Functional Programming For The Object Oriented Programmer Ebook Brian Marick

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 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. 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 own 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 structures. In the book you'll see how these 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 OO 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 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 p 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' fit in the print version, they have been made freely available online. (Alvin A 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.)

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 p everyday world. (The 8O 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. change in perspective can be worth 8O 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 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 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 Capstone: Secret messages! Unit 3 - PROGRAMMING IN TYPES Lesson 16 Creating types with "and" and "or" Lesson 17 Design by composition—Semigroups and Monoids Lesson 18 Parameterized types Lesson 19 The Maybe type: dealing with missing values Lesson 2O Capstone: Time series Unit 4 - IO IN HASKELL Lesson 21 Hello World!—int Interacting with the command line and lazy I/O Lesson 24 Working with files Lesson 25 Working with binary data Lesson 26 Capstone: Processing binary files and book data Unit 5 - WORKING WITH TYPE IN A CONTEXT Lesson 27 The Functor type class Lesson 28 A peek at the Applicative type class Lesson 29 Lists as context: a deeper look at the Applicative type class Lesson 3O Introducing the Monad type class Lesson 31 Making Monads easier with donotation Lesson 32 The list monad and list comprehensions Lesson 33 Capstone: SQL-like queries in Haskell Unit 6 - ORGANIZING CODE AND BUILDING PROJECTS Lesson 34 Organizing 35 Building projects with stack Lesson 36 Property testing with QuickCheck Lesson 37 Capstone: Building a prime-number library Unit 7 - PRACTICAL HASKELL Lesson 38 Errors in Haskell and the Either type Lesson 39 Making HTTP requests in Haskell Lesson 4O Working with JSON data by using Aeson Lesson 41 Using databases in Haskell arrays in Haskell Afterword - What's next? Appendix - Sample answers to exercise

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 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 programmingObject compositionHow to work with composit functionsFunctors (e.g., array.map)Monads (e.g., promises)TransducersLensesAll 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 frameworks of today. Unlike most programming books, this one may still be relevant 2O 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 5OO companies

In this "OBJECT ORIENTED AND FUNCTIONAL PROGRAMMING" book we will see the concepts of OOP and Functional Programming and we will analyze some functions with examples in C# and Python.First, it should be clarified that it is not the objective of this book to deepen or teach about the fundamental Functional Programming and see examples of it in those programming languages.We will observe the advantages, disadvantages and characteristics of these paradigms and we will see the Principles of Functional Programming with examples of lambda, closures, filter, map, reduce and about OOP we will learn their principles. We will also see relationship between OOP and FP.Easy. You don't need to know about it. This book provides basic concepts, for beginners, on OOP and Functional Programming.It is not essential that you have previous knowledge although it is necessary that you have some knowledge of computer science and programming in general (if it is C# or Python b Fundamentals of Programming will be provided.
Object-oriented Vs. Functional Programming
Scalability = Functional Programming + Objects
Functional Programming in JavaScript
Scala: From a Functional Programming Perspective
Functional Programming in Java
Learning Functional Programming
Scala Cookbook

In this book, you will find how to use JavaScript as a functional programming language. It turns out that JavaScript has everything it needs to be used as a functionallanguage. We just have to remove features from the language starting with the 'this' keyword. Functions are values. Functions can operate on other functions. Inner functions can access variables from the outer functions even after the outer functions have executed. Functional programming makes code easier to read, understand, test, and debug. Here are some of the things you will learn: How to disable 'this' and enable immutable data objects using a linter How to work with immutable objects and collections How to do data transformations using core operations like filter, map, sort, or reduce How to use statements like if and switch in a functional way How to create pipelines and use currying to pass additional data How to create and use functors and monads How to work with promises and observables Understand the Elm Architecture

