont-Know-JS/tree/master/async & performance](https://github.com/getify/You-Dont-Know-JS/tree/master/async%20&%20performance) 下载。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无需联系我们获得许可。比如，用本书的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN，比如：“*You Don't Know JavaScript: Types & Grammar* by Kyle Simpson (O'Reilly). Copyright 2015 Getify Solutions, Inc., 978-1-491-90419-0”。

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

Safari® Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学习和认证培训时，都将 Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 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、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

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。本书第一部分“类型和语法”的网站地址是 <http://shop.oreilly.com/product/0636920033745.do>。本书第二部分“异步和性能”的网址是 <http://shop.oreilly.com/product/0636920033752.do>。

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

致谢

我要感谢很多人，是他们的帮助让本书以及整个系列得以出版。

首先，我要感谢我的妻子 Christen Simpson 以及我的两个孩子 Ethan 和 Emily，容忍我整天坐在电脑前工作。即使不写作的时候，我的眼睛也总是盯着屏幕做一些与 JavaScript 相关的工作。我牺牲了很多陪伴家人的时间，这个系列的丛书才得以让读者深入全面地介绍 JavaScript。对于家庭，我亏欠太多。

我要感谢 O'Reilly 的编辑 Simon St.Laurent 和 Brian MacDonald，以及所有其他的编辑和市场工作人员。和他们一起工作非常愉快；本系列丛书的写作、编辑和制作都以开源方式进行，在此实验过程中，他们给予了非常多的帮助。

我要感谢所有为本系列丛书提供建议和校正的人，包括 Shelley Powers、Tim Ferro、Evan Borden、Forrest L. Norvell、Jennifer Davis、Jesse Harlin 等。十分感谢 David Walsh 和 Jake Archibald 为本书作序。

我要感谢 JavaScript 社区中的许多人，包括 TC39 委员会的成员们，将他们的知识与我们分享，并且耐心详尽地回答我无休止的提问。他们是 John-David Dalton、Juriy “kangax” Zaytsev、Mathias Bynens、Rick Waldron、Axel Rauschmayer、Nicholas Zakas、Angus Croll、Jordan Harband、Reginald Braithwaite、Dave Herman、Brendan Eich、Allen Wirfs-Brock、Bradley Meck、Domenic Denicola、David Walsh、Tim Disney、Kris Kowal、Peter van der Zee、Andrea Giammarchi、Kit Cambridge，等等。还有很多人，我无法一一感谢。

“你不知道的 JavaScript” 系列丛书是由 Kickstarter 发起的，我要感谢近 500 名慷慨的支持者，没有他们的支持就没有这套系列丛书：

Jan Szpila、nokiko、Murali Krishnamoorthy、Ryan Joy、Craig Patchett、pdqtrader、Dale Fukami、ray hatfield、R0drigo Perez [Mx]、Dan Petitt、Jack Franklin、Andrew Berry、Brian Grinstead、Rob Sutherland、Sergi Meseguer、Phillip Gourley、Mark Watson、Jeff Carouth、Alfredo Sumaran、Martin Sachse、Marcio Barrios、Dan、AimelyneM、Matt Sullivan、

