

The
Pragmatic
Programmers

TURING

图灵程序设计丛书

CoffeeScript

Accelerated JavaScript Development

深入浅出

CoffeeScript

[英] Trevor Burnham 著
寸志 译



人民邮电出版社
POSTS & TELECOM PRESS

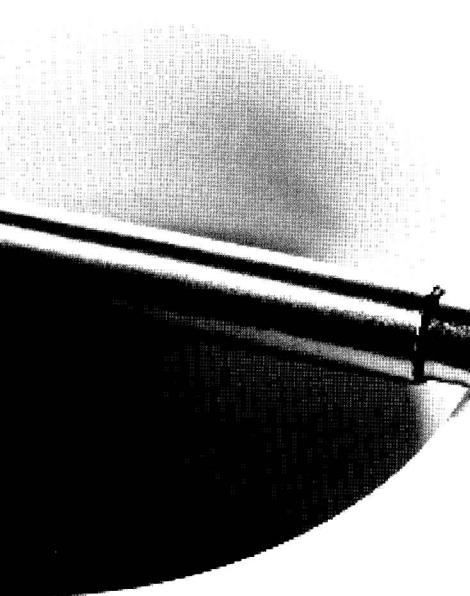
TURING 图灵程序设计丛书

CoffeeScript

Accelerated JavaScript Development

深入浅出

CoffeeScript



人民邮电出版社
北京

图书在版编目 (C I P) 数据

深入浅出CoffeeScript / (英) 伯纳姆
(Burnham, T.) 著 ; 寸志译. -- 北京 : 人民邮电出版社,
2012. 5

(图灵程序设计丛书)

书名原文: CoffeeScript: Accelerated JavaScript
Development

ISBN 978-7-115-27974-3

I. ①深… II. ①伯… ②寸… III. ①JAVA语言—程
序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第065297号

内 容 提 要

CoffeeScript 是一门新语言, 是 JavaScript 预编译器。由它生成的 JavaScript 代码能兼容所有浏览器环境, 可读性更强、更优雅。本书通过一个贯穿全书的小游戏, 从基础知识讲起, 全面透彻地介绍了 CoffeeScript, 并展示了它与 jQuery 与 Node 如何搭配运行。

本书适合 Web 开发人员及对 CoffeeScript 感兴趣的读者。

图灵程序设计丛书

深入浅出CoffeeScript

-
- ◆ 著 [英] Trevor Burnham
译 寸 志
责任编辑 毛倩倩
执行编辑 李 鑫 刘美英
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 7.5
字数: 178千字 2012年5月第1版
印数: 1-4 000册 2012年5月北京第1次印刷
著作权合同登记号 图字: 01-2012-2439 号
ISBN 978-7-115-27974-3
-

定价: 35.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

Copyright © 2011 Pragmatic Programmers, LLC. Original English language edition, entitled *CoffeeScript: Accelerated JavaScript Development*.

Simplified Chinese-language edition copyright © 2012 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由The Pragmatic Programmers, LLC授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书赞誉

“就如CoffeeScript本身一样，Trevor单刀直入，为你指出CoffeeScript的优点，教你如何编写简洁明了的CoffeeScript代码。”

——Scott Leberknight, Near Infinity公司首席架构师

“尽管CoffeeScript还是一门新语言，但它几乎无处不在。这本书将为你展示CoffeeScript到底有多强大和有趣。”

——Stan Angeloff, PSP Web Technologies, 保加利亚地区总经理

“CoffeeScript可能会成为Web程序开发领域最伟大的革新之一。自从我第一次发现它就再没写过一行纯粹的JavaScript代码。希望读者读完这本精彩的书后也能有相同的感悟。”

——Nic Williams博士, Mocra公司CEO/创始人

“本书是极佳的CoffeeScript入门指南，它出自CoffeeScript社区最有威望的成员之一。无论你是前端工程师还是后端开发人员，本书都将助你在短时间内熟悉CoffeeScript。本书实为CoffeeScript开发者必备！”

