

YAHOO! PRESS

JavaScript: The Good Parts
Unearthing the Excellence in JavaScript



JavaScript

语言精粹

Douglas Crockford 著
赵泽欣 鄢学鹏 译

O'REILLY®



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

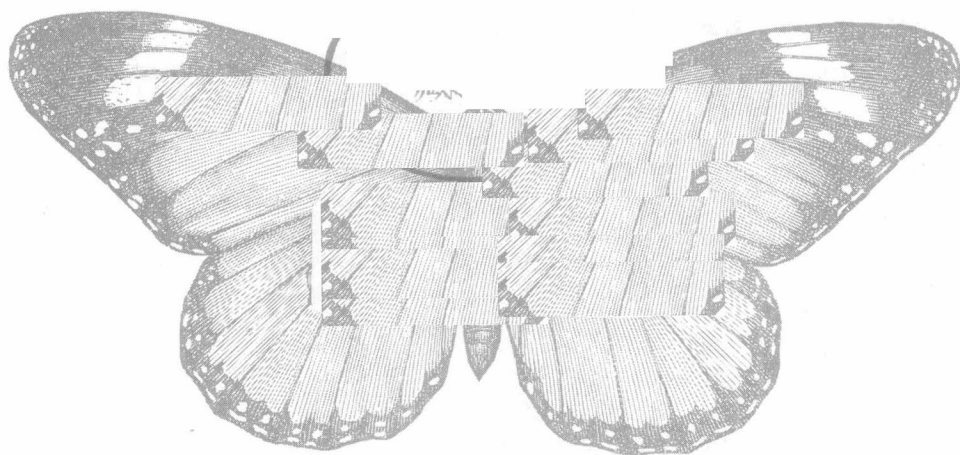
O'REILLY®

JavaScript 语言精粹

JavaScript: The Good Parts

Douglas Crockford 著

赵泽欣 鄢学鹏 译



電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

JavaScript 曾是“世界上最被误解的语言”，因为它担负太多的特性，包括糟糕的交互和失败的设计，但随着 Ajax 的到来，JavaScript “从最受误解的编程语言演变为最流行的语言”，这除了幸运之外，也证明了它其实是一门优秀的语言。Douglas Crockford 在本书中剥开了 JavaScript 沾污的外衣，抽离出一个具有更好可靠性、可读性和可维护性的 JavaScript 子集，让你看到一门优雅的、轻量级的和非常富有表现力的语言。作者从语法、对象、函数、继承、数组、正则表达式、方法、样式和优美的特性这 9 个方面来呈现这门语言真正的精华部分，通过它们完全可以构建出优雅高效的代码。作者还通过附录列出了这门语言的糟粕和鸡肋部分，且告诉你如何避免它们。最后还介绍了 JSLint，通过它的检验，能有效地保障我们的代码品质。

这是一本介绍 JavaScript 语言本质的权威书籍，值得任何正在或准备从事 JavaScript 开发的人阅读，并且需要反复阅读。学习、理解、实践大师的思想，我们才可能站在巨人的肩上，才有机会超越大师，这本书就是开始。

978-0-596-51774-8 JavaScript: The Good Parts. Copyright © 2008 by O'Reilly Media, Inc.

Simplified Chinese edition, jointly published by O'Reilly Media Inc. and Publishing House of Electronics Industry, 2009. Authorized translation of the English edition, 2009 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.

本书中文简体版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社，未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字：01-2009-0879

图书在版编目 (CIP) 数据

JavaScript 语言精粹 / (美) 克洛克福德 (Crockford, D) 著；赵泽欣，鄢学鹏译. —北京：电子工业出版社，2009.4
书名原文: JavaScript: The Good Parts

ISBN 978-7-121-08437-9

I. J… II. ①克…②赵…③鄢… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 030107 号

责任编辑：陈元玉

项目管理：梁 晶

印 刷：北京市天竺颖华印刷厂

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：11 字数：175 千字

印 次：2009 年 4 月第 1 次印刷

定 价：35.00 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 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为20世纪最重要的50本书之一)到GNN(最早的Internet 门户和商业网站)，再到 WebSite (第一个桌面 PC 的Web服务器软件)，O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