Delnatte Pierre-Antoine, Jake Smith, Eugen Tudorancea, Iris, David Trinh, simonstl, Ray Daly, Uros Gruber, Justin Myers, Shai Zonis, Mom & Dad, Devin Clark, Dennis Palmer, Brian Panahi Johnson, Josh Marshall, Marshall, Dennis Kerr, Matt Steele, Erik Slagter, Sacah, Justin Rainbow, Christian Nilsson, Delapouite, D.Pereira, Nicolas Hoizey, George V. Reilly, Dan Reeves, Bruno Laturner, Chad Jennings, Shane King, Jeremiah Lee Cohick, od3n, Stan Yamane, Marko Vucinic, Jim B, Stephen Collins, Ægir Þorsteinsson, Eric Pederson, Owain, Nathan Smith, Jeanetteurphy, Alexandre ELISÉ, Chris Peterson, Rik Watson, Luke Matthews, Justin Lowery, Morten Nielsen, Vernon Kesner, Chetan Shenoy, Paul Tregoin, Marc Grabanski, Dion Almaer, Andrew Sullivan, Keith Elsass, Tom Burke, Brian Ashenfelter, David Stuart, Karl Swedberg, Graeme, Brandon Hays, John Christopher, Gior, manoj reddy, Chad Smith, Jared Harbour, Minoru TODA, Chris Wigley, Daniel Mee, Mike, Handyface, Alex Jahraus, Carl Furrow, Rob Foulkrod, Max Shishkin, Leigh Penny Jr., Robert Ferguson, Mike van Hoenselaar, Hasse Schougaard, rajan venkataguru, Jeff Adams, Trae Robbins, Rolf Langenhuijzen, Jorge Antunes, Alex Koloskov, Hugh Greenish, Tim Jones, Jose Ochoa, Michael Brennan-White, Naga Harish Muvva, Barkóczi Dávid, Kitt Hodsdon, Paul McGraw, Sascha Goldhofer, Andrew Metcalf, Markus Krogh, Michael Mathews, Matt Jared, Juanfran, Georgie Kirschner, Kenny Lee, Ted Zhang, Amit Pahwa, Inbal Sinai, Dan Raine, Schabse Laks, Michael Tervoort, Alexandre Abreu, Alan Joseph Williams, NicolasD, Cindy Wong, Reg Braithwaite, LocalPCGuy, Jon Friskies, Chris Merriman, John Pena, Jacob Katz, Sue Lockwood, Magnus Johansson, Jeremy Crapsey, Grzegorz Pawłowski, nico nuzzaci, Christine Wilks, Hans Bergren, charles montgomery, Ariel Fogel, Ivan Kolev, Daniel Campos, Hugh Wood, Christian Bradford, Frédéric Harper, Ionuț Dan Popa, Jeff Trimble, Rupert Wood, Trey Carrico, Pancho Lopez, Joël kuyten, Tom A Marra, Jeff Jewiss, Jacob Rios, Paolo Di Stefano, Soledad Penades, Chris Gerber, Andrey Dolganov, Wil Moore III, Thomas Martineau, Kareem, Ben Thouret, Udi Nir, Morgan Laupies, jory carson-burson, Nathan L Smith, Eric Damon Walters, Derry Lozano-Hoyland, Geoffrey Wiseman, mkeehner, KatieK, Scott MacFarlane, Brian LaShomb, Adrien Mas, christopher ross, Ian Littman, Dan Atkinson, Elliot Jobe, Nick Dozier, Peter Wooley, John Hoover, dan, Martin A. Jackson, Héctor Fernando Hurtado, andy enamorato, Paul Seltmann, Melissa Gore, Dave Pollard, Jack Smith, Philip Da Silva, Guy Israeli, @megalithic, Damian Crawford, Felix Gliesche, April Carter Grant, Heidi, jim tierney, Andrea Giammarchi, Nico Vignola, Don Jones, Chris Hartjes, Alex Howes, john gibbon, David J. Groom, BBox, Yu Dilys Sun, Nate Steiner, Brandon Satrom, Brian Wyant, Wesley Hales, Ian Pouncey, Timothy Kevin Oxley, George Terezakis, sanjay raj, Jordan Harband, Marko McLion, Wolfgang Kaufmann, Pascal Peuckert, Dave Nugent, Markus Liebelt, Welling Guzman, Nick Cooley, Daniel Mesquita, Robert Syvarth, Chris Coyier, Rémy Bach, Adam Dougal, Alistair