——Sam Stephenson, JavaScript框架Prototype创始人

“Trevor将语言概述和真实例子结合得非常完美，可想而知我为什么会把CoffeeScript当成iOS、Android和WebOS移动开发的秘密武器了。”

——Wynn Netherland, Changelog共同创办人

“快快准备好，跟着Trevor Burnham享受这从JavaScript到CoffeeScript的旅行，再一次体会Web开发的乐趣吧！”

——Javier Collado, Canonical有限责任公司自动化测试工程师

译者序

CoffeeScript金科玉律：“它只是JavaScript!”

——coffeescript.org

CoffeeScript之父Jeremy Ashkenas在Twitter上有个很好笑的段子。

他说：“我非常想知道在GitHire上有多少人拥有5年CoffeeScript经验，很显然我有很多地方需要向他们请教。”

有人答道：“就是，这些人大概是在一边煮咖啡，一边写脚本。”

的确，与Python或Ruby相比，CoffeeScript非常年轻，到现在才两年出头的时间。

还清楚地记得，我是在了解Zombie.js时第一次接触到CoffeeScript。Zombie.js是一个用于客户端JavaScript测试的轻型框架，在查看Zombie.js的源码时（<http://zombie.labnotes.org/source/>），我就被CoffeeScript的优雅和对应的漂亮文档所吸引了。

谁说使用CoffeeScript不是如边煮咖啡边写脚本一样怡然自得呢？真的，这东西很酷！

CoffeeScript之于JavaScript，就如Less或Sass之于CSS。它吸收了JavaScript语言的精华，并且添加了很多现代脚本语言（Python和Ruby等）所具有的特性，比如列表解析、字符串插值、参数列、吸收操作符等。我想CoffeeScript就是大师Douglas Crockford所想要的那种JavaScript子集（如他在《JavaScript语言精粹》的第10章中所说：“精简的JavaScript不是一个严格的子集，我添加了少许特性……”）。它能够减少三分之一以上的代码量，但由它得到的JavaScript代码去除了语言怪癖，能够兼容所有引擎环境！CoffeeScript编译生成的JavaScript代码可读性很强，而且优雅程度不亚于直接写出的JavaScript代码，你甚至能看到JavaScript在各方面的最佳实践。

虽然CoffeeScript非常年轻，但因为Rails 3.1的直接集成和在Node.js开发方面的天然优势，CoffeeScript在很多方面都有了应用，比方说之前提到的浏览器模拟器Zombie.js，还有基于Express的高级Node.js开发框架Zappa，甚至使用CoffeeScript写成的PSD文件解析器psd.js（<http://meltingice.github.com/psd.js/>）。

但无论CoffeeScript如何优秀和使用广泛，总之要记住它的金科玉律：“它只是JavaScript!”

这句话是本书的精髓。作者Trevor Burnham通过一个贯穿全书的5×5拼字游戏，从基础开始，将CoffeeScript各方面的知识讲解得通俗易懂，如何与jQuery这类非常流行的类库完美集成，如何游刃有余地结合Socket.IO实现Node.js双通道异步通信。JavaScript能做的事情，CoffeeScript也可以，而且做起来更快、更优雅！

想要站在JavaScript开发和Web开发的最前沿吗？这本书正好适合你！

感谢图灵公司引进本书，并给了我翻译的机会。感谢傅志红老师、李松峰老师和李鑫老师，在翻译过程中给予我诸多帮助和鼓励。感谢图灵社区的朋友，他们在阅读样章之后给了我很多反馈。感谢大众点评网尤其是前端团队在翻译过程中对我的理解和支持。感谢Jeremy Ashkenas设计了这门优雅的语言，还要感谢本书作者Trevor Burnham耐心为我解答原文中我不懂的地方。

