

《Clojure编程》

图书基本信息

书名：《Clojure编程》

13位ISBN编号：9787121197185

10位ISBN编号：7121197189

出版时间：2013-3-26

出版社：电子工业出版社

作者：Chas Emerick, Brian Carper, Christophe Grand

页数：618

译者：徐明明, 杨寿勋

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Clojure编程》

内容概要

Clojure是一种实用的通用语言，它是传奇语言LISP的方言，可与Ruby、Python等动态语言相媲美，更以无缝Java库、服务，以及拥有JVM系统得天独厚的资源优势而胜出。本书既可以用来熟悉Clojure基础知识与常见例子，也可了解其相关的实践领域与话题，更可以看到这一JVM平台上的LISP如何帮助消除不必要的复杂性，为大家在编程实践中解决最具挑战性的问题开辟新的选择——更具灵活性，更适于Web编程和操作数据库，可以应付更为苛刻的应用程序安全要求，更有效的并发性和并行处理、数据分析能力，以及在未来云环境下的更大的发展潜力。

《Clojure编程》

作者简介

作者介绍：

Chas Emerick是软件公司Snowtide informatics的创始人。自2008年以来，致力于Clojure语言本身以及一些开源项目的开发；长期在cemerick.com撰写Clojure、软件开发实践、创业精神，及其他主题的文章。

Brian Carper是一位心理学研究领域的专业程序员，用Clojure从事数据分析与web开发。他开发过一个把Clojure代码转化为CSS代码的编译器以及一个操作关系型数据库的类库，经常在briancarper.net上发表Clojure文章。

Christophe Grand是一个独立顾问，现居法国里昂。他主要从事Clojure培训与编码，参与了Clojure语言的开发，撰写过Enlive和 Moustache类库，也是Counterclockwise——Clojure IDE for Eclipse的贡献者，经常在clj-me.cgrand.net上发表有关Clojure的文章。

译者介绍：

徐明明，阿里巴巴资深软件开发工程师，热衷于研究开源技术及语言；目前网络浏览量最大的Clojure入门教程（<http://xumingming.sinaapp.com/302/clojure-functional-programming-for-the-jvm-clojure-tutorial/>）即译自其手；参与Twitter开源实时计算框架Storm的开发，是该项目的核心贡献者(该项目的主要语言就是Clojure)，同时也翻译了该项目的很多文档，推进了该项目在中国的应用。

书籍目录

第1章 进入Clojure仙境

1

为什么要选择Clojure?

1

获取Clojure

3

Clojure REPL

3

不！括号真的不会让你瞎了眼

6

表达式、操作符、语法以及优先级

7

同像性

10

Clojure Reader

12

命名空间

21

符号解析

23

特殊形式

24

小结

46

这只是开始

48

第1部分

49

第2章 函数式编程

51

所谓函数式编程，到底意味着什么？

52

谈谈值的重要性

52

作为头等公民的函数以及高阶函数

59

函数（功能）的组合

68

纯函数

76

现实生活中的函数式编程

80

第3章 集合类与数据结构

83

抽象优于实现

84

访问集合元素的简洁方式

111	
数据结构的类型	
115	
不可变性和持久性	
123	
元数据	
136	
用Clojure的集合来小试牛刀	
137	
总结	
159	
第4章 多线程和并发	
161	
计算在时间和空间内的转换	
162	
简单的并行化	
168	
状态和标识	
170	
Clojure的引用类型	
172	
并发操作的分类	
174	
原子类型 (Atom)	
176	
通知和约束	
179	
ref	
182	
var	
200	
Agent	
211	
使用Java的并发原语	
227	
总结	
228	
第2部分	
229	
第5章 宏	
231	
宏到底是什么？	
231	
编写你的第一个宏	
237	
调试宏	
239	
语法	
242	

什么时候使用宏

245

宏卫生

247

宏的常见用法和模式

253

隐藏参数：&env 和 &form

255

深入 -> 和 ->>

263

总结

266

第6章 数据类型和协议

267

协议 (Protocol)