This is Volume 7 of Trends in Functional Programming (TFP). It contains a refereed selection of the papers that were presented at TFP 2006: the Seventh Symposium on Trends in Functional Programming. which took place in Nottingham, 19-21 April, 2006.TFP is an international forum for researchers from all functional programming communities spanning the entire width of topics in the field. Its goal is to provide a broad view of current and future trends in functional programming in a lively and friendly setting, thus promoting new research directions related to the field of functional programming and other fields of computer science.True to the spirit of TFP, the selection of papers in this volume covers a wide range of topics, including dependently typed programming, generic programming, purely functional data structures, function synthesis, declarative debugging, implementation of functional programming languages, and memory management. A particular emerging trend is that of dependently typed programming, reflected by a number of papers in the present selection and by the co-location of TFP and Types 2006.

Scala is a highly expressive, concise and scalable language. It is also the most prominent method of the new and exciting methodology known as object-functional programming. In this book, the authors show how Scala grows to the needs of the programmer, whether professional or hobbyist. They teach Scala with a step-by-step approach and explain how to exploit the full power of the industry-proven JVM technology. Readers can then dive into specially chosen design challenges and implementation problems, inspired by the trials of real-world software engineering. It also helps readers to embrace the power of static typing and automatic type inference. In addition, the book shows how to use the dual-object and functional-oriented natures combined at Scala's core, and so write code that is less 'boilerplate', giving a genuine increase in productivity.

Get up to speed on Scala--the JVM, JavaScript, and natively compiled language that offers all the benefits of functional programming, a modern object model, 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. You'll learn why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. Despite the reinvigoration of Java and the introduction of Kotlin, Scala hasn't been sitting still. This third edition covers the new features in Scala 3.0, with updates throughout the book. Programming Scala is ideal for beginning to advanced developers who want a complete understanding of Scala's design philosophy and features with a thoroughly practical focus. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big data and distributed apps, using Scala's functional combinators and tools including Akka and Spark Use traits for mixin composition and pattern matching for data extraction Learn the sophisticated type system that combines functional programming and object-oriented programming concepts
Everyday Scripting with Ruby
Functional Programming in C++
A Desktop Quick Reference
Metaprogramming in R
Mastering Functional Programming
Java 8 Lambdas
R in a Nutshell

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

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

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

Functional programming languages like F#, Erlang, and Scala are attractingattention as an efficient way to handle the new requirements for programmingmulti-processor and high-availability applications. Microsoft's new F# is a truefunctional language and C# uses functional language features for LINQ andother recent advances. Real-World Functional Programming is a unique tutorial that explores thefunctional programming model through the F# and C# languages. The clearlypresented ideas and examples teach readers how functional programming differsfrom other approaches. It explains how ideas look in F#-a functionallanguage-as well as how they can be successfully used to solve programmingproblems in C#. Readers build on what they know about .NET and learn wherea functional approach makes the most sense and how to apply it effectively inthose cases. The reader should have a good working knowledge of C#. No prior exposure toF# or functional programming is required. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book.
Write Lean Programs for the JVM
An Introduction to Functional Programming Through Lambda Calculus
Object Oriented and Functional Programming
Composing Software
(Scala Edition)
Learning Java Functional Programming
The Little LISPer

How can you overcome JavaScript language oddities and unsafe features? With this book, you'll learn how to create code that's beautiful, safe, and simple to understand and test by using JavaScript's functional programming support. Author Michael Fogus shows you how to apply functional-style concepts with Underscore.js, a JavaScript library that facilitates functional programming techniques. Sample code is available on GitHub at https://github.com/funjs/book-source. Fogus helps you think in a functional way to help you minimize complexity in the programs you build. If you're a JavaScript programmer hoping to learn functional programming techniques, or a functional programmer looking to learn JavaScript, this book is the ideal introduction. Use applicative programming with first-class functions Understand how and why you might leverage variable scoping and closures Delve into higher-order functions—and learn how they take other functions as arguments for maximum advantage Explore ways to compose new functions from existing functions Get around JavaScript's limitations for using recursive functions Reduce, hide, or eliminate the footprint of state change in your programs Practice flow-based programming with chains and functional pipelines Discover how to code without using classes

