

---

# Functional Programming For The Object Oriented Programmer Ebook Brian Marick

---

The One Thing You Cannot Do in Object-Oriented Programming but You Can in  
Functional FP vs OOP | For Dummies Ditch your Favorite Programming Paradigm How  
to Transition from Object Oriented Programming to Functional Programming  
Functional programming - A general introduction [VDM19] The 4 Rules of Simple  
Design In Functional Programming by Uberto Barbini How to Transition from Object  
Oriented Programming to Functional Programming (subtitled) Python Basics Live  
Stream Don't Know Where To Start Functional Programming for the Object Oriented -  
Øystein Kolsrud Write AWESOME Code With These 3 Functional Programming  
Concepts Book review: \"An Accessible Introduction to Common Lisp and Functional  
Programming\" (2015) Why Isn't Functional Programming the Norm? - Richard

Feldman Hitler reacts to functional programming Talk: Functional Programming You Already Know - Kevlin Henney (Curbralan) - Codemotion Rome 2015 Object Oriented Programming vs Functional Programming Functional Programming IS NO BETTER than Object Oriented Programming | Prime Reacts A Delicious \$15 Functional Programming e-book Bundle Book Read: Introduction \"Mostly Adequate Guide to Functional Programming\" Functional Programming in 40 Minutes • Russ Olsen • GOTO 2018 Object-oriented Programming in 7 minutes | Mosh  
Object Oriented and Functional Programming  
Eloquent JavaScript, 3rd Edition  
An Exploration of Functional Programming and Object Composition in JavaScript (Scala Edition)  
Advanced R  
Paradigm Over Syntax  
Functional Programming for Java Developers  
Programming Scala  
Functional Programming in JavaScript  
Functional Programming in R  
Get Programming with Haskell  
Advanced Statistical Programming for Data Science, Analysis and Finance  
Pragmatic Functional Programming

Metaprogramming in R  
Grokking Functional Programming  
Tools for Better Concurrency, Abstraction, and Agility  
Programming Scala  
Bridging the Divide Between Opposing Paradigms  
Becoming Functional  
Advanced Statistical Programming for Data Science, Analysis and Finance  
Reliable Software Releases Through Build, Test, and Deployment Automation  
Functional techniques for sequential and parallel programming with Scala  
An Introduction to the Programming Language  
Scala Cookbook

*Functional  
Programming  
For The Object  
Oriented  
Programmer  
Ebook Brian  
Marick*

*OMB No.  
1915024754803  
edited by*

---

**BRADSHAW SIMONE**

---

Object Oriented and

Functional Programming  
"O'Reilly Media, Inc."  
Learn how to manipulate  
functions and expressions  
to modify how the R  
language interprets itself.  
This book is an  
introduction to

metaprogramming in the  
R language, so you will  
write programs to  
manipulate other  
programs.  
Metaprogramming in R  
shows you how to treat  
code as data that you can

generate, analyze, or modify. R is a very high-level language where all operations are functions and all functions are data that can be manipulated. This book shows you how to leverage R's natural flexibility in how function calls and expressions are evaluated, to create small domain-specific languages to extend R within the R language itself. What You'll Learn Find out about the anatomy of a function in R Look inside a function call Work with R expressions and environments

Manipulate expressions in R Use substitutions Who This Book Is For Those with at least some experience with R and certainly for those with experience in other programming languages.

### **ELOQUENT JAVASCRIPT, 3RD EDITION**

Packt Publishing Ltd  
Summary Functional Programming in JavaScript teaches JavaScript developers functional techniques that will improve extensibility, modularity, reusability,

testability, and performance. Through concrete examples and jargon-free explanations, this book teaches you how to apply functional programming to real-life development tasks Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology In complex web applications, the low-level details of your JavaScript code can obscure the workings of the system as a whole. As a coding style, functional

programming (FP) promotes loosely coupled relationships among the components of your application, making the big picture easier to design, communicate, and maintain. About the Book Functional Programming in JavaScript teaches you techniques to improve your web applications - their extensibility, modularity, reusability, and testability, as well as their performance. This easy-to-read book uses concrete examples and clear explanations to show you how to use

