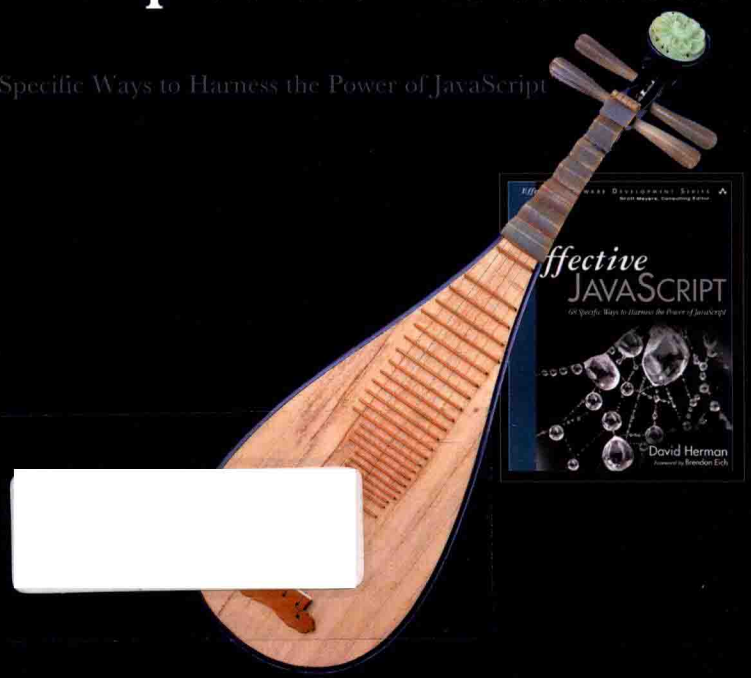
Broadview®
www.broadview.com.cn
PEARSON
· 原味精品书系

# Effective JavaScript（英文版）

## 编写高质量JavaScript代码的68个有效方法

Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript

[美] David Herman 著




中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

· 原味精品书系 ·

# Effective JavaScript（英文版）

## 编写高质量JavaScript代码的68个有效方法

Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript

[美] David Herman 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

## 内容简介

本书由资深 JavaScript 技术专家 David Herman 所著。书中基于 JavaScript 标准的新版本前所未有地阐明了 JavaScript 语言的内部运作机制——帮助你充分利用 JavaScript 语言的表现力。通过全书归纳的 68 个行之有效的方法和大量具体实例，作者详细讲解了如何更有效地运用这门灵活且富有表现力的语言，以及如何规避其缺陷。你将学到如何选择正确的编程风格，管理一些超出意料的问题，以及成功使用 JavaScript 编程完成从数据结构到并发的方方面面。

无论你写了多久的 JavaScript 代码，本书都将有助于增进你对这门强大的编程语言的理解，助你编写更可预测、更可靠且具维护性的程序。



版权贸易合同登记号　图字：01-2015-6098

图书在版编目（CIP）数据

Effective Javascript：编写高质量 JavaScript 代码的 68 个有效方法 =Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript：英文 /（美）赫尔曼（Herman,D.）著. — 北京：电子工业出版社，2016.4
（原味精品书系）
ISBN 978-7-121-27303-2
I. ① E…II. ① 赫…III. ① JAVA 语言－程序设计－英文 IV. ① TP312
中国版本图书馆 CIP 数据核字（2015）第 231509 号

策划编辑：张春雨　刘　芸
责任编辑：徐津平
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980 1/16　　印张：14　　字数：270 千字
版　　次：2016 年 4 月第 1 版
印　　次：2016 年 4 月第 1 次印刷
定　　价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

# 本书赞誉

“这是一本绝不辜负 Effective 软件开发系列期望的编程书籍。对于任何一位想要做到严谨编程的 JavaScript 开发者来说，这本书绝对不容错过。本书阐述了 JavaScript 内部工作细节，以期帮助读者更好地利用 JavaScript 语言优势。”

——Erik Arvidsson，高级软件工程师

“很少有像 David 这样的编程语言极客能如此舒适、友好地讲解编程语言。他带我们领会 JavaScript 的语法和语义，这个过程既令人陶醉又极其深刻。本书以舒适的节奏额外提供了一些‘有问题’的现实案例。当读完这本书后，你会感觉自己获得了一种强大而全面的掌控能力。”

——Paul Irish，Google Chrome 开发主管

“在阅读本书之前，我以为它只是另一本关于如何更好地使用 JavaScript 编程的书籍。然而它远不止如此，它还会使你更深入地理解 JavaScript 这门语言，这一点至关重要。如果没有这层对 JavaScript 的深入理解，那么你绝不会懂得语言本身的任何东西，只会知道其他的程序员是如何编写代码的。

