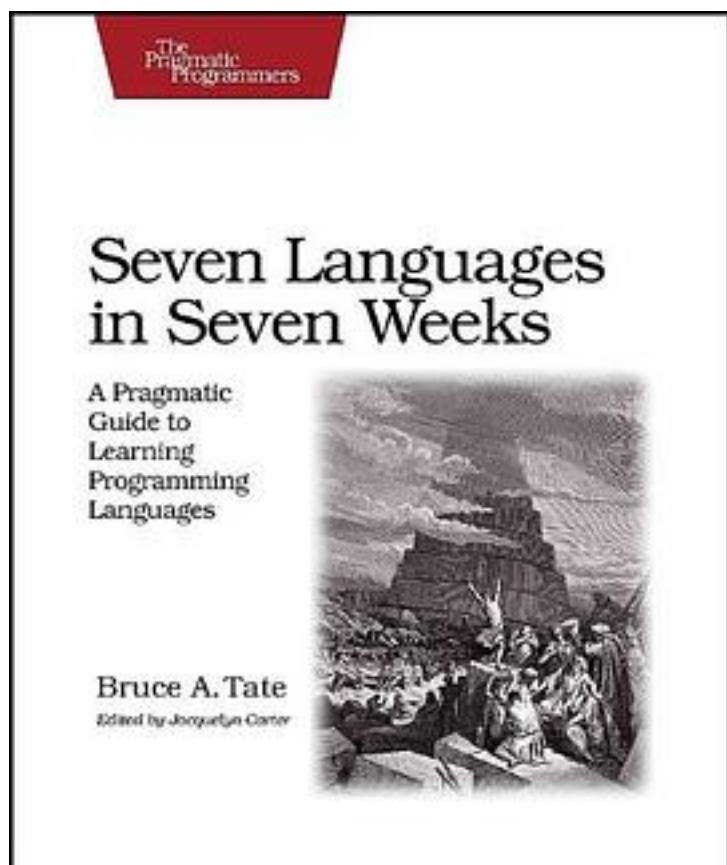


Seven Languages in Seven Weeks



[Seven Languages in Seven Weeks_下载链接1_](#)

著者:[美] Bruce A • Tate

出版者:Pragmatic Bookshelf

出版时间:2010-10-10

装帧:Paperback

isbn:9781934356593

Ruby, Io, Prolog, Scala, Erlang, Clojure, Haskell. With *Seven Languages in Seven Weeks*, by Bruce A. Tate, you'll go beyond the syntax-and beyond the 20-minute tutorial you'll find someplace online. This book has an audacious goal: to present a meaningful exploration of seven languages within a single book. Rather than serve as a complete reference or installation guide, *Seven Languages* hits what's essential and unique

about each language. Moreover, this approach will help teach you how to grok new languages.

For each language, you'll solve a nontrivial problem, using techniques that show off the language's most important features. As the book proceeds, you'll discover the strengths and weaknesses of the languages, while dissecting the process of learning languages quickly--for example, finding the typing and programming models, decision structures, and how you interact with them.

Among this group of seven, you'll explore the most critical programming models of our time. Learn the dynamic typing that makes Ruby, Python, and Perl so flexible and compelling. Understand the underlying prototype system that's at the heart of JavaScript. See how pattern matching in Prolog shaped the development of Scala and Erlang. Discover how pure functional programming in Haskell is different from the Lisp family of languages, including Clojure.

Explore the concurrency techniques that are quickly becoming the backbone of a new generation of Internet applications. Find out how to use Erlang's let-it-crash philosophy for building fault-tolerant systems. Understand the actor model that drives concurrency design in Io and Scala. Learn how Clojure uses versioning to solve some of the most difficult concurrency problems.

It's all here, all in one place. Use the concepts from one language to find creative solutions in another-or discover a language that may become one of your favorites.

作者介绍:

(Biography)

I started in this industry back in 1985, as a co-op with IBM in Austin. I joined IBM full time in 1987, and spent 13 years with them. I later left to join a startup, and ultimately started my own business where I focus on helping customers build software with lightweight technologies.