functional programming in real life. If you're new to functional programming, you'll appreciate this guide's many insightful comparisons to imperative or object-oriented programming that help you understand functional design. By the end, you'll think about application design in a fresh new way, and you may even grow to appreciate monads! What's Inside High-value FP techniques for real-world uses Using FP where it makes the most sense Separating the logic

of your system from implementation details FP-style error handling, testing, and debugging All code samples use JavaScript ES6 (ES 2015) About the Reader Written for developers with a solid grasp of JavaScript fundamentals and web application design. About the Author Luis Atencio is a software engineer and architect building enterprise applications in Java, PHP, and JavaScript. Table of Contents PART 1 THINK FUNCTIONALLY Becoming functional Higher-order JavaScript

## PART 2 GET FUNCTIONAL

Few data structures,  
many operations Toward  
modular, reusable code  
Design patterns against  
complexity PART 3

ENHANCING YOUR  
FUNCTIONAL SKILLS

Bulletproofing your code  
Functional optimizations  
Managing asynchronous  
events and data

An Exploration of  
Functional Programming  
and Object Composition in

JavaScript Simon and  
Schuster

Create robust and  
maintainable Java  
applications using the

functional style of  
programming About This  
Book Explore how you can  
blend object-oriented and  
functional programming  
styles in Java Use lambda  
expressions to write  
flexible and succinct code  
A tutorial that strengthens  
your fundamentals in  
functional programming  
techniques to enhance  
your applications Who  
This Book Is For If you are  
a Java developer with  
object-oriented  
experience and want to  
use a functional  
programming approach in  
your applications, then

this book is for you. All  
you need to get started is  
familiarity with basic Java  
object-oriented  
programming concepts.  
What You Will Learn Use  
lambda expressions to  
simplify code Use  
function composition to  
achieve code fluency  
Apply streams to simply  
implementations and  
achieve parallelism  
Incorporate recursion to  
support an application's  
functionality Provide more  
robust implementations  
using Optionals  
Implement design  
patterns with less code

Refactor object-oriented code to create a functional solution Use debugging and testing techniques specific to functional programs In Detail Functional programming is an increasingly popular technology that allows you to simplify many tasks that are often cumbersome and awkward using an object-oriented approach. It is important to understand this approach and know how and when to apply it. Functional programming requires a different

mindset, but once mastered it can be very rewarding. This book simplifies the learning process as a problem is described followed by its implementation using an object-oriented approach and then a solution is provided using appropriate functional programming techniques. Writing succinct and maintainable code is facilitated by many functional programming techniques including lambda expressions and streams. In this book, you will see numerous

examples of how these techniques can be applied starting with an introduction to lambda expressions. Next, you will see how they can replace older approaches and be combined to achieve surprisingly elegant solutions to problems. This is followed by the investigation of related concepts such as the Optional class and monads, which offer an additional approach to handle problems. Design patterns have been instrumental in solving common problems. You

will learn how these are enhanced with functional techniques. To transition from an object-oriented approach to a functional one, it is useful to have IDE support. IDE tools to refactor, debug, and test functional programs are demonstrated through the chapters. The end of the book brings together many of these functional programming techniques to create a more comprehensive application. You will find this book a very useful resource to learn and apply functional

programming techniques in Java. Style and approach In this tutorial, each chapter starts with an introduction to the terms and concepts covered in that chapter. It quickly progresses to contrast an object-oriented approach with a functional approach using numerous code examples. (Scala Edition) Packt Publishing Ltd  
All software design is composition: the act of breaking complex problems down into smaller problems and composing those

solutions. Most developers have a limited understanding of compositional techniques. It's time for that to change. In "Composing Software", Eric Elliott shares the fundamentals of composition, including both function composition and object composition, and explores them in the context of JavaScript. The book covers the foundations of both functional programming and object oriented programming to help the reader better understand how to build and structure



complex applications using simple building blocks. You'll learn: Functional programming Object composition How to work with composite data structures Closures Higher order functions Functors (e.g., `array.map`) Monads (e.g., `promises`) Transducers Lenses All of this in the context of JavaScript, the most used programming language in the world. But the learning doesn't stop at JavaScript. You'll be able to apply these lessons to any language.