Douglas Crockford 是一位大师。

翻译大师的作品，一边是感到万分的荣幸，一边也是兢兢小心。因为吉尔伯特·海特（美国教育家）曾经说过：写了一本很糟糕的书只是犯错而已，而把一本好书翻译得很糟糕则是犯罪。但这样的大师经典之作，即便是冒着犯罪的风险，也值得翻译出来并推荐给大家。一直到现在，依然有很多资深的开发人员对 JavaScript 存有偏见。秦歌和我，分别负责雅虎口碑网和淘宝网的前端开发组，对此的感受更为深刻。但即便是他们也不得不承认，JavaScript 正日益成为互联网中最普及和最重要的开发语言。

Crockford 曾写过很著名的一篇文章——《JavaScript：世界上最被误解的语言》。建议看到这里的所有读者都找来这篇文章 (<http://javascript.crockford.com/javascript.html>) 并仔细阅读。早期的商业原因和规范缺欠给 JavaScript 这门语言蒙上了阴影；Copy+Paste 式滥用也让 JavaScript 显得廉价不堪；更糟糕的是，还有大量不负责任的书籍把蹩脚的用例奉为正统，印成了铅字，让新手们从一开始就走上了歧途。不可否认，JavaScript 自身确实存在着不少瑕疵，但瑕不掩瑜。Crockford 为此凭借他广博的学识和丰富的经验提炼出了 JavaScript 的精华子集。开发人员只要在这个子集的范畴中编程，就既能使用 JavaScript 强大的表现力和卓越的动态性，又能免去许多无端的调试烦恼和安全隐忧。

这本书很薄，但承载的内容却非常丰厚和深入。翻译的过程中我也常感汗颜，原来自诩对 JavaScript 颇为了解的我深刻感受到自己知识面的浅薄和不完整，于是翻译的过程也成为了自己检讨和学习的过程，收获颇丰。Crockford 在前言中告诫大家，这本书是需要反复阅读的。我们同样推荐所有的读者这样做。

我想每一个热爱技术的开发人员都希望自己有一天成为某个领域的大师。我通过翻译大师的著作也得到了一个启示。“取其精华，去其糟粕”本就是前人告诉我们的学习态度与方法，对日新月异的 IT 领域来说更该如此。当我们面对这些层出不穷的新技术新理念时，不要匆忙地照单全收或全盘否定。找到最适合工作或自己最感兴趣的技术，并用科学的方法潜下心来坚持学习和研究，我们同样也可以成为大师！

“大师牛人，宁有种乎？”

最后，我要感谢博文视点的编辑赵士威在本书翻译过程中给予我们的莫大帮助。还有周筠

老师，她爽朗的笑声让人备感亲切。我还要感谢我的同事，来自美国 NCSU 的晓荷，是博采中外的她给我建议，把 JavaScript 的“好、中、坏”特性翻译为更贴切的“精华、鸡肋、糟粕”。当然，家中的领导（负责接管稿费）是一定要特别感谢的。相信我，如果你身后没有一位善解人意的女人，还是不要去做翻译的好。

赵泽欣（小马），鄢学鹏（秦歌）

2008 年 11 月于杭州城西

前言

Preface

要是有所得罪请原谅。本是出自一番好意，
只是想显点粗浅技艺，那才是我们的初衷。

——威廉·莎士比亚，《仲夏夜之梦 (A Midsummer Night's Dream)》

这是一本关于 JavaScript 编程语言的书。它的读者是那些因为偶然事件或好奇心驱使而首次冒险进入 JavaScript 世界的程序员。它也是为那些处于 JavaScript 新手水平并准备更深入地了解这门语言的程序员准备的。JavaScript 是一门令人惊讶的强大语言。它不按常规给出一些挑战，但是作为一门轻量级的语言，它是易于掌握的。

在这里我的目标是帮助你学习 JavaScript 的编程思想。我将展示这门语言的组成部分，并且让你逐步上手，学会如何组合各个部分。这不是一本参考书。它不会对这门语言和它的怪癖进行全面而详尽的介绍。它不包含你希望知道的一切。那些东西你很容易在网上找到。反之，这本书仅包含那些真正重要的东西。

这本书不是写给初学者的。我希望某天写一本叫《JavaScript: 第一部分 (JavaScript: The First Parts)》的书，但是此书非彼书。这也不是一本关于 Ajax 或 Web 编程的书。本书关注的就是 JavaScript，它只是 Web 开发者必须掌握的一门语言。

这不是一本傻瓜书。这本书虽然薄，但知识点密集。它包括了大量的内容。如果为了理解它而不得不反复阅读，请别沮丧。你的付出将会有所回报。

本书的约定

Conventions Used in This Book

本书使用下列排版约定。

斜体 (Italic)