最后还要感谢我的爸爸、妈妈和女友吴竞男，谢谢你们的支持和理解。尤其是小吴，算是我译稿的第一个读者，谢谢你的诸多批评和建议。

寸志

2012年2月27日

序 言

JavaScript生而自由，但直到现在，它依然处处受到制约。

它从来都不是一门好用的编程语言：运行速度非常慢，在不同的浏览器中有不同的怪异实现，20世纪90年代后期它就牢牢定格在了犹如琥珀般的时间标本里。也许你曾经使用它来实现过下拉菜单或者可排序列表，但是你可能并不享受这样的经历。

我们是幸运的，如今的JavaScript正享受着一场当之无愧的复兴。由于各浏览器厂商的不懈努力，目前它已成为速度最快的主流动态语言。从服务器端到Photoshop，它无处不在，并且它是唯一——门可以在Web各个层面使用的编程语言。

CoffeeScript非常小巧，它的设计初衷就是让开发者更方便地使用JavaScript的精华部分：第一流的函数、类哈希对象，甚至还有被误解颇深的原型链。如果没什么问题的话，你最多只需要写三分之二的代码量，就能生成和原来一样多的JavaScript代码。

CoffeeScript很重视代码的可读性以及消除语法混乱。同时，CoffeeScript与JavaScript之间保持了一对一的关系，这就意味着应该不存在性能耗损的问题。事实上，由于编译器的一些优化，很多JavaScript类库在移植到CoffeeScript之后反而运行得更快了。

选中本书是你的幸运。自从项目早期以来，Trevor就满腔热情地为CoffeeScript作出贡献，没有人能比他更了解这门语言的细枝末节，以及这门语言的特性和疏漏背后的争论史。本书是专家亲临上阵的CoffeeScript入门指导。

我敢肯定，CoffeeScript肯定会造就几个项目，我迫不及待地想听到与此相关的令人振奋的消息——天知道呢——你可能会受此启发创造出一个属于自己的小语言。

Jeremy Ashkenas, CoffeeScript之父

2011年4月

前言

从来没有人认为JavaScript会成为世界上最重要的语言。它从Scheme和Self那里借鉴设计思想，揉合了C语言的代码风格，在十天之内就被临时拼凑了出来。以至于它的名字也是一个令人尴尬的组合——它和另一个几乎没有任何共同之处的语言^①（除少数几个关键字之外）联系到了一起^②。但JavaScript一经推出就势不可挡。它作为唯一所有主流浏览器都支持的脚本语言，很快便成为了Web开发领域的“世界语”。而在21世纪初期，随着Ajax^③技术的风靡，起初仅为网页做些小点缀、微不足道的JavaScript，瞬间作为富应用程序开发语言而羽翼丰满起来。

随着JavaScript的走红，不满之声也从四面八方涌来。有人指责它有大量的语言“怪癖”^④及各种浏览器间实现不一致的问题。其他人则抱怨它缺乏类和继承的特性。而那些刚刚从Ruby或Python入门编程的新人也不太喜欢JavaScript，觉得使用它纷杂的大括号、圆括号以及分号阻碍了他们施展拳脚。

有几个敢于冒险的家伙做了一些Web应用开发框架，使用这些框架可以把其他语言写的代码编译为JavaScript。比较著名的有Google的GWT和280 North公司的Objective-J。但是，没有几个程序员愿意在浏览器和他们之间架上一个厚重的抽象层。他们情愿继续小心翼翼地处理着JavaScript的各种缺陷，委屈求全于它的“精华部分”（也就是Douglas Crockford^⑤在2008年出版的《JavaScript语言精粹》中描述的那些特性）。

直到现在依然如此。

镇上来的新小伙

2009年圣诞节，Jeremy Ashkenas首次发布了CoffeeScript语言，他称之为“JavaScript那不怎

① 这里指的是Java。JavaScript原名叫LiveScript，当时Java比较时髦，且网景公司正和Sun公司（Java拥有者）宣布合作，为搭上顺风车，故将LiveScript更名为JavaScript。但事实上Java与JavaScript是两个差别很大的编程语言。