This book is about the timeless principles of software composition and its lessons will outlast the hot languages and frameworks of today. Unlike most programming books, this one may still be relevant 20 years from now. This book began life as a popular blog post series that attracted hundreds of thousands of readers and influenced the way software is built at many high growth tech startups and fortune 500 companies

## ADVANCED R

Simon and Schuster  
The step-by-step guide to going live with new software releases faster - reducing risk and delivering more value sooner! \* \*Fast, simple, repeatable techniques for deploying working code to production in hours or days, not months!  
\*Crafting custom processes that get developers from idea to value faster than ever.  
\*Best practices for everything from source code control to

dependency management and in-production tracing. \*Common obstacles to rapid release - and pragmatic solutions. In too many organizations, build, testing, and deployment processes can take six months or more. That's simply far too long for today's businesses. But it doesn't have to be that way. It's possible to deploy working code to production in hours or days after development work is complete - and Go Live presents comprehensive processes

and techniques for doing so. Written by two of the world's most experienced software project leaders, this book demonstrates how to dramatically increase speed while reducing risk and improving code quality at the same time. The authors cover all facets of build, testing, and deployment, including: configuration management, source code control, release planning, auditing, compliance, integration, build automation, and more. They introduce a

wide range of advanced techniques, including inproduction monitoring and tracing, dependency management, and the effective use of virtualization. For each area, they explain the issues, show how to mitigate the risks, and present best practices. Throughout, Go Live focuses on powerful opportunities for individual improvement, clearly and simply explaining skills and techniques so they can be used every day on real projects. With this book's

help, any development organization can move from idea to release faster -- and deliver far more value, far more rapidly. *Paradigm Over Syntax* Createspace Independent Publishing Platform

If you've had trouble trying to learn Functional Programming (FP), you're not alone. In this book, Alvin Alexander -- author of the Scala Cookbook and former teacher of Java and Object-Oriented Programming (OOP) classes -- writes about his own problems in trying to understand FP, and how

he finally conquered it. What he originally learned is that experienced FP developers are driven by two goals: to use only immutable values, and write only pure functions. What he later learned is that they have these goals as the result of another larger goal: they want all of their code to look and work just like algebra. While that sounds simple, it turns out that these goals require them to use many advanced Scala features - which they often use all at the same time. As a

result, their code can look completely foreign to novice FP developers. As Mr. Alexander writes, "When you first see their code it's easy to ask, 'Why would anyone write code like this?'" Mr. Alexander answers that "Why?" question by explaining the benefits of writing pure functional code. Once you understand those benefits -- your motivation for learning FP -- he shares five rules for programming in the book: All fields must be immutable ('val' fields). All functions must be pure

functions. Null values are not allowed. Whenever you use an 'if' you must also use an 'else'. You won't create OOP classes that encapsulate data and behavior; instead you'll design data structures using Scala 'case' classes, and write pure functions that operate on those data structures. In the book you'll see how those five, simple rules naturally lead you to write pure, functional code that reads like algebra. He also shares one more Golden Rule for learning: Always ask "Why"? Lessons in the

book include: How and why to write only pure functions Why pure function signatures are much more important than OOP method signatures Why recursion is a natural tool for functional programming, and how to write recursive algorithms Because the Scala 'for' expression is so important to FP, dozens of pages explain the details of how it works In the end you'll see that monads aren't that difficult because they're a natural extension of the Five Rules The book finishes

with lessons on FP data modeling, and two main approaches for organizing your pure functions As Mr. Alexander writes, "In this book I take the time to explain all of the concepts that are used to write FP code in Scala. As I learned from my own experience, once you understand the Five Rules and the small concepts, you can understand Scala/FP." Please note that because of the limits on how large a printed book can be, the paperback version does not include all of the chapters that are in the