268

扩展已有的类型

270

定义你自己的类型

274

实现协议

284

协议自省

293

协议函数分派的边界场景

295

自己实现一个set

296

总结

304

第7章 多重方法

305

多重方法基础

305

通往层级之路

308

层级

310

真正实现多重！

315

还有几件事

317

最后的思考

321

第3部分

323

第8章 Clojure项目的组织与构建

325

项目布局

325	
构建	
339	
最后的思考	
355	
第9章 Java及JVM互操作	
357	
JVM是Clojure的基础	
358	
Java类、方法和字段的使用	
358	
便利的互操作工具	
361	
异常与错误处理	
363	
为了效率进行类型提示	
367	
定义类、实现接口	
372	
在Java里使用Clojure	
386	
乐于合作的伙伴	
393	
第10章 面向REPL的编程	
395	
交互式开发	
395	
工具集	
400	
在REPL里调试、监测和打补丁	
412	
重定义结构的限制	
416	
小结	
418	
第4部分	
419	
第11章 数字与数学	
421	
Clojure的数字	
421	
Clojure数学	
427	
相等与等值	
432	
优化数值效率	
436	
用Clojure可视化芒德布罗集	
448	

第12章 设计模式	
457	
依赖注入	
459	
策略模式	
462	
责任链	
463	
面向方面的编程	
466	
最后的思考	
470	
第13章 测试	
471	
不可变值与纯函数	
471	
clojure.test	
473	
HTML DSL的成长	
482	
依赖断言	
486	
第14章 使用关系数据库	
491	
clojure.java.jdbc	
491	
Korma	
498	
Hibernate	
503	
最后的思考	
510	
第15章 使用非关系型数据库	
511	
安装CouchDB和Clutch	
512	
基本的CRUD操作	
512	
视图	
513	
_changes: 把CouchDB滥用做消息队列	
519	
可随意点选的消息队列	
521	
最后的思考	
524	
第16章 Clojure与Web	
525	
Clojure栈	

525
基石：Ring
526
用Compojure路由请求
533
使用模板
543
最后的思考
552
第17章 布署Clojure Web应用程序
553
Java与Clojure Web架构
553
在本地运行Web应用
561
Web应用程序布署
562
超越简单Web应用程序布署
565
第5部分
567
第18章 明智地选择Clojure类型定义形式
569
第19章 在工作场所引进Clojure
573
只是事实...
573
强调生产效率
574
强调社群
576
审慎
577
第20章 下一步？
579
(dissoc Clojure ‘ JVM)
579
4Clojure
580
Overtone
581
core.logic
581
Pallet
582
Avout
582
Heroku上的Clojure
583

关于作者
585

章节摘录

版权页：插图： I/O、事务以及嵌套的Send 跟ref和atom不同的是，可以非常安全地利用agent来协调I/O或者其他类型的阻塞操作。这使得agent成为任何使用ref以及Clojure的STM来维持状态的程序中不可或缺的一个组件。而且由于agent独特的语义使得它们成为简化涉及异步I/O操作的理想组件——即使你不使用ref。因为agent会串行执行所有发送给它的action，因此它给有副作用的操作提供一个很自然的同步点。你可以建立一个agent来保持比如一个指向文件或者网络socket的OutputStream、一个数据库的连接，或者消息队列的连接等。你可以确定发送到agent的每个action在执行期间都能独占这个连接。这使得它非常容易融合到Clojure的环境中来——包括ref和atom，它们的目标就是最小化它们跟周围环境的副作用。你可能在想，agent怎么能用在STM的事务中呢？发送一个agent action本身就是一个有副作用的操作，那么看起来似乎不能把它放在一个事务里面去执行，因为会重试。进而可能会导致副作用发生多次，从而导致最终结果不对。不过实际上不是这样的。agent是跟Clojure的STM实现紧密融合的，在一个事务内通过send和send-off发送的action会被保持到事务成功提交。这意味着如果一个事务被重试100次，被发送给agent的action也在事务成功提交之后被执行一次。类似的，在一个agent action内部，对send和send-off的调用称为“嵌套发送”——也会被保持到直到这个action完成。在这两种情况下，被发送的action都可能因为一个校验器的失败而被全部抛弃掉。为了展示这些语义并且看看它们到底能给我们带来什么，让我们通过几个例子看看如何使用agent跟ref以及STM合作，在一个高度并行化、高IO负载的情况下简化对I/O操作的协调。利用agent来记录引用状态的变更日志 在181页“对Ref进行修改的细节”一节开发的游戏使用ref来保持角色的状态，证实了Clojure的STM在这种多人并发场景下的能力。但是任何跟这个类似的游戏，特别是那种多个玩家同时玩的游戏，都会对玩家的行为进行跟踪和保存，以及它们的角色所受的影响（加血、减血）。当然我们不会把这种类型日志、持久化以及其他I/O相关的东西写进核心游戏引擎：任何想做的持久化最终会由于事务重启而不一致。