——译者注

② 参阅《编程人生》（*Coders At Work*）中Peter Seibel对JavaScript之父Brendan Eich的采访。

③ 参阅<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>，这篇文章首次使用词语Ajax介绍该技术。——译者注

④ <http://wtfs.com/>，一个专门收集JavaScript“怪癖”的网站。

⑤ Douglas Crockford，JavaScript开发社区教父，畅销书《JavaScript语言精粹》（*JavaScript: The Good Parts*）的作者。JavaScript程序员把该书奉为圣经，遵循上面的原则来开发JavaScript程序。——译者注

么惹人注目的小兄弟”。每个月，Ashkenas和其他人都会给CoffeeScript添加大量的新特性，很快，该项目便在GitHub^①上吸引了成百上千的关注者。2010年3月，原本使用Ruby编写的CoffeeScript编译器，也由CoffeeScript重写的版本替代了。

2010年圣诞节CoffeeScript发布1.0版，成为GitHub上最受关注的项目之一。

2011年4月，当David Heinemeier Hansson^②证实了有关于在Ruby on Rails 3.1中将支持CoffeeScript的传言时，CoffeeScript又受到了一波关注。

为何CoffeeScript流行得如此之快？我想有3个原因——它的熟悉度、安全性和可读性。

取其精华

JavaScript庞杂而无所不包。在提供众多优秀的函数式语言特性的同时，它还保留了命令式语言的风格。JavaScript这种微妙而强大的功能恰恰会让初学者倍感挫折：函数可以作为传递的参数，也可以作为函数的返回值，任何时候都可以给对象添加新的方法——一言以蔽之，在JavaScript中，函数是第一级对象。

CoffeeScript中保留了这些优良的特性，此外，它还提供了一套语法糖，以便开发者更好地使用它们。

编译优化

想象一下，如果有这么一门编程语言，它没有语法错误，且能让计算机忽略拼写错误的同时还竭力理解你的代码，那这世界该多美好啊！当然了，程序并不会总像开发者所期望的那样运行，但这也正是测试工作的意义所在。

现在再想象一下，你草草地把一次性写好的代码发布出来（其中包含拼写错误），全世界数不清的机器以它们略有不同的方式处理着你的小错误。突然之间，电脑忽略的那些语句直接宕掉了成千上万人使用的应用程序。

令人懊恼的是，现实世界就是这样。JavaScript没有标准的解释器，而数以百计的浏览器和服务端框架都以各自的标准运行JavaScript。跨平台的不兼容问题让程序员们头痛不已。

当然，CoffeeScript并不能解决所有问题，但是其编译器尽力确保输出的JavaScript代码符合Lint^③标准，该标准可以过滤掉常见的人为错误和不规范的惯用句法。当你输入`2=3`或者类似的毫无意义的代码时，CoffeeScript编译器就会主动提醒你——错误越早发现越好。

代码精简

写CoffeeScript很容易上瘾，为什么？请看下面这段JavaScript代码：

① 是一个基于互联网的存取服务，用于使用Git版本控制系统的软件开发项目。GitHub是当今最流行的Git存取站点。
——译者注

② 37signals公司合伙人，Ruby on Rails之父。——译者注

③ <http://www.javascriptlint.com/>：JavaScript验证器。

```
function cube(num) {
  return Math.pow(num, 3);
}
var list = [1, 2, 3, 4, 5];
var cubedList = [];
for (var i = 0; i < list.length; i++) {
  cubedList.push(cube(list[i]));
}
```

再看下面这段实现同样功能的CoffeeScript:

```
cube = (num) -> Math.pow num, 3
list = [1, 2, 3, 4, 5]
cubedList = (cube num for num in list)
```