Kindle eBook. The following lessons are not in the paperback version: Grandma's Cookies (a story about pure functions) The ScalaCheck lessons The Type Classes lessons The appendices Because those lessons didn't fit in the print version, they have been made freely available online. (Alvin Alexander ([alvinalexander.com](http://alvinalexander.com)) wrote the popular Scala Cookbook for O'Reilly, and also self-published two other books, How I Sold My Business: A Personal Diary, and A Survival

Guide for New Consultants.) **Functional Programming for Java Developers** Springer Your guide to the functional programming paradigm Functional programming mainly sees use in math computations, including those used in Artificial Intelligence and gaming. This programming paradigm makes algorithms used for math calculations easier to understand and provides a concise method of coding algorithms by

people who aren't developers. Current books on the market have a significant learning curve because they're written for developers, by developers—until now. Functional Programming for Dummies explores the differences between the pure (as represented by the Haskell language) and impure (as represented by the Python language) approaches to functional programming for readers just like you. The pure approach is best suited to researchers who have no desire to create

production code but do need to test algorithms fully and demonstrate their usefulness to peers. The impure approach is best suited to production environments because it's possible to mix coding paradigms in a single application to produce a result more quickly. Functional Programming For Dummies uses this two-pronged approach to give you an all-in-one approach to a coding methodology that can otherwise be hard to grasp. Learn pure and impure when it comes to

coding Dive into the processes that most functional programmers use to derive, analyze and prove the worth of algorithms Benefit from examples that are provided in both Python and Haskell Glean the expertise of an expert author who has written some of the market-leading programming books to date If you're ready to massage data to understand how things work in new ways, you've come to the right place! [Programming Scala](#) O'Reilly Media

Summary Get Programming with Haskell leads you through short lessons, examples, and exercises designed to make Haskell your own. It has crystal-clear illustrations and guided practice. You will write and test dozens of interesting programs and dive into custom Haskell modules. You will gain a new perspective on programming plus the practical ability to use Haskell in the everyday world. (The 80 IQ points: not guaranteed.) Purchase of the print book includes

a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Programming languages often differ only around the edges—a few keywords, libraries, or platform choices. Haskell gives you an entirely new point of view. To the software pioneer Alan Kay, a change in perspective can be worth 80 IQ points and Haskellers agree on the dramatic benefits of thinking the Haskell way—thinking functionally, with type

safety, mathematical certainty, and more. In this hands-on book, that's exactly what you'll learn to do. What's Inside Thinking in Haskell Functional programming basics Programming in types Real-world applications for Haskell About the Reader Written for readers who know one or more programming languages. Table of Contents Lesson 1 Getting started with Haskell Unit 1 - FOUNDATIONS OF FUNCTIONAL PROGRAMMING Lesson 2 Functions and functional

programming Lesson 3 Lambda functions and lexical scope Lesson 4 First-class functions Lesson 5 Closures and partial application Lesson 6 Lists Lesson 7 Rules for recursion and pattern matching Lesson 8 Writing recursive functions Lesson 9 Higher-order functions Lesson 10 Capstone: Functional object-oriented programming with robots! Unit 2 - INTRODUCING TYPES Lesson 11 Type basics Lesson 12 Creating your own types Lesson 13 Type classes Lesson 14

