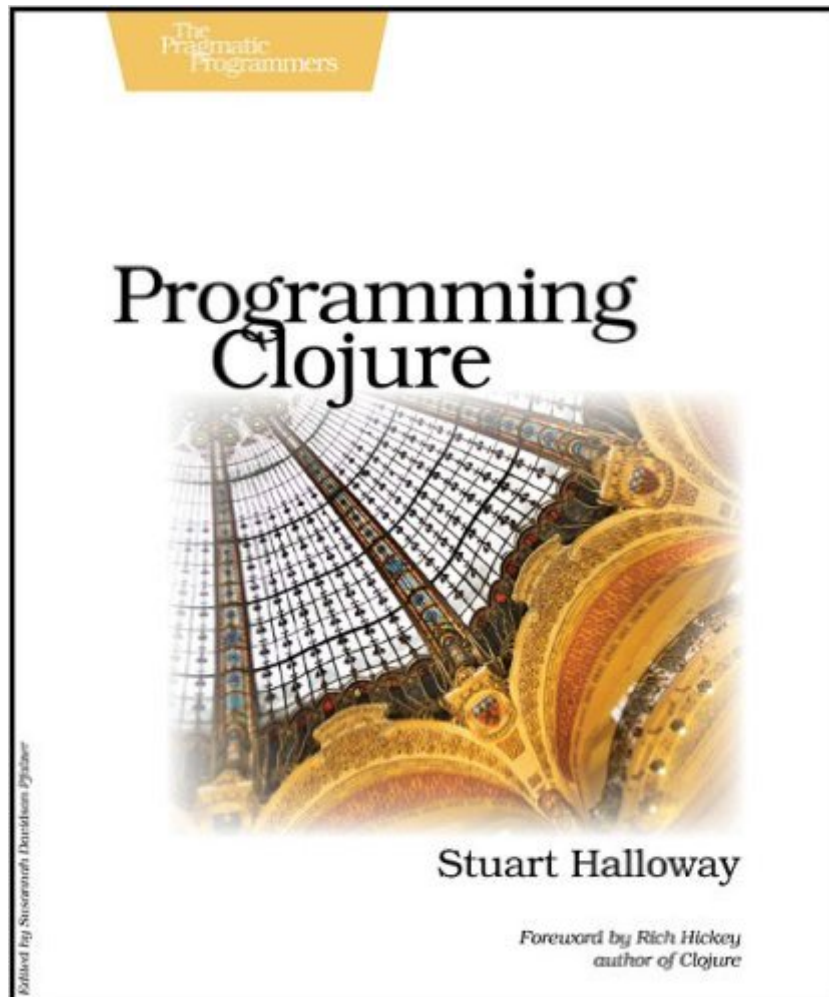# Programming Clojure (Pragmatic Programmers)

Synopsis

Clojure is a dynamic language for the Java Virtual Machine, with a compelling combination of features:Clojure is elegant. Clojure's clean, careful design lets you write programs that get right to the essence of a problem, without a lot of clutter and ceremony.Clojure is Lisp reloaded. Clojure has the power inherent in Lisp, but is not constrained by the history of Lisp.Clojure is a functional language. Data structures are immutable, and functions tend to be side-effect free. This makes it easier to write correct programs, and to compose large programs from smaller ones.Clojure is concurrent. Rather than error-prone locking, Clojure provides software transactional memory.Clojure embraces Java. Calling from Clojure to Java is direct, and goes through no translation layer.Clojure is fast. Wherever you need it, you can get the exact same performance that you could get from hand-written Java code.Many other languages offer some of these features, but the combination of them all makes Clojure sparkle. Programming Clojure shows you why these features are so important, and how you can use Clojure to build powerful programs quickly.

Book Information

Series: Pragmatic Programmers

Paperback: 304 pages

Publisher: Pragmatic Bookshelf; 1 edition (June 7, 2009)

Language: English

ISBN-10: 1934356336

ISBN-13: 978-1934356333

Product Dimensions: 7.5 x 0.9 x 9 inches

Shipping Weight: 1.4 pounds

Average Customer Review: 4.4 out of 5 starsÂ Â See all reviewsÂ (26 customer reviews)

Best Sellers Rank: #1,637,477 in Books (See Top 100 in Books)   #94 inÂ Books > Computers & Technology > Programming > Languages & Tools > Lisp   #119 inÂ Books > Computers & Technology > Programming > Functional   #539 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

It is hard to be objective when reviewing this book. I have struggled to separate my enthusiasm for "Clojure The Language" from my honest opinion of "Clojure The Book".Clojure is a remarkable language.Clojure The Language deserves five stars every day of the week and twice on Sunday.Clojure "The Book", on the other hand, is simply passable. The book is "okay".Clojure is