I've been writing technical books for more than 10 years now, with the last 7 coming since 2000. I write for the love of the craft.

Others have told me that my fundamental strength as an author is the ability to quickly recognize emerging trends. I do tend to find emerging frameworks just as they become popular, and that skill is a mixed blessing that--combined with my complete lack of political tact--gets me in trouble sometimes, as it did with Bitter Java (Java is too hard), Beyond Java (Java is not going to last forever), and most recently, From Java to Ruby: Things Every Manager should Know (there's a better language for some problems, but our managers don't know it yet.)

My promise to you is this: I will always seek to find better ways to do things, and will work hard to tell you the truth, without regard for any notion of political correctness. Thanks for reading.

目录: Changes 9
Beta 2.0 9
Foreword 10

1 Introduction	13
1.1 Method to the Madness	13
1.2 The Languages	15
1.3 Buy this Book	17
1.4 Don' t Buy this Book	18
1.5 Acknowledgements	21
1.6 A final charge	22
2 Ruby	23
2.1 Quick history	24
2.2 Day 1: Finding a nanny	25
2.3 Day 2: Floating down from the sky	33
2.4 Day 3: Serious change	45
2.5 Wrapping Up Ruby	52
3 Io	56
3.1 Io	56
3.2 Day 1: Skipping school, hanging out	57
3.3 Day 2: The Sausage King	68
3.4 Day 3: The Parade and Other Strange Places	75
3.5 Wrapping Up Io	84
4 Prolog	88
4.1 About Prolog	89
4.2 Day 1. An excellent driver	89
4.3 Day 2: Fifteen Minutes to Wapner	101
4.4 Day 3: Blowing up Vegas	112
4.5 Wrapping Up	123
5 Scala	127
5.1 About Scala	127
5.2 Day 1: The Castle on the Hill	131
5.3 Day 2: Clipping bushes and other new tricks	146
5.4 Day 3: Cutting through the fluff	159
5.5 Wrapping Up Scala	167
6 Erlang	172
6.1 Introducing Erlang	172
6.2 Day 1: Appearing Human	176
6.3 Day 2: Changing Forms	186
6.4 Day 3: The Red Pill	197
6.5 Wrapping Up Erlang	208
7 Clojure	212
7.1 Coming Soon	212
8 Haskell	213
8.1 Coming Soon	213
9 Wrap up	214
9.1 Coming Soon	214
A Bibliography	215
Index	216
• • • • •	(收起)

[Seven Languages in Seven Weeks_ 下载链接1](#)

标签

programming

编程

编程语言

计算机

Languages

pragmatic

Programming

计算机科学

评论

想法确实很好，执行上差了点，下一版估计会很不错。Scala这几年大行其道不是偶然啊。Java虽然被吐槽了十年，JVM却越来越主宰互联网了。看完之后最想认真学一下的，一是Scala，二是Haskell。Clojure的Syntax对于没有Lisp背景的人来说还是稍微难接受了点。本科的时候经常听到的说法是，语言不重要，老师们尤其喜欢讲这句话。后来无论是看Joel，Paul Graham还是这本7语言，基本都是在颠覆这个说法。

本来对这本书不抱什么希望，不过看过后觉得还是有点用，作者也挺用心的，并不是简单的列举语法，虽然对于学习一门语言来说几十页的篇幅远远不够，但看完这个之后就能简单入门了，可以开始看代码在实践中学习，而Programming XX之类的书经常if语句都能讲一章，200页过去了还没开始讲这个语言最重要的功能，反而会让初学者迷失并打击积极性

对了快速了解一门语言还是不错的。看了其中ruby, scala, haskell几种，FP成分由少到纯。ruby和haskell都很简洁优美，scala则感笨重。大概拜spark提振所赐，现实中scala的职缺比haskell十倍还多。clojure有空再看

不同的语言或者语言范式是用来解决不同领域的问题的，强烈推荐这本书，它可以作为语言学习的目录书籍。

其实我这个名字应该改一改 Seven Language in Seven Days，哈哈，
10年的大年30那几天就是这个陪我过的

多收了三五种Hello World的写法，主要作用还是让人对几种语言产生基本的观感，具体细节和扩展要另寻它处。Prolog最让人耳目一新，Clojure作为LISP的方言也很想了解一下。（2012-05-24在读）