Duggin, David Loidolt, Ed Richer, Brian Chenault, GoldFire Studios, Carles Andrés, Carlos Cabo, Yuya Saito, roberto ricardo, Barnett Klane, Mike Moore, Kevin Marx, Justin Love, Joe Taylor, Paul Dijou, Michael Kohler, Rob Cassie, Mike Tierney, Cody Leroy Lindley, tofuji, Shimon Schwartz, Raymond, Luc De Brouwer, David Hayes, Rhys Brett-Bowen, Dmitry, Aziz Khoury, Dean, Scott Tolinski - Level Up, Clement Boirie, Djordje Lukic, Anton Kotenko, Rafael Corral, Philip Hurwitz, Jonathan Pidgeon, Jason Campbell, Joseph C., SwiftOne, Jan Hohner, Derick Bailey, getify, Daniel Cousineau, Chris Charlton, Eric Turner, David Turner, Joël Galerán, Dharma Vagabond, adam, Dirk van Bergen, dave ♡🎵★ furf, Vedran Zakanj, Ryan McAllen, Natalie Patrice Tucker, Eric J. Bivona, Adam Spooner, Aaron Cavano, Kelly Packer, Eric J, Martin Drenovac, Emilis, Michael Pelikan, Scott F. Walter, Josh Freeman, Brandon Hudgeons, vijay chennupati, Bill Glennon, Robin R., Troy Forster, otaku_coder, Brad, Scott, Frederick Ostrander, Adam Brill, Seb Flippence, Michael Anderson, Jacob, Adam Randlett, Standard, Joshua Clanton, Sebastian Kouba, Chris Deck, SwordFire, Hannes Papenberg, Richard Woerber, hnzz, Rob Crowther, Jediah Broadbent, Sergey Chernyshev, Jay-Ar Jamon, Ben Combee, luciano bonachela, Mark Tomlinson, Kit Cambridge, Michael Melgares, Jacob Adams, Adrian Bruinhout, Bev Wieber, Scott Puleo, Thomas Herzog, April Leone, Daniel Mizieliński, Kees van Ginkel, Jon Abrams, Erwin Heiser, Avi Laviad, David newell, Jean-Francois Turcot, Niko Roberts, Erik Dana, Charles Neill, Aaron Holmes, Grzegorz Ziółkowski, Nathan Youngman, Timothy, Jacob Mather, Michael Allan, Mohit Seth, Ryan Ewing, Benjamin Van Treese, Marcelo Santos, Denis Wolf, Phil Keys, Chris Yung, Timo Tjihof, Martin Lekvall, Agendine, Greg Whitworth, Helen Humphrey, Dougal Campbell, Johannes Harth, Bruno Girin, Brian Hough, Darren Newton, Craig McPheat, Olivier Tille, Dennis Roethig, Mathias Bynens, Brendan Stromberger, sundeep, John Meyer, Ron Male, John F Croston III, gigante, Carl Berghem, B.J. May, Rebekah Tyler, Ted Foxberry, Jordan Reese, Terry Sutor, afeliz, Tom Kiefer, Darragh Duffy, Kevin Vanderbeken, Andy Pearson, Simon Mac Donald, Abid Din, Chris Joel, Tomas Theunissen, David Dick, Paul Grock, Brandon Wood, John Weis, dgrebb, Nick Jenkins, Chuck Lane, Johnny Megahan, marzsmán, Tatu Tamminen, Geoffrey Knauth, Alexander Tarmolov, Jeremy Tymes, Chad Auld, Sean Parmelee, Rob Staenke, Dan Bender, Yannick derwa, Joshua Jones, Geert Plaisier, Tom LeZotte, Christen Simpson, Stefan Bruvik, Justin Falcone, Carlos Santana, Michael Weiss, Pablo Villoslada, Peter deHaan, Dimitris Iliopoulos, seyDoggy, Adam Jordens, Noah Kantrowitz, Amol M, Matthew Winnard, Dirk Ginader, Phinam Bui, David Rapson, Andrew Baxter, Florian Bougel, Michael George, Alban Escalier, Daniel Sellers, Sasha Rudan, John Green, Robert Kowalski, David I. Teixeira (@ditma, Charles Carpenter, Justin Yost, Sam S, Denis Ciccale, Kevin Sheurs, Yannick Croissant, Pau Fracés, Stephen McGowan, Shawn Searcy, Chris Ruppel, Kevin Lamping, Jessica Campbell, Christopher

Schmitt、Sablons、Jonathan Reisdorf、Bunni Gek、Teddy Huff、Michael Mullany、Michael Fürstenberg、Carl Henderson、Rick Yoesting、Scott Nichols、Hernán Ciudad、Andrew Maier、Mike Stapp、Jesse Shawl、Sérgio Lopes、jsulak、Shawn Price、Joel Clermont、Chris Ridmann、Sean Timm、Jason Finch、Aiden Montgomery、Elijah Manor、Derek Gathright、Jesse Harlin、Dillon Curry、Courtney Myers、Diego Cadenas、Arne de Bree、João Paulo Dubas、James Taylor、Philipp Kraeutli、Mihai Păun、Sam Gharegozlou、joshjs、Matt Murchison、Eric Windham、Timo Behrmann、Andrew Hall、joshua price、Théophile Villard。

这套系列丛书的写作、编辑和制作都是以开源的方式进行的。我们要感谢 GitHub 让这一切成为可能！

再次向我没能提及的支持者们表示感谢。这套系列丛书属于我们每一个人，希望它能够帮助更多的人更好地了解 JavaScript，让当前和未来的社区贡献者受益。

目录

前言	XI
----	----

第一部分 类型和语法

序	2
第 1 章 类型	3
1.1 类型	4
1.2 内置类型	4
1.3 值和类型	6
1.3.1 undefined 和 undeclared	6
1.3.2 typeof Undeclared	7
1.4 小结	10
第 2 章 值	11
2.1 数组	11
2.2 字符串	13
2.3 数字	15
2.3.1 数字的语法	16
2.3.2 较小的数值	18
2.3.3 整数的安全范围	19
2.3.4 整数检测	20
2.3.5 32 位有符号整数	20
2.4 特殊数值	21
2.4.1 不是值的值	21