仔细观察就会发现,这段代码的字符数缩减了一半,代码行数减少了一半还多!类似这样的好处在CoffeeScript中随处可见。正如Paul Graham^①所说:“简洁就是力量!”^②

简短的代码易于编写和阅读,更关键的是易于修改。代码越长就越麻烦,因为任何显著的改动都需要花费大力气。而短小精悍的代码块只需要敲两下键盘就能改好,并且还能促成一种更加敏捷、快速迭代的开发风格。

值得一提的是,切换到CoffeeScript并不是一个“非此即彼”的命题——JavaScript和CoffeeScript的代码还是可以自由地互相融合。CoffeeScript中的字符串和数字与JavaScript中的没什么不同;甚至于像Backbone.js^③这样的JavaScript框架中的自定义类也能在CoffeeScript中运行。所以不必担心在CoffeeScript中调用JavaScript代码会出现什么问题,反之亦然。我们将在第5章讨论如何将CoffeeScript与JavaScript最流行的框架jQuery一起使用。

在CoffeeScript中嵌入JavaScript代码

还应该在这里提一下如何将JavaScript代码嵌入到CoffeeScript代码中。如下所示,使用反引号“```”包裹要嵌入的JavaScript代码即可:

```
console.log `impatient ? useBackticks() :learnCoffeeScript()`
```

CoffeeScript编译器会直接忽略掉反引号里的内容,这也就是说,比如你在反引号内声明一个变量,该变量并不受CoffeeScript作用域规则的约束。

在写CoffeeScript的时候,我没有一次需要使用反引号。毕竟,这种方式看不顺眼还好,怕的是搞不好就会出错。借用Troy McClure的金玉良言:“知道它怎么回事了吧——最好别用。”^④

故事就讲到这里,动手编代码才是真功夫,其他的只是等同于meta(即谈论代码编写),Jeff

① 美国著名程序员、风险投资家、博客和技术作家,著名创业投资公司Y Combinator创始人之一,《黑客与画家》作者。——译者注

② <http://www.paulgraham.com/power.html>: 简洁就是力量。

③ <http://documentcloud.github.com/backbone/>: 一个非常流行的用来开发Web应用的JavaScript MVC框架。

④ 出自美国著名动画情景喜剧《辛普森一家》,Troy McClure是该剧中的一个虚构人物。——译者注

Atwood曾说过：“过多的meta就是谋杀。”^①所以让我们再花那么一点点的篇幅来介绍下你手上这本书（我保证就只有几页了），然后我们就要开始噼里啪啦地敲出些“热乎乎”的代码了。

目标读者

如果你对学习CoffeeScript感兴趣，那本书正好适合你！然而，由于CoffeeScript和JavaScript有着千丝万缕的联系，所以实际上本书贯穿了两种语言——但没有足够的篇幅把这两种语言都教给你。因此，笔者假定你在阅读本书之前已经对JavaScript有所了解了。

当然你不必是John Resig一样的“JavaScript Ninja”（JavaScript忍者）^②。事实上，如果你只是个JavaScript业余爱好者那就太好了！通读本书会让你学到很多JavaScript知识。查看脚注给出的链接，就能找到我推荐的其他资源。如果你是编程新手，绝对应该先看看*Eloquent JavaScript*^③，网上也有一个可以互动的版本。如果你已经有所涉猎，但想更专业点儿，那就读读JavaScript Garden^④。如果你想要一个全面的参考，那没有比Mozilla Developer Network^⑤（Mozilla开发者社区）更适合你的了。

本书会提到很多Ruby的知识。CoffeeScript中诸多优秀的特性都是从Ruby那里来的，比如说隐式返回值、参数列以及if/unless修饰符等。同时也多亏了Rails 3.1，CoffeeScript在Ruby使用者中也有了大量的粉丝。所以，如果你是个Ruby爱好者，那非常棒，你已经先人一步了。如果不是，也别担心，看过一些例子，你就能轻松上手。