《Clojure编程》

编辑推荐

《Clojure编程》基于新版，提供大量优良的样本代码，可让读者在快速入门、提高的同时，并不遗漏必要的初级概念。抛开Clojure不谈，这也是一本值得推荐的编程语言书籍。或许你现在还不能充分理解Clojure和函数式编程的精髓，或许目前的日常工作还用不到Clojure，但这绝对是一个值得你重视的主题，本书据此精心设计。作者在这一领域有足够的发言权，有能力引导读者从下载/安装开始学起，直到掌握用起进行web开发。《Clojure编程》写作精良，平易近人，具有非常好的“信噪比”，可谓最实用而全面的Clojure图书，对Clojure的状态、标识及并发理念，函数式编程以及Clojure生态系统都有完整阐述。如果有人问你“我应该读什么书上的Clojure”，你可以坚定地指向它！

《Clojure编程》

名人推荐

“ 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

精彩短评

- 1、最新的Clojure著作，书本身给力，翻译团队更给力！
- 2、工具类书，不够有趣，没有aphyr大神的博客写的给力。
- 3、非常适合作为学习 Clojure 的参考书近期对 Clojure 比较感兴趣，准备学习一下，按照以往的习惯，先在网上看看资料，然后买几本书，对照着看，因为各个作者的侧重点不同，这样可以通过不同的角度来学习，效率比较高（个人经验，仅供参考）。共买了3本，除了本书之外另外两本是《Clojure 编程乐趣》和《Clojure 程序设计》，三本书大致翻阅了一遍，发现本书讲解比较深入详细，就集中看了前50页，也就是第一章，综合评价如下：优点：1、讲解很深入，涉及细节较多，大多细节写得不错；2、内容涉及的范围比较全面，不仅有语言方面的，更包含了编程思想、项目、工具、平台以及各种扩展知识；3、书末有详细的索引，虽然是英文的--不过总比没有索引强。特别提出的一点就是在我看过的讲解 Lisp 方言的图书中，唯有本书在第一章就把 Lisp 的同像性讲解得最浅显易懂，就是那个“代码即数据”、“可编程的编程语言”的特性。在第11页的这段话说得特别清楚：“Clojure 和所有的 Lisp 方言一样使用了一种不一样的方式：没有定义一种将会被转换成 AST（抽象语法树）的语法，Clojure 代码是直接表示抽象语法树的 Clojure 数据结构来写的-----也就是说，你写...下的 Clojure 代码相当于其他语言里面的抽象语法树”。缺点：1、版式设计比较差，各种不同的内容不太容易区分，跟其他两本书一比就知道差距了，不过考虑到本书的篇幅比较厚，也就不苛求了；2、有些细节交代得不够清楚，不具备指导作用，比如405页那个 Emacs 下 slime 的配置，基本上用不了。总体而言，还是一本不可多得的好书，所以给 4 颗星。提醒：1、个人感觉本书对读者没有 java 方面的技术背景要求；2、如果你之前学过 Lisp，哪怕只懂一点基础，那么阅读本书就比较容易理解，如果你熟悉 java 但是没有 Lisp 背景，可能理解起来会有一点慢。所以建议从 java 来的读者最好能具备一点 Lisp 的知识，会好很多。阅读更多 ›