表示专业词汇、链接 (URLs)、文件名和文件扩展名。

`等宽字体 (Constant width)`

表示广义上的计算机编码。它们包括命令、配置、变量、属性、键、请求、函数、方法、类型、类、模块、属性、参数、值、对象、事件、事件处理程序、XML 与 XHTML 标签、宏和关键字。

等宽粗体 (Constant width bold)

表示应该由用户按照字面输入的命令或其他文本。

代码用例

Using Code Examples

这本书是为了帮助你做好工作。一般来说，你可以在程序和文档中使用本书中的代码。你无须联系我们获取许可。例如，使用来自本书的几段代码写一个程序是不需要许可的。出售和散布 O'Reilly 书中用例的光盘 (CD-ROM) 是需要许可的。通过引用本书和用例代码来回答问题是不需要许可的。把本书中大量的用例代码并入到你的产品文档中是需要许可的。

我们赞赏但不强求注明信息来源。一条信息来源通常包括标题、作者、出版者和国际标准书号 (ISBN)。例如：“*JavaScript: The Good Parts* by Douglas Crockford. Copyright 2008 Yahoo! Inc., 978-0-596-51774-8。”。

如果你感到对示例代码的使用超出了正当引用或这里给出的许可范围，请随时通过 permissions@oreilly.com 联系我们。

如何联系我们

How to Contact Us

如果你想就本书发表评论或有任何疑问，敬请联系出版社：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

奥莱利技术咨询（北京）有限公司

北京市 西城区 西直门南大街 2 号 成铭大厦 C 座 807 室
邮政编码：100055
网页：<http://www.oreilly.com.cn>
E-mail：info@mail.oreilly.com.cn

北京博文视点资讯有限公司（武汉分部）

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室
邮政编码：430074
电话：(027) 87690813 传真：(027) 87690595
网页：<http://bv.csdn.net>

读者服务信箱：

reader@broadview.com.cn （读者信箱）
bvtougao@gmail.com （投稿信箱）

与本书有关的在线信息如下所示：

<http://www.oreilly.com/catalog/9780596517748> （原书）
<http://www.oreilly.com.cn/book.php?bn=978-7-121-08437-9> （中文版）

如果你想就本书发表评论或提问技术问题，请发送 E-mail 至：

bookquestions@oreilly.com

关于我们的书籍、会议、资源中心和 O'Reilly 网络的更多信息请登录我们的网址：

<http://www.oreilly.com>

致谢

Acknowledgments

要感谢那些指出我的很多严重错误的审稿者。在生活中，没有什么比有真正聪明的人指出你的过失更好的事情了。他们在书出版之前做这个事情就更好了。谢谢你们，Steve Souders、Bill Scott、Julien LeComte、Stoyan Stefanov、Eric Miraglia 和 Elliotte Rusty Harold。

要谢谢那些同我一起在 Electric Communities 和 State Software 工作并帮助我发现这门语言实际有很多精华的人们，特别是 Chip Morningstar、Randy Farmer、John La、Mark Miller、Scott Shattuck 和 Bill Edney。

要谢谢雅虎公司 (Yahoo! Inc.)，因为它给我时间去从事这个项目，并让我在一个如此之棒的地方工作，谢谢过去和现在在 Ajax Strike Force 的所有成员。我也应该谢谢 O'Reilly Media, Inc.，尤其是使事情变得如此顺利的 Mary Treseler、Simon St.Laurent 和 Sumita Mukherji。

特别感谢 Lisa Drake 教授所做的所有事情。同时，我应该谢谢那些一直为使 ECMAScript 成为一个更好的语言而奋斗的 ECMA TC39 (译注 1) 中的伙计们。

最后，谢谢 Brendan Eich，这位世界上最被误解的编程语言 (译注 2) 的设计者，没有他，这本书也就没有必要了。

译注 1: TC39 是研究 JavaScript 语言进化的技术委员会的名字。详情见: <http://www.ecma-international.org/memento/TC39.htm>。

译注 2: 本书作者曾写过一篇文章《JavaScript: The World's Most Misunderstood Programming Language》(<http://javascript.crockford.com/javascript.html>)。在 2008 年的 3 月 3 日他又写了一篇《The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language》(<http://javascript.crockford.com/popular.html>)。