2.4.2	undefined	21
2.4.3	特殊的数字	23
2.4.4	特殊等式	27
2.5	值和引用	28
2.6	小结	31
第3章	原生函数	33
3.1	内部属性 [[Class]]	34
3.2	封装对象包装	35
3.3	拆封	36
3.4	原生函数作为构造函数	37
3.4.1	Array(..)	37
3.4.2	Object(..)、Function(..) 和 RegExp(..)	40
3.4.3	Date(..) 和 Error(..)	41
3.4.4	Symbol(..)	42
3.4.5	原生原型	43
3.5	小结	45
第4章	强制类型转换	46
4.1	值类型转换	46
4.2	抽象值操作	47
4.2.1	Tostring	48
4.2.2	ToNumber	52
4.2.3	ToBoolean	53
4.3	显式强制类型转换	56
4.3.1	字符串和数字之间的显式转换	57
4.3.2	显式解析数字字符串	62
4.3.3	显式转换为布尔值	65
4.4	隐式强制类型转换	67
4.4.1	隐式地简化	67
4.4.2	字符串和数字之间的隐式强制类型转换	68
4.4.3	布尔值到数字的隐式强制类型转换	71
4.4.4	隐式强制类型转换为布尔值	72
4.4.5	和 &&	73
4.4.6	符号的强制类型转换	76
4.5	宽松相等和严格相等	77
4.5.1	相等比较操作的性能	77
4.5.2	抽象相等	78
4.5.3	比较少见的情况	83
4.6	抽象关系比较	89
4.7	小结	91

第 5 章 语法	92
5.1 语句和表达式	92
5.1.1 语句的结果值	93
5.1.2 表达式的副作用	95
5.1.3 上下文规则	99
5.2 运算符优先级	104
5.2.1 短路	107
5.2.2 更强的绑定	107
5.2.3 关联	108
5.2.4 释疑	110
5.3 自动分号	111
5.4 错误	113
5.5 函数参数	115
5.6 try..finally	117
5.7 switch	120
5.8 小结	122
附录 A 混合环境 JavaScript	123

第二部分 异步和性能

序	136
第 1 章 异步：现在与将来	138
1.1 分块的程序	139
1.2 事件循环	141
1.3 并行线程	143
1.4 并发	148
1.4.1 非交互	150
1.4.2 交互	150
1.4.3 协作	154
1.5 任务	156
1.6 语句顺序	157
1.7 小结	159
第 2 章 回调	161
2.1 continuation	162
2.2 顺序的大脑	163
2.2.1 执行与计划	164
2.2.2 嵌套回调与链式回调	165

2.3	信任问题	169
2.3.1	五个回调的故事	170
2.3.2	不只是别人的代码	171
2.4	省点回调	173
2.5	小结	176
第3章	Promise	178
3.1	什么是 Promise	179
3.1.1	未来值	179
3.1.2	完成事件	183
3.2	具有 then 方法的鸭子类型	188
3.3	Promise 信任问题	190
3.3.1	调用过早	190
3.3.2	调用过晚	191
3.3.3	回调未调用	192
3.3.4	调用次数过少或过多	193
3.3.5	未能传递参数 / 环境值	193
3.3.6	吞掉错误或异常	194
3.3.7	是可信任的 Promise 吗	195
3.3.8	建立信任	197
3.4	链式流	198
3.5	错误处理	206
3.5.1	绝望的陷阱	208
3.5.2	处理未捕获的情况	209
3.5.3	成功的坑	211
3.6	Promise 模式	212
3.6.1	Promise.all([..])	212
3.6.2	Promise.race([..])	213
3.6.3	all([..]) 和 race([..]) 的变体	216
3.6.4	并发迭代	217
3.7	Promise API 概述	219
3.7.1	new Promise(..) 构造器	219
3.7.2	Promise.resolve(..) 和 Promise.reject(..)	219
3.7.3	then(..) 和 catch(..)	220
3.7.4	Promise.all([..]) 和 Promise.race([..])	221
3.8	Promise 局限性	222
3.8.1	顺序错误处理	222
3.8.2	单一值	223
3.8.3	单决议	225
3.8.4	惯性	227
3.8.5	无法取消的 Promise	230