remarkable partly because Lisp is remarkable, and partly because Clojure has taken the difficult concept of concurrency and turned it into an exhilarating, attractive, efficient, sensible tool. I highly recommend that all curious programmers try the Clojure language.Unfortunately, I cannot give this book a high recommendation. This is not a bad book. But it is definitely not a 5-star book.The 5-star score would place this book in the same category as such beloved books as the Flanagan/Matsumoto Ruby book, or Meyers' Effective C++, or even Rossum's original (now deprecated) An Introduction to Python. Scoring the Stuart Halloway book as "on par" with those other books just simply does not ring true.The Stuart Halloway book is a fast and easy read. However, it is more like a series of really nice blog entries than a solid introductory programming language book. I enjoyed Stuart's writing style, and I even enjoyed all of the sample code.However, at the end of the day, "there is no there there".I needed to refer to other books in order to become PRODUCTIVE with Clojure. When I purchased the Halloway book, no other Clojure books had been written yet, so I relied upon books that were written about coding techniques in other flavors of Lisp.Reading the Halloway book was like watching a really fun and captivating movie trailer. It made me stop and say "Wow! That was cool! ... um... okay. Now can I watch the entire movie?"If you are already fluent in another Lisp dialect, then you probably do not need this book.In fact, if you already know some variant of Lisp, then you need to learn about the JVM and various Java platform ideas far more than you need to study Clojure itself. Clojure itself will come naturally to a "lisper". Yet you may be mystified about where to put your jar files and about what the various Java exceptions are telling you. And if you want to debug your Clojure programs, you will likely be using a Java debugger.If you know Lisp, you can combine that previous knowledge with the Clojure API documentation, and the excellent Mark Volkmann website, and that will probably be enough. An "industrial strength" book on Java (including various compilation and deployment techniques) will help you much more than a book on Clojure.If you have never programmed in any sort of Lisp at all, then I can almost guarantee that this book on Clojure will not be enough to get you fully "up and running" with Clojure. Most certainly you will learn how to write small scripts and games like the ones demonstrated in the book, but you will probably start to feel the weakness of this book as soon as you want to build something larger.

The chances are slim that you are considering buying this book because your job requires you use Clojure, but rather, like me, you are wondering whether Clojure is worth your time. Is it just another programming language fad? What can I do with it that I can't with more established languages like C++, Java or Python? If and only if you are satisfied that it is worth your time will you begin to want

to know all of the details of how to use Clojure.Programming Clojure strikes a nice balance between these two jobs, both justifying Clojure's existence, and lucidly explaining its key features. After a Foreword by Rich Hickey, the language's founder, the preface highlights its key features, and provides a frame of reference for how to think about Clojure from a variety of perspectives (coming from an OO, functional or lisp background). From there it's off to the races.I bought a preview of this book and actually spent the better part of a Saturday plowing through much of it on my laptop, switching over to the command line to play along. Since then, I have gone back to re-read different sections as needed. Though the public documents at clojure.org are quite good, this book is definitely worth paying for if you are serious about learning Clojure.Finally, it's worth mentioning that my experience buying the ebook / print combo directly from the publisher's site was good, providing both pdf and a variety of ebook formats (including one that looks nice on the kindle).

This book lacks anything close to sufficient depth. Clojure is a deep language and Programming Clojure simply fails to do it justice. Most of the book reads like API documentation. The example code, mostly one liners, leave the reader feeling unsatisfied. What if I want to write a program that's more than five lines of code?Stuart Halloway glosses over Clojure's fundamental language paradigms, like functional programming, recursion, and Java. While I suspect it would be near impossible to explain functional programming without devoting an entire book to it, Stuart Halloway doesn't even try, instead hoping that the reader will magically figure it out on his own.The layout of the book is plain frustrating, leading me to wonder if Halloway actually wrote the chapters in the order that they're presented. More than once, Halloway uses the obnoxious literary device of referring to chapters after the current one. "We'll talk about this more in Chapter 18!" Just let me read the chapters in order, Stuart!Save yourself some mental anguish and purchaseÂ The Joy of Clojure: Thinking the Clojure WayÂ orÂ Practical Clojure (Expert's Voice in Open Source)Â instead.

Clojure is a breath of fresh air as well as a history lesson to all of us Java programmers who were raised on object-orientation and (possibly) over-medicated with design patterns. Functional programming with a dynamic, Lisp-based language that has the power of the libraries within the Java ecosystem is both extremely freeing as well as extremely fun.Progamming Clojure provides gentle yet thorough coverage to what is a new world for many of us. I've found Clojure great for new, green-field development projects but it's also very nice in existing Java environments. For example, I've used the skills acquired in this book to use Clojure for writing Eclipse plug-ins, JSR-286 portlets and even for REST/JSON services running on Google App Engine. Frankly,

coming from a person who has also tried out many other alternative languages (including Python, Groovy, Scala, Ruby, etc.), Clojure has made programming more cerebral, productive and exciting all at the same time.

Programming Clojure (Pragmatic Programmers) Clojure Reactive Programming - How to Develop Concurrent and Asynchronous Applications with Clojure The Joy of Clojure: Thinking the Clojure Way 3D Game Programming for Kids: Create Interactive Worlds with JavaScript (Pragmatic Programmers) Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide (The Facets of Ruby) Programming Groovy: Dynamic Productivity for the Java Developer (Pragmatic Programmers) Programming Groovy 2: Dynamic Productivity for the Java Developer (Pragmatic Programmers) Practical Programming: An Introduction to Computer Science Using Python 3 (Pragmatic Programmers) Pragmatic Guide to Git (Pragmatic Guides) Web Design for Developers: A Programmer's Guide to Design Tools and Techniques (Pragmatic Programmers) Seven Concurrency Models in Seven Weeks: When Threads Unravel (The Pragmatic Programmers) The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers) The Cucumber Book: Behaviour-Driven Development for Testers and Developers (Pragmatic Programmers) An Illustrated Guide for z/Architecture Assembler Programmers: A compact reference for application programmers Delphi Nuts & Bolts for Experienced Programmers: For Experienced Programmers Certified Programming with Dependent Types: A Pragmatic Introduction to the Coq Proof Assistant (MIT Press) Programming Elixir 1.2: Functional |> Concurrent |> Pragmatic |> Fun Java: The Simple Guide to Learn Java Programming In No Time (Programming,Database, Java for dummies, coding books, java programming) (HTML,Javascript,Programming,Developers,Coding,CSS,PHP) (Volume 2) Clojure for the Brave and True: Learn the Ultimate Language and Become a Better Programmer The Joy of Clojure