电子档

长见识的书，可用于快速“了解”各种编程语言及范式间的特性。此外，作者诙谐的写作风格令本书增色不少。

花点时间俯瞰式的快速了解下各种语言基本特性，算是节约时间，提高效率的一种办法。

每年学一门新语言确实是好主意。读过这本书之后，就提前完成7年的计划啦

学习编程，几种类型都涉猎无疑功力大增。Erlang偏居一隅，不必学。Io倒不如JS。Prolog、Scala还是必须学。
作者过赞Ruby，小型网站Web开发尚可；还不如学习用途更广之Python。还想深入，纯FL则Haskell、LISP方言则Clojure。
还未提到的Aspect-Oriented的AspectJ和Data-Oriented的SQL，也是比不可少要学的。

这本书似乎是点到即止的那种类型，我期望更深入的解读。

计算机语言的比较语言学，挺有意思的。语言支撑思维。

这要是木有一点基础鬼才能看懂你在写什么！！！！

以前读过中文版

扫了scala的一段，什么时候有空继续扫erlang和clojure。

七周都没有学会小舌音

看了Clojure和Ruby两章，感觉不错，值得一读。

还是 Clojure 顺眼...

真的只是本简单的入门书，没有介绍一个语言为什么这样做，是最大的遗憾

[Seven Languages in Seven Weeks 下载链接1](#)

书评

“与其说这是项目组，不如说是以机械化方式生产软件的工厂。那时的我，就好比某个酷爱电影的家伙，却居住于偏远小镇，镇上只有一家影院，放的还都是些所谓的‘大片’。直到我自立门户，开始自己生产软件时，我才真正领略到独立电影之妙。就像独立电影不断推动电影业发展那样，...

读完蛮失望的。作者在前言中说：“I will take you beyond syntax”，要“get into the head of a language designer”。他写道“I won’t settle for a superficial treatment.”那么这本书成功与否，就看作者是否实现了他的承诺。在我看来，显然没有。在前言中，作者...

如果没有五线谱，作曲家只能用自然语言谱曲，那会是什么样子？翻一翻几何原本，看看欧几里德用自然语言写数学的公式和推导，那是多么蛋疼。五线谱和数学公式都是一种专门的语言。维特根斯坦说：凡是语言能表达的，都能说清楚，凡是语言不能表达的，都应该保持沉默。国内编...

从众多语言中，挑出本书包含的几门语言，这一过程也许不像你想得那么复杂。我们只不过发了些调查问卷，向本书的潜在读者请教了一番。调查数据汇总上来时，有八门语言入选希望最大。不过，我先是把JavaScript“踢”了出去，因为它实在是过于热门了，取而代之的是原型语言中热门...

编者按
在本文中，截选了七门各不相同的语言的概况，这七门语言，无论教还是学，对我们而言都是一个宏伟目标。书中的代码足以深刻阐释每一门语言的精髓。这七门语言都有非常优秀的支持社区，这也是我选择它们的原因之一。本书为你铺就的学习途径.....
第一周：Ruby 有糖相伴...

曾阅一博 <http://michaelochurch.wordpress.com/2012/07/27/six-languages-to-master/>，曰，当今必学编程语言有六：Python、C、Java、Scala、Clojure、ML。编程语言大类，莫非Object-Oriented，Procedural，Functional，Logical。而《7 languages in 7 weeks》则推荐如下...

首先，看了此书的英文版之后又看了中文版，翻译应该算值得信赖。Jolt的奖项也是对这本书的肯定。而最近这本书火起来，中文版也功不可没。众所周知，每一种语言或复杂或简单，使用者想要使用到一定境界，都不是短短一周可以达成的。所以这本书从根本上就不是像副标题中描述的...