- 4、以前自学过段时间common lisp，一直在实际工作中使用嵌入汇编和c，但是试着直接翻书后面的章节，还是有一些不能直接一眼看明白的内容。所以很满意这内容深度，对我来说恰好。总体感觉，这本书应该适合已经有一两个编程语言实际应用的基础，正在试着学习第二个或者第三个编程语言的同学。初学编程，入门的年轻同学的话，可能后面章节会特别难完全看懂
- 5、跳着看完了，对clojure有个初步的认识，不过还没来得及用来处理日常工作。
- 6、从小工到专家 进来时啥也不懂 出来时已是专家
- 7、看了一下目录，内容挺丰富的。总体来说值得
- 8、不适合入门。
- 9、看到try catch的地方，不忍直视的语法。
- 10、喜欢，务实，言简意赅
- 11、不太适合想立即上手的人
- 12、花了几天把第一章基础知识给看完了，目前在读第二章，给我的感觉就是这本书讲的很详细，例子给的充分。在内容组织编排上使用了大部分技术书所使用的循序渐进的方式，一点点加入更高级的内容介绍。这本书已经吸引了我！每天下班回来我都会翻几页。
- 13、[适合做参考书 以为作为入门读物不太合适 没有Function Programming经验 请先学习LISP] 重新打分给了四分(上次给的是三分) 有读了一次 感觉比上次看着有感觉了 可能是因为最近思考比较多吧 :) 算是本好书 最好能看看SICP 对比一下 学而不思则怠吗
- 14、读了少许，至少获知：Clojure 是一种Lisp“方言”；与Java代码无缝互操作；最纯的函数式语言；擅长高并发。
- 15、很好,细致
- 16、期待第二版!!!
- 17、Clojure 已经快忘光了。
- 18、粗略的翻阅了一遍，语言方面巨细无遗，可是真正看完之后总感觉缺点什么，实战方面推荐另一本clojure web，可以很快的开始web编程。思维转变比较大，一时间还适应不了，4clojure网站上的题不错，可以看到高手的解法
- 19、非常适合入门，即使不会java也能很好上手
- 20、很多概念没有介绍就突兀的出来了，要不是之前看过别的，根本不知道怎么回事。

《Clojure编程》

- 21、学习曲线稍陡峭，重实践，前半部分是精华，后半部分节奏过快...
- 22、讲的不少是和jvm相关的东西，不想接触jvm的同学不用看
- 23、翻译得太有才了。。。得中英对照才看得懂。
- 24、虽然应该说说书本身的内容，但是用橡皮擦可以把印的字擦掉你敢信？所以做笔记请确保不要用铅笔，免得你想改的同时把本来的字擦掉了。
- 25、Clojure高大全从入门到精不通
- 26、跟java编程思想一个尿性，得往返读，再读the joy of clojure提高
- 27、工作中用到Storm，通过这本书，看到了Clojure的全貌
- 28、书非常不错，正在拜读。这次在第三方商店购买，建议快递不要太暴力，一本新书整的跟二手的似的，都开胶了，哎！
- 29、内容不错，适合有经验的。