Using type classes Lesson 15	I/O Lesson 23	with donotation Lesson 32
Capstone: Secret messages! Unit 3 - PROGRAMMING IN TYPES	Lesson 24 Working with files	The list monad and list comprehensions Lesson 33
Lesson 16 Creating types with "and" and "or"	Lesson 25 Working with binary data	Capstone: SQL-like queries in Haskell Unit 6 - ORGANIZING CODE AND BUILDING PROJECTS
Lesson 17 Design by composition—Semigroups and Monoids	Lesson 26 Capstone: Processing binary files and book data	Lesson 34 Organizing Haskell code with modules
Lesson 18 Parameterized types	Unit 5 - WORKING WITH TYPE IN A CONTEXT	Lesson 35 Building projects with stack
Lesson 19 The Maybe type: dealing with missing values	Lesson 27 The Functor type class	Lesson 36 Property testing with QuickCheck
Lesson 20 Capstone: Time series	Lesson 28 A peek at the Applicative type class: using functions in a context	Lesson 37 Capstone: Building a prime-number library
Unit 4 - IO IN HASKELL	Lesson 29 Lists as context: a deeper look at the Applicative type class	Unit 7 - PRACTICAL HASKELL
Lesson 21 Hello World!—introducing IO types	Lesson 30 Introducing the Monad type class	Lesson 38 Errors in Haskell and the Either type
Lesson 22 Interacting with the command line and lazy	Lesson 31 Making Monads easier	Lesson 39 Making HTTP requests in Haskell



Lesson 40 Working with JSON data by using Aeson  
Lesson 41 Using databases in Haskell  
Lesson 42 Efficient, stateful arrays in Haskell  
Afterword - What's next?  
Appendix - Sample answers to exercise  
*Functional Programming in JavaScript* Courier Corporation  
Learn how to think and write code like a functional programmer. With this practical guide, software developers familiar with object-oriented programming will dive into the core

concepts of functional programming and learn how to use both functional and OOP features together on large or complex software projects. Author Jack Widman uses samples from Java, Python, C#, Scala, and JavaScript to help you gain a new perspective and a set of tools for managing the complexity in your problem domain. You'll be able to write code that's simpler, reusable, easier to test and modify, and more consistently correct. This book also shows you

how to use patterns from category theory to help bridge the gap between OOP and functional programming. Learn functional programming fundamentals and explore the way functional programmers approach problems Understand how FP differs from object-oriented and imperative programming Use a set of practical, applicable design patterns that model reality in a functional way Learn how to incorporate FP and OOP features into software projects Apply functional

design patterns appropriately and use them to write correct, robust, and easily modifiable code.

*Functional Programming in R* "O'Reilly Media, Inc."

If you're considering R for statistical computing and data visualization, this book provides a quick and practical guide to just about everything you can do with the open source R language and software environment. You'll learn how to write R functions and use R packages to help you prepare, visualize, and analyze

data. Author Joseph Adler illustrates each process with a wealth of examples from medicine, business, and sports. Updated for R 2.14 and 2.15, this second edition includes new and expanded chapters on R performance, the ggplot2 data visualization package, and parallel R computing with Hadoop. Get started quickly with an R tutorial and hundreds of examples. Explore R syntax, objects, and other language details. Find thousands of user-contributed R packages online, including

Bioconductor. Learn how to use R to prepare data for analysis. Visualize your data with R's graphics, lattice, and ggplot2 packages. Use R to calculate statistical tests, fit models, and compute probability distributions. Speed up intensive computations by writing parallel R programs for Hadoop. Get a complete desktop reference to R.

## **GET PROGRAMMING WITH HASKELL**

Addison-Wesley Professional  
Software development

today is embracing functional programming (FP), whether it's for writing concurrent programs or for managing Big Data. Where does that leave Java developers? This concise book offers a pragmatic, approachable introduction to FP for Java developers or anyone who uses an object-oriented language. Dean Wampler, Java expert and author of *Programming Scala* (O'Reilly), shows you how to apply FP principles such as immutability, avoidance of side-effects, and higher-order

functions to your Java code. Each chapter provides exercises to help you practice what you've learned. Once you grasp the benefits of functional programming, you'll discover that it improves all of the code you write. Learn basic FP principles and apply them to object-oriented programming. Discover how FP is more concise and modular than OOP. Get useful FP lessons for your Java type design—such as avoiding nulls. Design data structures and algorithms using functional

programming principles. Write concurrent programs using the Actor model and software transactional memory. Use functional libraries and frameworks for Java—and learn where to go next to deepen your functional programming skills. [Advanced Statistical Programming for Data Science, Analysis and Finance](#) "O'Reilly Media, Inc." Well-respected text for computer science students provides an accessible introduction to functional programming.