如果对本书有任何不清楚的地方，笔者建议你到本书的论坛^⑥上去提问。尽管笔者尽力做到意思清晰，但计算机才是唯一能完全、直接地理解编程语言的东西——但它们不需要买什么书。

本书结构

我们将从挖掘编译和运行CoffeeScript代码的多种方式开始我们的CoffeeScript之旅。接下来我们会深入了解CoffeeScript的语言细节。每章都会介绍一些概念和规则，把它们结合到贯穿本书的示例项目中去（下一节将做详细介绍）。

要想掌握CoffeeScript，必须了解它与JavaScript领域的其他技术结合在一起如何运行。因此在介绍完语言的基础知识之后，我们将简单地了解下jQuery和Node.js。前者是最流行的JavaScript框架，后者则是一个令人兴奋的新项目——能够让JavaScript在浏览器之外运行！我们不会深入介绍这两个工具，主要看一下它们与CoffeeScript结合使用时的绝佳搭配。将它们的力量结合在一起，

① <http://www.codinghorror.com/blog/2009/07/meta-is-murder.html>：《就事论事就是谋杀》。

② John Resig, jQuery之父，“JavaScript Ninja”引自他即将出版的书*Secrets of the JavaScript Ninja*，该书深刻探讨了JavaScript不为人知的一面，是大牛John Resig的又一力作。——译者注

③ <http://eloquentjavascript.net/>

④ <http://javascriptgarden.info/>

⑤ <https://developer.mozilla.org/en/JavaScript/Guide>

⑥ <http://forums.pragprog.com/forums/169>

短短几个小时内就能写出一个完整的多入游戏来。

另外，不管处于什么水平，你都需要确保自己完成每章后面的习题。它们虽然简单但也有一定的挑战性，旨在帮助你了解一些令CoffeeScript程序员防不胜防的陷阱。试着自己把它们做完，之后可参阅附录A中的习题答案。

本书的示例代码、勘误表和讨论区都可以在PragProg相关页面找到：<http://pragprog.com/titles/tbcoffee/coffeescript>。

关于范例游戏：5×5

每章的最后部分，我们会把新的概念应用于一个自创的名为5×5的拼字游戏之中。顾名思义，5×5游戏在一个5×5的网格上进行。游戏开始时每个小格子里会有一个随机的字母，然后玩家轮流交换方格中的字母，每一次交换之后为所有形成的单词打分（通常，每个移动的小格子可与周围的格子形成4个新的单词：横排一个、竖排一个以及两条对角线上的两个——只把从左到右方向上的单词算在内）^①，如图1所示。