目录

Table of Contents

前言	I
第 1 章 精华	1
1.1 为什么要使用 JavaScript	2
1.2 分析 JavaScript	2
1.3 一个简单的试验场	4
第 2 章 语法	5
2.1 空白	5
2.2 标识符	6
2.3 数字	7
2.4 字符串	8
2.5 语句	10
2.6 表达式	15
2.7 字面量	17
2.8 函数	19
第 3 章 对象	20
3.1 对象字面量	20
3.2 检索	21
3.3 更新	22
3.4 引用	22
3.5 原型	22
3.6 反射	23
3.7 枚举	24
3.8 删除	24
3.9 减少全局变量污染	25

第 4 章	函数	26
4.1	函数对象	26
4.2	函数字面量	27
4.3	调用	27
4.4	参数	31
4.5	返回	31
4.6	异常	32
4.7	给类型增加方法	32
4.8	递归	34
4.9	作用域	36
4.10	闭包	37
4.11	回调	40
4.12	模块	40
4.13	级联	42
4.14	套用	43
4.15	记忆	44
第 5 章	继承	46
5.1	伪类	47
5.2	对象说明符	50
5.3	原型	50
5.4	函数化	52
5.5	部件	55
第 6 章	数组	58
6.1	数组字面量	58
6.2	长度	59
6.3	删除	60
6.4	枚举	60
6.5	混淆的地方	61
6.6	方法	62
6.7	维度	63
第 7 章	正则表达式	65
7.1	一个例子	66
7.2	结构	70
7.3	元素	72

第 8 章 方法	78
第 9 章 代码风格	94
第 10 章 优美的特性	98
附录 A: 糟粕	101
附录 B: 鸡肋	109
附录 C: JSLint	115
附录 D: 语法图	125
附录 E: JSON	136
索引	147

……我不过略有一些讨人喜欢的地方而已，怎么会有什么迷人的魔力？
——威廉·莎士比亚，《温莎的风流娘儿们 (The Merry Wives of Windsor)》

当我还是一个初出茅庐的程序员时，我想了解所用到的语言的每个特性。我写程序时会尝试去使用所有的特性。我觉得这是炫耀的好方法，而我也的确出了不少风头，因为我对各个特性了如指掌，谁有问题我都能解答。

最终，我认定这些特性中有一部分特性带来的麻烦远远超出它们的价值。其中，一些特性是因为规范很不完善，从而可能导致可移植性的问题；一些特性会导致生成难以理解和修改的代码；一些特性促使我的代码风格过于复杂且易于出错；还有一些特性就是设计错误。有时候语言的设计者也会犯错。

大多数编程语言都有精华部分和鸡肋部分。我发现如果只使用精华部分而避免使用鸡肋的部分，我可以成为一个更好的程序员。毕竟，用糟糕的部件怎么可能构建出好东西呢？

标准委员会想要移除一门语言中的缺陷部分，这几乎是不可能的，因为这样做会损害所有依赖于那些鸡肋部分的糟糕的程序。除了在已存在的一大堆缺陷上堆积更多的特性，他们通常无能为力。并且新旧特性并不总是能和谐共处，可能从而产生出更多的鸡肋部分。

但是，你有权力定义你自己的子集。你完全可以基于精华部分去编写更好的程序。

JavaScript 中鸡肋部分的比重超出了预料。在短到令人吃惊的时间里，它从不存在发展到全球采用。它从来没有在实验室里被试用和打磨。当它还非常粗糙时，它就被直接集成到网景的 Navigator 2 浏览器中。随着 Java™ 的小应用程序 (Java™ applets) 的失败，JavaScript 变成了默认的“网页语言”。作为一门编程语言，JavaScript 的流行几乎完全不受它的质量的影响。

好在 JavaScript 有一些非常精华的部分。在 JavaScript 中，美丽的、优雅的、富有表现力的语言特性就像珍珠和一堆鱼目混杂在一起一样。JavaScript 最本质的部分被深深地隐藏着，以至于多年来对它的主流观点是：JavaScript 就是一个丑陋的、没用的玩具。本书的目的就是要揭示 JavaScript 中的精华，让大家知道它是一门杰出的动态编程语言。JavaScript 就像一块大理石，我要切除那些不好的特性直到这门语言的真实的本质自我显露出来。我相信