Cogent examples illuminate the central ideas, and numerous exercises offer reinforcement. Includes solutions. 1989 edition. *Pragmatic Functional Programming* Pragmatic Bookshelf

Grokking Functional Programming is a practical book written especially for object-oriented programmers. Grokking Functional Programming teaches you first to break down problems in a new way so you can approach them from a FP mindset.

Following carefully-selected examples with thorough, carefully-paced explanations, you'll immerse yourself in FP concept by concept. Along the way, exercises, checks for understanding, and even the occasional puzzler give you opportunities to think and practice what you're learning. Grokking Functional Programming is a practical book written especially for object-oriented programmers. It will help you map familiar ideas like objects and composition to FP

concepts such as programming with immutable data and higher-order functions. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

*Metaprogramming in R* "O'Reilly Media, Inc."

If you're a developer with core Java SE skills, this hands-on book takes you through the language changes in Java 8 triggered by the addition of lambda expressions. You'll learn through code examples, exercises, and

fluid explanations how these anonymous functions will help you write simple, clean, library-level code that solves business problems. Lambda expressions are a fairly simple change to Java, and the first part of the book shows you how to use them properly. Later chapters show you how lambda functions help you improve performance with parallelism, write simpler concurrent code, and model your domain more accurately, including building better DSLs. Use

exercises in each chapter to help you master lambda expressions in Java 8 quickly Explore streams, advanced collections, and other Java 8 library improvements Leverage multicore CPUs and improve performance with data parallelism Use techniques to “lambdify” your existing codebase or library code Learn practical solutions for lambda expression unit testing and debugging Implement SOLID principles of object-oriented programming with lambdas Write

concurrent applications that efficiently perform message passing and non-blocking I/O

**Grokking Functional Programming** Springer Nature

Provides information on the basics of the Ruby scripting language and how to create scripts using test-driven design.

Tools for Better Concurrency, Abstraction, and Agility Springer

Get up to speed on Scala, the JVM language that offers all the benefits of a modern object model, functional programming,

and an advanced type system. Packed with code examples, this comprehensive book shows you how to be productive with the language and ecosystem right away, and explains why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. This second edition covers recent language features, with new chapters on pattern matching, comprehensions, and advanced functional programming. You'll also

learn about Scala's command-line tools, third-party tools, libraries, and language-aware plugins for editors and IDEs. This book is ideal for beginning and advanced Scala developers alike. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big-data apps, using Scala's functional combinators Use traits for mixin composition and pattern matching for data extraction Learn the

sophisticated type system that combines FP and object-oriented programming concepts Explore Scala-specific concurrency tools, including Akka Understand how to develop rich domain-specific languages Learn good design techniques for building scalable and robust Scala applications *Programming Scala* Prentice Hall Intermediate level, for programmers fairly familiar with Java, but new to the functional style of programming and lambda

expressions. Get ready to program in a whole new way. Functional Programming in Java will help you quickly get on top of the new, essential Java 8 language features and the functional style that will change and improve your code. This short, targeted book will help you make the paradigm shift from the old imperative way to a less error-prone, more elegant, and concise coding style that's also a breeze to parallelize. You'll explore the syntax and semantics of lambda

expressions, method and constructor references, and functional interfaces. You'll design and write applications better using the new standards in Java 8 and the JDK. Lambda expressions are lightweight, highly concise anonymous methods backed by functional interfaces in Java 8. You can use them to leap forward into a whole new world of programming in Java. With functional programming capabilities, which have been around for decades in other languages, you can now

write elegant, concise, less error-prone code using standard Java. This book will guide you through the paradigm change, offer the essential details about the new features, and show you how to transition from your old way of coding to an improved style. In this book you'll see popular design patterns, such as decorator, builder, and strategy, come to life to solve common design problems, but with little ceremony and effort. With these new capabilities in hand, Functional

Programming in Java will help you pick up techniques to implement designs that were beyond easy reach in earlier versions of Java. You'll see how you can reap the benefits of tail call optimization, memoization, and effortless parallelization techniques. Java 8 will change the way you write applications. If you're eager to take advantage of the new features in the language, this is the book for you. What you need: Java 8 with support for lambda expressions and

the JDK is required to make use of the concepts and the examples in this book.