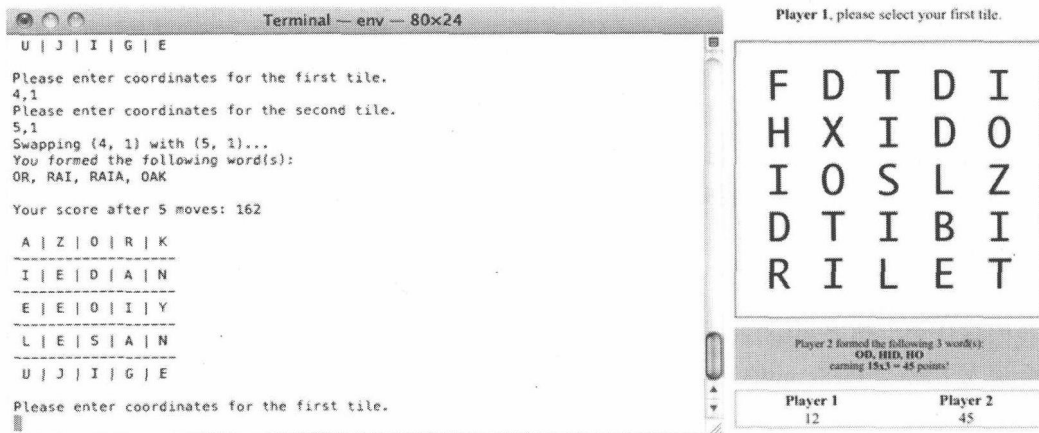


图1 命令行版和Web版5×5游戏，它们共用一份逻辑层处理代码

记分规则是以组成单词的那些字母在Scrabble[®]游戏中的点值叠加为基础，再乘以形成的不相同的单词的个数。因此，拿最好的情况来说，一次交换形成8个不同的单词，则每个单词的分数就都乘以8。已经在之前游戏中使用过的单词则不计算在内。

① 依据本书中5×5游戏的有效单词判断规则来看，每个方向至多不止形成一个单词，因此这里的表述有误，包括后面的“拿最好的情况来说，一次交换形成8个不同的单词……”也有误，最好情况也不止8个单词。

——译者注

② Scrabble是西方流行的英语文字图版游戏，该游戏不同字母有不同分数，是根据在标准书写英语中出现频率计算的，本书范例游戏5×5也采用这种计分规则。——译者注

我们将在第2~4章中创建该游戏的命令行版本，然后在第5章将其移植到浏览器端；最后在第6章最终实现多人游戏的功能。你会发现从命令行移植到浏览器再到服务器上是如此容易——因为它们用的是同一种语言CoffeeScript。

CoffeeScript 社区

一门伟大的语言少不了强大社区的支持。如果你遇到了问题，该到哪里求助呢？

到StackOverflow^①上提问是个很好的办法（记得给你的问题打上coffeescript标签），附上困扰你的代码将会更加有效。如果想更快得到答案，通常可以到Freenode IRC^②的#coffeescript频道找些友善的家伙问问。要是想讨论关于CoffeeScript杂七杂八的事情，试试谷歌小组。对于更为严肃的问题，比如说可能存在的bug，可以把它提交到GitHub上^③。你也可以在那儿对新的语言特性提出要求。CoffeeScript还在不断的完善之中，开发团队欢迎任何形式的反馈。

说到文档，你或许已经在<http://coffeescript.org>上看到过漂亮的官方文档了。还有一份官方的wiki，网址为<http://github.com/jashkenas/coffee-script/wiki>。当然现在也包括你手上这本书。

如何联系我？我在Twitter上有@CoffeeScript账号，大家可以@我；也可以通过一种比较怀旧的方式——给我发邮件，我的邮箱：trevorburnham@gmail.com。

激动人心的Web开发之旅就要开始了，欢迎加入！

① <http://stackoverflow.com>，一个非常流行的程序员问答社区，由被奉为“程序员部落酋长”的Joel Spolsky（《软件随想录》作者）创建。

② Freenode IRC 是专为开源和自由软件社区开设的传统聊天室，大家遇到问题时，可以到这里请教高手，问题将即时得到解答。——译者注

③ <http://github.com/jashkenas/coffee-script/issues>

致 谢

CoffeeScript是一门新语言，但是它从一开始就吸引了各种各样热情备至的关注者，其数量令开发者始料未及。这些不可思议的力量来自在IRC、GitHub、Hacker News、博客、Twitter以及其他地方，是我撰写本书的强大动力。我向在CoffeeScript发展初期热情期待它的每个人表示感谢。

当然，我要感谢Jeremy Ashkenas创造了这门语言，他还非常慷慨地为本书作了序。CoffeeScript不可能找到比他更好的BDFL^①了。同样也要感谢CoffeeScript的其他贡献者，由于他们的名字太多，我就不在此一一列举了^②。

感谢诸位技术评审——任何遗留的错误绝对都是“我的不对”。Javier Collado、Kevin Gisi、Darcy Laycock、Scott Leberknight、Sam Stephenson、Travis Swicegood、Federico Tomassetti、Stefan Turalski和Nic Williams博士给予我大量非常实用的反馈。尤其感谢Jeremy Ashkenas和Michael Ficarra，他们是CoffeeScript项目的核心贡献者，两位从百忙之中抽出时间给我讲解了这门语言的很多细节。还要感谢JavaScript之父Brendan Eich，他温文尔雅，为本书多处指点迷津。