我精雕细琢出来的优雅子集大大地优于这门语言的整体，它更可靠、更易读、更易于维护。

这本书不打算全面描述这门语言。反之，它将专注在精华部分上，同时会偶尔警告要去避免鸡肋的部分。这里将被描述的子集可以用来构造可靠的、易读的大小程序。通过仅专注于精华部分，我们就可以缩短学习时间，增强健壮性，并且还能拯救一些树木（译注 1）。

或许只学习精华部分的最大好处就是你可以不用考虑鸡肋的部分。忘掉不好的模式是非常困难的。这是一个非常痛苦的工作，我们中的大多数人都会很不愿意面对。有时候，制定语言的子集是为了让学生更好地学习。但在这里，我制定的 JavaScript 子集是为了让专业人员更好的工作。

1.1 为什么要使用 JavaScript

Why JavaScript?

JavaScript 是一门重要的语言，因为它是 web 浏览器的语言。它与浏览器的结合使它成为世界上最流行的编程语言之一。同时，它也是世界上最被轻视的编程语言之一。浏览器的 API 和文档对象模型（DOM）相当糟糕，导致 JavaScript 遭到不公平的指责。在任何语言中处理 DOM 都是一件痛苦的事情，它的规范制定得很拙劣并且实现互不一致。本书很少涉及 DOM，我认为写一本关于 DOM 的精华的书就像执行一项不可能完成的任务。

JavaScript 是最被轻视的语言，因为它不是所谓的主流语言（SOME OTHER LANGUAGE）（译注 2）。如果你擅长某些主流语言，但却在一个只支持 JavaScript 的环境中编程，那么被迫使用 JavaScript 确是相当令人厌烦的。在那种情形下，大多数人觉得没必要去先学好 JavaScript，但结果他们会惊讶地发现，JavaScript 跟他们宁愿使用的主流语言有很大不同，而且这些不同至为关键。

JavaScript 令人惊异的事情是，在对这门语言没有太多了解，甚至对编程都没有太多了解的情况下，你也能用它来完成工作。它是一门拥有极强表达能力的语言。当你知道要做什么时，它甚至能表现得更好。编程是很困难的事情。绝不应该在对此一无所知时便开始你的工作。

1.2 分析 JavaScript

Analyzing JavaScript

JavaScript 建立在一些非常好的想法和少数非常坏的想法之上。

那些非常好的想法包括函数、弱类型、动态对象和一个富有表现力的对象字面量表示法。那些坏的想法包括基于全局变量的编程模型。

译注 1：作者这里幽默的暗示这本书只关注精华部分，所以书变薄了，用的纸张少了，就可以少砍伐一些树木。

译注 2：专指一些主流语言，像 C、C++、Java、Perl、Python 等。

JavaScript 的函数是(主要)基于词法作用域(lexical scoping)(译注 3)的顶级对象。JavaScript 是第一个成为主流的 lambda(译注 4)语言。实际上,相对 Java 而言,JavaScript 与 Lisp(译注 5)和 Scheme(译注 6)有更多的共同点。它是披着 C 外衣的 Lisp。这使得 JavaScript 成为一个非常强大的语言。

现今大部分编程语言中都流行要求强类型。其原理在于强类型允许编译器在编译时检测错误。我们能越早检测和修复错误,付出的代价就越小。JavaScript 是一门弱类型的语言,所以 JavaScript 编译器不能检测出类型错误。这可能让从强类型语言转向 JavaScript 的开发人员感到恐慌。但事实证明,强类型并不会让你的测试工作轻松。并且我在工作中发现,强类型检查找到的那种错误并不是令我头痛的错误。另一方面,我发现弱类型是自由的。我无须建立复杂的类层次,我永远不用做强制造型,也不用疲于应付类型系统以得到想要的行为。

JavaScript 有非常强大的对象字面量表示法。通过列出对象的组成部分,它们就能简单地被创建出来。这种表示法是促使我创立流行的数据交换格式——JSON 的灵感(译注 7)。(附录 E 中将会有更多关于 JSON 的内容。)

原型继承是 JavaScript 中一个有争议的特性。JavaScript 有一个无类别的(class-free)对象系统,在这个系统中,对象直接从其他对象继承属性。这真的很强大,但是对那些被训练使用类去创建对象的程序员们来说,原型继承是一个陌生的概念。如果你尝试对 JavaScript 直接应用基于类的设计模式,你将会遭受挫折。但是,如果你学习使用 JavaScript 的原型本质,那么你的努力将会有所回报。