### **Bridging the Divide Between Opposing Paradigms**

Advanced R Summary Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to the everyday business of coding. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples

and exercises that open up the world of functional programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Functional programming (FP) is a style of software development emphasizing functions that don't depend on program state. Functional code is easier to test and reuse, simpler to parallelize, and less prone to bugs than other code. Scala is an emerging JVM language that offers strong support



for FP. Its familiar syntax and transparent interoperability with Java make Scala a great place to start learning FP. About the Book Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to their everyday work. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. This book

assumes no prior experience with functional programming. Some prior exposure to Scala or Java is helpful. What's Inside Functional programming concepts The whys and hows of FP How to write multicore programs Exercises and checks for understanding About the Authors Paul Chiusano and Rúnar Bjarnason are recognized experts in functional programming with Scala and are core contributors to the Scalaz library. Table of Contents PART 1 INTRODUCTION TO FUNCTIONAL

PROGRAMMING What is functional programming? Getting started with functional programming in Scala Functional data structures Handling errors without exceptions Strictness and laziness Purely functional state PART 2 FUNCTIONAL DESIGN AND COMBINATOR LIBRARIES Purely functional parallelism Property-based testing Parser combinators PART 3 COMMON STRUCTURES IN FUNCTIONAL DESIGN Monoids Monads Applicative and

traversable functors PART  
4 EFFECTS AND I/O  
External effects and I/O  
Local effects and mutable  
state Stream processing  
and incremental I/O

## **BECOMING FUNCTIONAL**

Simon and Schuster  
If you have an imperative  
(and probably object-  
oriented) programming  
background, this hands-on  
book will guide you  
through the alien world of  
functional programming.  
Author Joshua Backfield  
begins slowly by showing  
you how to apply the

most useful  
implementation concepts  
before taking you further  
into functional-style  
concepts and practices. In  
each chapter, you'll learn  
a functional concept and  
then use it to refactor the  
fictional XXY company's  
imperative-style legacy  
code, writing and testing  
the functional code  
yourself. As you progress  
through the book, you'll  
migrate from Java 7 to  
Groovy and finally to  
Scala as the need for  
better functional language  
support gradually  
increases. Learn why

today's finely tuned  
applications work better  
with functional code  
Transform imperative-  
style patterns into  
functional code, following  
basic steps Get up to  
speed with Groovy and  
Scala through examples  
Understand how first-class  
functions are passed and  
returned from other  
functions Convert existing  
methods into pure  
functions, and loops into  
recursive methods  
Change mutable variables  
into immutable variables  
Get hands-on experience  
with statements and

nonstrict evaluations Use  
functional programming  
alongside object-oriented  
design

*Advanced Statistical  
Programming for Data  
Science, Analysis and  
Finance* CRC Press

In large projects,  
programmers tend to get  
overwhelmed by their  
complexity. It can be hard  
to keep track of all the  
interdependencies in the  
code-base and how its

state changes on runtime.  
The solution to these  
problems is Functional  
Programming, a paradigm  
specifically designed to  
deal with the complexity  
of software development.  
Mastering ...

Related with Functional Programming For The Object Oriented Programmer Ebook  
Brian Marick:

[© Functional Programming For The Object Oriented Programmer Ebook Brian Marick  
Sympathetic Tone In Writing](#)

[© Functional Programming For The Object Oriented Programmer Ebook Brian Marick  
Symbolic Representation In Math](#)

[© Functional Programming For The Object Oriented Programmer Ebook Brian Marick  
Synthes Tfn Technique Guide](#)