感谢Pragmatic Bookshelf的编辑们。首先也是最应该感谢的是Michael Swaine，你能做本书的编辑让我深感自豪。也要感谢主编Susannah Pfalzer以及Dave Thomas和Andy Hunt两位贵人，这本书出自一个名不见经传的作者之手，写了一个几乎无人知晓的语言，但他们依然冒险将它出版了。

最后，还要感谢我的父母Scott Burnham和Teresa Burnham。他们对我的支持以及为我做出的典范，有不可估量的价值。

① BDFL是Benevolent Dictator For Life的缩写，译作“仁慈的独裁者”。此处的意思是在CoffeeScript开发过程中，Jeremy Ashkenas在必要时或者是在争论不休的时候可以作出最终的决定。——译者注

② <https://github.com/jashkenas/coffee-script/contributors>

目 录

第 1 章 入门指南	1	3.2.1 区间	35
1.1 安装 CoffeeScript	1	3.2.2 切分和剪接	35
1.2 CoffeeScript 编辑器	4	3.3 集合的迭代	37
1.3 “邂逅” coffee	5	3.4 条件迭代	39
1.3.1 编译为 JavaScript	6	3.5 列表解析	40
1.3.2 REPL	7	3.6 模式匹配（或解构赋值）	41
1.4 调试 CoffeeScript	8	3.7 项目：5×5 单人游戏	42
1.5 预备	9	3.8 进阶	48
第 2 章 函数、作用域和上下文	11	3.9 练习	48
2.1 函数基础知识	11	第 4 章 模块与类	51
2.1.1 访问 arguments 对象	13	4.1 模块：解构程序	51
2.1.2 条件表达式和异常	14	4.2 原型的威力	53
2.2 作用域：你在哪里看到它们	16	4.3 类：原型函数	55
2.3 上下文	18	4.4 使用 extends 来继承	56
2.4 属性参数（@arg）	21	4.5 项目：重构 5×5 游戏	59
2.5 默认参数（arg=）	22	4.5.1 Dictionary 类	60
2.6 参数列（...）	24	4.5.2 Grid 类	61
2.7 项目：5×5 游戏输入分析器	25	4.5.3 Player 类	62
2.8 做得好，年轻的学徒	29	4.5.4 Console.Coffee 接口	63
2.9 练习	29	4.6 就如“一勺糖”	63
第 3 章 集合与迭代	31	4.7 练习	64
3.1 作为哈希表的对象	31	第 5 章 jQuery Web 交互开发	66
3.1.1 JavaScript 基础知识：一节 JavaScript 补习课	31	5.1 jQuery 之道	66
3.1.2 精简的 JSON	32	5.2 操作 DOM	67
3.1.3 同名键值对	33	5.3 学会选择	68
3.1.4 吸收操作符：'a?.b'	33	5.4 响应事件	69
3.2 数组	34	5.5 项目：基于浏览器的 5×5 游戏	71
		5.5.1 index.html	71

5.5.2 style.css	72	6.4 项目：多人 5×5 游戏	84
5.5.3 jq5×5.coffee	73	6.4.1 5×5server.coffee	85
5.6 未来是 jQuery 化的	77	6.4.2 5×5client.coffee	89
5.7 练习	77	6.4.3 都结束了	91
第 6 章 Node.js 服务器端程序	79	6.5 客户端、服务器端——有何不同	91
6.1 什么是 Node.js	79	6.6 练习	91
6.2 使用 exports 和 require 构建 模块化代码	80	附录 A 练习答案	92
6.3 异步思想	81	附录 B 运行 CoffeeScript 的几种方法	98
		附录 C JavaScript 开发者备忘录	102