JavaScript 在关键思想的选择上饱受非议。虽然在大多数情况下,这些选择是合适的。但是有一个选择相当糟糕:JavaScript 依赖于全局变量来进行连接。所有编译单元的所有顶级变量被撮合到一个被称为全局对象的公共命名空间中。这是一件糟糕的事情,因为全局变量是魔鬼,并且在 JavaScript 中它们是基础性的。幸好,我们接下来会看到,JavaScript 也给我们提供了缓解这个问题的处理方法。

在少数情况下,我们不能忽略鸡肋的部分。另外还有一些不可避免的糟粕,当涉及这些部分时,我会将它们指出来。它们也被总结在附录 A 中。但是我们将成功地避免本书中提到的大多数鸡肋的部分,它们中的大部分被总结在附录 B 中。如果你想学习那些鸡肋的部分及如何拙劣地使用它们,请参阅任何其他 JavaScript 书籍。

译注 3: JavaScript 中的函数是根据词法来划分作用域的,而不是动态地划分作用域的。具体内容参见《JavaScript 权威指南》中译第 5 版相关章节——8.8.1 词法作用域。

译注 4: Lambda 演算是一套用于研究函数定义、函数应用和递归的形式系统。它对函数式编程有巨大的影响,比如 Lisp 语言、ML 语言和 Haskell 语言。更多详细内容请参见 http://zh.wikipedia.org/wiki/λ_演算。

译注 5: LISP(全名 LISt Processor,即链表处理语言),由约翰·麦卡锡在 1960 年左右创造的一种基于 λ 演算的函数式编程语言。更多详细内容请参见 <http://zh.wikipedia.org/wiki/Lisp>。

译注 6: Scheme,一种多范型的编程语言,它是两种 Lisp 主要的方言之一。更多详细内容请见 <http://zh.wikipedia.org/wiki/Scheme>。

译注 7: 本书作者也是 JSON(JavaScript Object Notation)的创立者。官方网站中文版网址是 <http://json.org/json-zh.html>。

定义 JavaScript (又称 JScript) 的标准是 ECMAScript 编程语言第 3 版, 它可以从 <http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf> 获得。本书中描述的语言是 ECMAScript 的一个适当的子集。本书并不描述整个语言, 因为它不包含鸡肋的部分。这里的处理并没有穷尽, 一些模棱两可的情况被回避了。在实践中, 你也应该这样。模棱两可的情况会带来风险和麻烦事。

附录 C 描述了一个叫 JSLint 的编程工具, 它是一个 JavaScript 解析器, 它能分析 JavaScript 问题并报告它包含的缺点。JSLint 提供了一般在 JavaScript 开发中缺少的严谨性。它能让你确信你的程序只包含精华的部分。

JavaScript 是一门有许多差异的语言。它包含很多错误和尖锐的边角 (sharp edges), 所以你可能会疑惑: “为什么我要使用 JavaScript?” 有两个答案。第一个是你没有选择。Web 已变成一个重要的应用开发平台, 而 JavaScript 是唯一一门所有浏览器都可以识别的语言。很不幸, Java 在浏览器环境中失败了, 否则想用强类型语言的人来说就有其他选择了。但是 Java 确实失败了, 而 JavaScript 仍蓬勃发展着, 这恰恰说明 JavaScript 确有其过人之处。

另一个答案是, 尽管 JavaScript 有缺陷, 但是它真的很优秀。它既轻量级又富有表现力。并且一旦你熟练掌握了它, 就会发现函数式编程是一件很有趣的事。

但是为了更好地使用这门语言, 你必须知道它的局限。我将会无情地揭示它们。不要因此而气馁。这门语言的精华部分足以弥补它鸡肋的不足。

1.3 一个简单的试验场

A Simple Testing Ground

如果你有一个 Web 浏览器和任意一个文本编辑器, 那么你就有了运行 JavaScript 程序所需要的一切。首先, 请创建一个 HTML 文件, 可以命名为 *program.html*:

```
<html><body><pre><script src="program.js">
</script></pre></body></html>
```

接下来, 在同一个文件夹内, 创建一个脚本文件, 可以命名为 *program.js*:

```
document.writeln('Hello, world!');
```

下一步, 用你的浏览器打开你的 HTML 文件去查看结果。本书贯彻始终都会用到一个 *method* 方法去定义新方法。下面是它的定义:

```
Function.prototype.method = function (name, func) {
    this.prototype[name] = func;
    return this;
};
```

我会在第 4 章解释它。