“如果你想成为一名真正优秀的 JavaScript 开发者，那么请阅读此书。就我而言，我多么希望在第一次用 JavaScript 编程时就已经阅读了它。”

——Anton Kovalyov，JSHint 开发者

“如果你正在寻找一本正式且极具可读性及洞察力的 JavaScript 语言书籍的话，那本书正是这样一本书籍。JavaScript 开发者能够从其中找到珍贵的知识宝藏，甚至技术精湛的 JavaScript 程序员也一定能从中获益。对于有其他语言经验而想一头扎进 JavaScript 世界的从业人员来说，本书是迅速学习 JavaScript 的必读之物。然而，不管你的背景如何，都不得不承认作者 Dave Herman 在探索 JavaScript 方面做得非常棒——JavaScript 的优势部分、不足部分或介于两者之间的所有内容都囊括于本书之中。”

——Rebecca Murphey，Bocoup 高级 JavaScript 开发者

“对于任何一位理解 JavaScript 并且想要完全掌握它的人来说，本书都是必不可少的读物。Dave Herman 带给了读者深刻的、具有研究和实践意义的 JavaScript 语言理解，通过一个接一个的例子指导并帮助读者达到与他同样的理解高度。这不是一本寻求捷径的书籍，

恰恰相反，它是一本难得的将经验提炼为指南的书籍。它是一本为数不多的让我毫不犹豫推荐的关于 JavaScript 的书籍。”

——Alex Russell，TC39 成员，Google 软件工程师

“很少有人能有机会学到大师的手艺。这本书弥补了这种遗憾，跟随本书学习 David 对 JavaScript 的研究，就像随一位时间旅行哲学家回到公元前 5 世纪与柏拉图一同学习一般。”

——Rick Waldron，JavaScript 传教士，Bocoup

献给丽莎，我的挚爱。

# 推荐序

众所周知，我在 1995 年 5 月用 10 天时间创建了 JavaScript 语言。迫于现实的压力和冲突管理的势在必行，我将 JavaScript 设计为“看起来像 Java”、“对初学者来说极简单”、“在网景浏览器中几乎能控制它的一切”的编程语言。

鉴于极具挑战性的要求和非常短的时间表，我的解决方案除了正确获得了两大特性（第一类函数、对象原型）之外，还将 JavaScript 设计得从一开始就极具延展性。我知道一些开发者不得不为最开始的几个版本“打补丁”来修正错误，这些先驱的方法比我使用内置库胡乱拼凑的方法要好。举例来说，虽然许多编程语言都限制了可变性，在运行时不能修改或扩展内置对象，或者标准库的名称绑定不能通过赋值被覆盖，但是 JavaScript 几乎允许完全改变每个对象。

总的来说，我相信这是一个很好的设计决定。它明确了某些领域的挑战（如在浏览器的安全边界内，安全地混用可信和不可信的代码）。对于 JavaScript 来说，支持所谓的猴子补丁是很重要的。凭借它，开发者可以编辑标准对象来解决 Bug 和仿效“未来”的功能改造一些旧的浏览器（称为 polyfill 库的 shim，在美式英语中称为 spackle）。

除了这些平凡的应用之外，JavaScript 的可塑性鼓励用户沿着几个更富有创造性的方向形成和发展创新网络。这使得用户仿效其他编程语言，创建了许多工具包和框架库：基于 Ruby 的 Prototype、基于 Python 的 MochiKit、基于 Java 的 Dojo 及基于 Smalltalk 的 TIBET。随后是 jQuery 库（“JavaScript 的新浪潮”），2007 年，我第一次看到它，我觉得自己似乎是一个后来者。它暴风雨般地引领着 JavaScript 的世界，异于其他编程语言，从旧的 JavaScript 库中学习，开辟了浏览器的“查询和做”（query and do）模型，从根本上简化了浏览器。

这引领着 JavaScript 用户群及其创新网络，使 JavaScript 的风格自成一派。这种风格仍在仿效和简化其他的程序库，也促成了现代 Web 标准化的工作。

在这一演变过程中，JavaScript 保持了向后兼容，当然默认情况下，它是可变的。另外，在 ECMAScript 标准最新的版本中新增了一些方法来冻结对象和封闭对象的属性，以防止对象扩展和属性被覆盖。JavaScript 的演进之旅远未结束。这就像生活的语言和生物系统一样，变化始终存在。在此之前，我无法预见一个单一的横扫其他程序库的“标准库”或编码风格。

任何语言都是怪癖的，或者语言几乎都是受限地执行常见的上佳实践。JavaScript 是一门怪癖的或者充满限制主义色彩的编程语言。因此，与大多数其他的编程语言相比，想要高效地使用 JavaScript 编程，开发人员必须学习和追求优秀的风格、正确的用法和上佳的实践。