1、小规模折腾Clojure也有一阵子了。几本主要的入门书籍里面，这本Clojure Programming，还有一本Programming Clojure（Clojure程序设计）以及最近刚出来中文版的The Joy of Clojure（Clojure编程乐趣）我都买来看了看。当然还有其他的书，不过这三本应该是看的人最多的，而且其他的书，内容过于老旧，可以略过不看。所以今天主要说说这三本书。说起来，这三本书里面，这本Clojure Programming是我最满意的一本，这也是为什么我挑这一本写书评。原因主要在于对语言的介绍非常全面，几乎涵盖了Clojure的所有细节，甚至包括其在JVM层面可能出现的一些问题。这里做个简单的对比，Programming Clojure那本对于解构的介绍只有不到1页的篇幅，而这本书用了好几页详细阐述这个特性，如果是完全没有接触过lisp的新手，看完Programming Clojure估计还是搞不清楚解构该怎么玩。像这样可以对比的情况在这两本书之间有很多，可以说这本把很多地方都讲得很透彻，即便完全没有接触过lisp和函数式编程的新手也能够慢慢的读书，体会Clojure的独特之处。另外一点值得一提的是，这本书的内容还比较新。书中所有的代码都兼容Clojure 1.3.0以上的环境，并在1.4.0版的Clojure上通过测试运行。当我读这本书的时候，Clojure的最新版本号是1.5.1，这之间的差别并不大。由于Clojure是Lisp方言，语法相对比较固定，所以一般不会出现在运行代码的时候出现与较新的环境不一致的语法。这本书的第三个优点就是内容涉猎比较广，除了基本的Clojure语言特性和函数式编程的介绍以外，还有较为深入的与Java互操作的内容，以及实际项目里的代码规范介绍，以及数据库，Web编程的简单介绍。在最后还提供一些进一步学习的网站资源和开源项目。可以说这本书的内容非常全面。因此，如果是完全没有接触过Lisp和函数式编程的新手，如果要来看一本书来学习Clojure，那么我推荐优先考虑这一本。这里并不是说另外两本书不好。Programming Clojure这本，我觉得讲得比较简单明快，适合有点Lisp基础的人，而The Joy of Clojure这本则更多的是讲Clojure这门语言背后的哲学。这两本在我眼里，都不算新手向的读物。

2、读了第一章，翻译得挺不错的，讲得也很详尽，推荐。感觉读起来比另外一本《Clojure程序设计》要更清晰。整本书的覆盖的范围也较广，从语言本身，跟JVM的交互，FP编程，Web开发，数据库操作都有涉及，适合放在案头参考。

3、如果你没有FP经验，或许真的看完这本书你不知道该如何构建一个项目。因为Clojure给我们的选择实在太多了，没有长时间的实践很难找到最佳实践，好像说了句废话，因为解决方案如此之多，那就只能是不停地实践才能知道哪种是最优路径。另外，我不得不说Clojure是FP领域里最优美的语法，你会深深的爱上这门简洁生动的语言，因为她表达力实在太强了。此书的作者在多个领域都有涉足，另外有很多语言真是精辟之极，举个简单例子（纯凭记忆，可能话语不能与原书匹配）：1.不要给我看流程图，把你的数据结构给我，基本这个系统我也就知道怎么运转的了。2.函数的参数名称越具体，说明你的函数功能就越不够通用。当然，这些话不能算绝对，因为在我看来Erlang里面的流程那是相当的重要，但不能放到这个情况下讨论，毕竟Erlang是FP的COP编程，不能放到一起比较。曾经我痛恨Java啰嗦的语法，极弱的表达能力却要用更臃肿不堪的设计模式来弥补，在很长一段时间里我曾认为设计模式竟然是神秘的银弹。多么恨自己浪费了那么多时间，用这些所谓的奇淫技巧，而且还自得其乐却早已深陷其中，当我发现外面的天空如此绚丽的时候，我完全的忽略了终极目的：从解决实际问题入手！这才是真理！这也是FP的魅力，每个FP的热爱者都应该学习Clojure，她没有那些花哨的东西，但却给你强有力的解决方案，STM也不是银弹，其实Clojure没有发明太多新东西，只是将极简做的更极致，我离真正理解Clojure还有距离，目前写到这里。

4、这本书给总体上不错，翻译质量也比较高，但是有些地方翻译还是很难理解，有些地方怪怪的，比如308页的：“层级”用复数，在中文中这个复数如何体现？书中给了大量的例子，有助于理解，非常好。如果每隔章节增加一些习题就更好了。此外，多重转发这一章感觉不怎么样。Clojure作为一种Lisp方言，与使用广泛的命令式语言有很大的不同，所以这本书，但是在编程的世界，有一些概念是相通的，命令式编程背景的程序员来学习函数式编程则会发现一个更大的世界。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com