虽然这本书可真够贵的，中文版不到300页，码洋近60元。但是这书是我见过的计算机教程中信息密度最高的一本。作者很淡定的说，你会google我也会google，没必要写一些你随便搜搜就能找到的东西。于是就有这种三十页就包括一门语言的书。而且每个语言又分三段，大概就是语法入门，...

Bruce A.Tate. 七周七语言[M]. 巨成,戴玮,白明,译. 人民邮电出版社,2012-5. ISBN 978-7-115-27611-7. Ruby ---- > Matz:
1993年，當我看到Perl的時候，不知怎麼的，這種混合了Lisp和Smalltalk特徵的面向對象語言讓我的靈感一下子迸發出來。我意識到Perl將成爲一門可提高我們...

合上书，五味杂陈。期间旅程，有因顿悟而欣喜，有因迷茫而忧郁，有因好奇而憧憬，有因晦涩而烦躁，有因好恶而斥责，有因困惑而愤怒……各色滋味聚上心头，汇成一股七彩斑斓的钦佩。## 欣喜
看到那十四行诗般的罗马数字DSL，不由惊叹method_missing还可以这么用。最早是在Pete...

作者从ruby开始，给大家带来了7盘小菜，说实话，如果你未曾接触过其中的语言，都会让你看的一头雾水，当然你要是有一打语言的经验，或者自己亲自设计过语言（命令式，OO，原型，FP），那你应该会有更多的体会。
对于作者的一些跟各种语言设计师的对话，从中你能得到每种语言的...

书写得很一般，分享一下心得：学编程就好比学游泳，不下水是学不会的。掌握新的技能，不要先想着独立解决什么问题，而是先重复一下前人做过的东西。简而言之：先模仿，再创新。
lo语言基于对象和消息。万事皆消息，万物皆对象，消息和对象构成了万事万物。btw，用在并发中...

不够聪明,所以理解不了lisp及其方言.看完之后对一些新兴的语言有了解,不过也仅限于了解.真的想熟悉一门语言,还是要亲自动手开发一个项目才行.帮助程序员拓宽一下知识面吧,但谈不上有帮助.可能还是因为不够聪明...

这不是一本教我每种语言写一个Hellow

World的书，虽然只有不到三百页，还是尽力展现了每种语言的特质。因为只有很短的篇幅，又要介绍很酷的语言特性，这本书采用和不同的叙述方法，基本不会介绍语法的全貌，而是通过自己语言的特性和相邻语言的关联性，来直击要害。我感觉...

书中的每一种语言都有各自的特色，每一种语言代表着一种编程范式。知晓这些语言，了解和理解这些语言所蕴含的知识，对于软件开发者而言是非常有必要的。这会开阔他们的眼界、开拓他们的思维，让他们在实际工作中更容易和更愿意去探索解决问题的不同方法。我推荐每一个软件开发...

开始还以为是单纯介绍基本语法然后扯两个例子就算一门语言，看了众多好评之后决定买了，结果真心满意，适合扩展视野。。。而且此次购物更是一天内送达，从未如此爽过。-----书不厚，但写的...

前端开发一枚，主要会的语言有JS、PHP、Python，因为最近在学函数式才看到了这本书，想多了解了解编程范式。看看函数式到底是个什么地位。周末走马观花地花了一下午看了看面向对象，Java、Ruby原型，Io、JS函数式，Haskell是最纯的函数式语言，难度较大，也是最近在学...

书名给人感觉国内的那种什么21天学通***似的！结果书一直压箱底，后来一次偶然机会捡起来随便翻了下觉得内容很赞，且值得多看几次...本书并不推荐给职场新人或者没有夸语言经验的同学阅读。不过有python基础的同学除外，从头到尾其实可以看到python里面的很多思想借鉴该书。尤...

挑着看的，看了Ruby，Io，Clojure，Haskell几章。开拓了视野，原来编程语言还可以这么玩。不得不说，Python果然是集大成者，吸收各种语言的特性，难怪用起来这么爽。但也丧失了各种语言的最核心的优势，比如Ruby的元编程能力，Lisp的宏等等。这本书中介绍的几种编程语言都或...

[Seven Languages in Seven Weeks_下载链接1](#)