当考虑如何做到最高效，我相信避免过度编程和构建刚性或教条式的风格指南是至关重要的。

本书以一种平衡的方式讲解 JavaScript 编程。它基于一些具体的实证和经验，而不迂回于刚性或过度的方法。我认为对于许多寻找在不牺牲表现力的前提下，还可以自由采用新思想和编程范式来编写高效 JavaScript 的人来说，本书是一位重要的助手和可信赖的向导。它也是一本备受关注、充满乐趣和拥有许多令人叹为观止的示例的读物。

最后，自 2006 年以来，我有幸结识 David Herman。那是我第一次以 Mozilla 代表的身份邀请他作为特邀专家在 Ecma 标准化机构工作。Dave 深刻、谦逊的专家意见及他的热情让 JavaScript 在书中的每一页大放异彩。本书精彩绝伦！

——Brendan Eich

# 前言

学习一门编程语言，需要熟悉它的语法、形式和结构，这样我们才能编写合法的、符合语义的、具有意义和行为正确的程序。但除此之外，掌握一门语言需要理解其语用，即使用语言特性构建高效程序的方法。后一种范畴是特别微妙的，尤其是对 JavaScript 这样一种灵活而富有表现力的编程语言来说。

这是一本关于 JavaScript 语用学的书。这不是一本入门书籍，我假设你已经在一定程度上熟悉了 JavaScript 和通常的编程。有很多优秀的 JavaScript 入门书籍可供参考，例如，Douglas Crockford 的《JavaScript 语言精粹》和 Marijn Haverbeke 的《JavaScript 编程精解》。本书的目的是帮助你加深理解如何有效地使用 JavaScript 构建更可预测、可靠和可维护的 JavaScript 应用程序和库。

## JavaScript 与 ECMAScript

在深入阅读本书之前向你澄清一些术语是很有必要的。这是一本关于举世皆知的 JavaScript 编程语言的书籍。然而，该语言官方标准定义的规范的名称是 ECMAScript。历史很令人费解，但这可以归结为版权问题：由于法律原因，Ecma 国际标准化组织不能使用“JavaScript”作为其标准名称。更糟的是，标准化组织将其原来的名称 ECMA（“欧洲计算机制造商协会”的英文首字母缩写）改为不是全大写的 Ecma 国际标准化组织。在那时，ECMAScript 这个名字大约也就注定了。

正式来说，当人们提到 ECMAScript 时，通常是指由 Ecma 国际标准化组织制定的“理想”语言。与此同时，JavaScript 这个名字意味着来自语言本身的所有事物，例如某个供应商特定的 JavaScript 引擎。通常情况下，这两个术语是通用的。为了保持清晰度和一致性，在本书中，我将只使用 ECMAScript 来谈论官方标准，使用 JavaScript 指代语言，另外，使用常见的缩写 ES5 来指代第 5 版的 ECMAScript 标准。

## 关于 Web

避开 Web 来谈 JavaScript 是很难的。到目前为止，JavaScript 是唯一为用于客户端应用程序脚本的所有主流浏览器提供内置支持的编程语言。此外，近年来，随着 Node.js 平台的问世，JavaScript 已经成为一门实现服务器端应用程序的流行编程语言。

不过，本书是关于 JavaScript 而非 Web 的编程。有时，谈论一些 Web 相关的例子和应用程序的概念是为了帮助读者理解。但是，这本书的重点是 JavaScript 语言的语法、语义和语用，而不是 Web 平台的 API 和技术。

## 关于并发

JavaScript 一个新奇的方面是在并发环境中其行为是完全不明朗的。ECMAScript 标准（包括第 5 版）关于 JavaScript 程序在交互式或并发环境中的行为只字未提。第 7 章涉及并发，因此，我只是从技术角度介绍一些非官方的 JavaScript 特性。但实际上，所有主流的 JavaScript 引擎都有一个共同的并发模型。尽管在标准中未提及并发，但是致力于并发和交互式的程序是 JavaScript 编程的一个核心概念。事实上，未来版本的 ECMAScript 标准可能会正式地标准化这些 JavaScript 并发模型的共享方面。

# 致谢

这本书在很大程度上要归功于JavaScript的发明者——Brendan Eich。我非常感谢Brendan能邀请我参与JavaScript标准化工作，并对我的Mozilla职业生涯给予指导和支持。

本书中的大部分材料是受优秀的博客文章和在线论文的启发。我从Ben “cowboy” Alman、Erik Arvidsson、Mathias Bynens、Tim “creationix” Caswell、Michaeljohn “inimino” Clement、Angus Croll、Andrew Dupont、Ariya Hidayat、Steven Levithan、Pan Thomakos、Jeff Walden，以及Juriy “kangax” Zaytsev的博客中学到很多东西。当然，本书的最终资源来自ECMAScript规范。ECMAScript规范自第5版以来由Allen Wirfs-Brock不知疲倦地编辑和更新。Mozilla开发者网络仍然是JavaScript API和特性最令人印象深刻的、高品质的在线资源之一。

在策划和写作这本书的过程中，我有许多顾问。在我开始写作之前，John Resig就以作者的角度给了我很多有用的建议。Blake Kaplan和Patrick Walton在早期阶段帮我整理想法和规划这本书的组织结构。在写作的过程中，Brian Anderson、Norbert Lindenberg、Sam Tobin-Hochstadt、Rick Waldron和Patrick Walton给了我很好的建议。

很高兴能够和Pearson的工作人员共事。Olivia Basegio、Audrey Doyle、Trina MacDonald、Scott Meyers和Chris Zahn一直关注我提出的问题，对我的拖延报以耐心，并通融我的请求。我无法想象还能有比这更愉快的写作经历。我对能为Effective系列写一本书感到非常荣幸。因为很久以前我就是*Effective C++*的粉丝，也曾经怀疑自己是否有亲自书写一本Effective系列书籍的荣幸。

我也简直不敢相信自己有这样的好运气，能够找到梦之队一样的技术编辑。我很荣幸Erik Arvidsson、Rebecca Murphey、Rick Waldron和Richard Worth同意编辑这本书，他们为我提供了许多宝贵的批评和建议。他们多次纠正了书中一些真正令人尴尬的错误。

写一本书比我预想的要难得多。如果不是朋友和同事的支持，我可能已经失去了勇气。在我怀疑自己的时候，Andy Denmark、Rick Waldron和Travis Winfrey总是给予我鼓励。

我绝大部分时候是在旧金山柏丽附近的神话般的Java Beach咖啡厅里写作这本书的。那里的工作人员都知道我的名字，并且不等我点餐，他们就知道我想要点什么。我很感谢他们提供了一个舒适的工作场所，并给我提供食物和咖啡。

我的毛茸茸的猫科小友Schmoopy为本书做出了它的最大贡献。至少，它不停地跳上我的膝盖，坐在屏幕前（有可能是笔记本电脑比较温暖）。Schmoopy自2006年以来一直是我的忠实伙伴，我不能想象没有小毛球的生活。

我的整个家庭对这个项目自始至终都很支持，并充满期待。遗憾的是，我无法在我的爷爷和奶奶（Frank 和 Miriam Slamar）去世之前和他们分享这本书的成品。但他们会为我感到激动和自豪，而且本书中有一小段我儿时与爷爷 Frank 编写 BASIC 程序的经历。

最后，我要感谢我一生的挚爱 Lisa Silveria，对于她的付出我无以为报。

# 关于作者

**David Herman** 是 Mozilla 研究院的高级研究员。他拥有格林内尔学院的计算机科学学士学位和美国东北大学的计算机科学硕士及博士学位。David 是 Ecma TC39 委员会成员，负责 JavaScript 的标准化工作。

# 目录

# 1 Accustoming Yourself to JavaScript

JavaScript was designed to feel familiar. With syntax reminiscent of Java and constructs common to many scripting languages (such as functions, arrays, dictionaries, and regular expressions), JavaScript seems like a quick learn to anyone with a little programming experience. And for novice programmers, it's possible to get started writing programs with relatively little training thanks to the small number of core concepts in the language.

As approachable as JavaScript is, mastering the language takes more time, and requires a deeper understanding of its semantics, its idiosyncrasies, and its most effective idioms. Each chapter of this book covers a different thematic area of effective JavaScript. This first chapter begins with some of the most fundamental topics.

## Item 1: Know Which JavaScript You Are Using

Like most successful technologies, JavaScript has evolved over time. Originally marketed as a complement to Java for programming interactive web pages, JavaScript eventually supplanted Java as the web's dominant programming language. JavaScript's popularity led to its formalization in 1997 as an international standard, known officially as ECMAScript. Today there are many competing implementations of JavaScript providing conformance to various versions of the ECMAScript standard.

The third edition of the ECMAScript standard (commonly referred to as ES3), which was finalized in 1999, continues to be the most widely adopted version of JavaScript. The next major advancement to the standard was Edition 5, or ES5, which was released in 2009. ES5 introduced a number of new features as well as standardizing some widely supported but previously unspecified features. Because ES5 support is not yet ubiquitous, I will point out throughout this book whenever a particular Item or piece of advice is specific to ES5.