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 fests, 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 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

Summary Functional Programming in C++ teaches developers the practical side of functional programming and the tools that C++ provides to develop software in the functional style. This in-depth guide is full of useful diagrams that help you understand FP concepts and begin to think functionally. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Well-written code is easier to test and reuse, simpler to parallelize, and less error prone. Mastering the functional style of programming can help you tackle the demands of modern apps and will lead to simpler expression of complex program logic, graceful error handling, and elegant concurrency. C++ supports FP with templates, lambdas, and other core language features, along with many parts of the STL. About the Book Functional Programming in C++ helps you unleash the functional side of your brain, as you gain a powerful new perspective on C++ coding. You'll discover dozens of examples, diagrams, and illustrations that break down the functional concepts you can apply in C++, including lazy evaluation, function objects and invokables, algebraic data types, and more. As you read, you'll match FP techniques with practical scenarios where they offer the most benefit. What's inside Writing safer code with no performance penalties Explicitly handling errors through the type system Extending C++ with new control structures Composing tasks with DSLs About the Reader Written for developers with two or more years of experience coding in C++. About the Author Ivan Čukić is a core developer at KDE and has been coding in C++ since 1998. He teaches modern C++ and functional programming at the Faculty of Mathematics at the University of Belgrade. Table of Contents Introduction to functional programming Getting started with functional programming Function objects Creating new functions from the old ones Purity: Avoiding mutable state Lazy evaluation Ranges Functional data structures Algebraic data types and pattern matching Monads Template metaprogramming Functional design for concurrent systems Testing and debugging

Advanced Statistical Programming for Data Science, Analysis and Finance

Functional Python Programming

Real-World Functional Programming

Functional techniques for sequential and parallel programming with Scala

Steps in Scala

Discover the power of functional programming, generator functions, lazy evaluation, the built-in itertools library, and monads, 2nd Edition

Functional JavaScript

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 though 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.

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

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.

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 simplyfy 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 slowly 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.

Tools for Better Concurrency, Abstraction, and Agility

Get Programming with Haskell

An Introduction to the Programming Language

Functional Programming in Scala

Unlearning objects

A Beginner's Guide to Scala, Object Orientation and Functional Programming

A Modern Introduction to Programming

If you're familiar with functional programming basics and want to gain a much deeper understanding, this in-depth guide takes you beyond syntax and demonstrates how you need to think in a new way. Software architect Neal Ford shows intermediate to advanced developers how functional coding allows you to step back a level of abstraction so you can see your programming problem with greater clarity. Each chapter shows you various examples of functional thinking, using numerous code examples from Java 8 and other JVM languages that include functional capabilities. This book may bend your mind, but you'll come away with a much better grasp of functional programming concepts. Understand why many imperative languages are adding functional capabilities Compare functional and imperative solutions to common problems Examine ways to cede control of routine chores to the runtime Learn how memoization and laziness eliminate hand-crafted solutions Explore functional approaches to design patterns and code reuse View real-world examples of functional thinking with Java 8, and in functional architectures and web frameworks Learn the pros and cons of living in a paradigmatically richer world If you're new to functional programming, check out Josh Backfield's book Becoming Functional.

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

Provides a guide to using Scala and Clojure to solve in-depth programming problems.

Eliminate the unavoidable complexity of object-oriented designs. Using the persistent data structures built into most modern programming languages, Data-oriented programming cleanly separates code and data, which simplifies state management and eases concurrency. Data-Oriented Programming teaches you to design and implement software using the data-oriented programming paradigm. In it, you'll learn author Yehonathan Sharvit's unique approach to DOP that he has developed over a decade of experience. Every chapter contains a new light bulb moment that will change the way you think about programming. As you read, you'll build a library management system using the DOP paradigm. You'll design data models for business entities, manipulate immutable data collections, and write unit tests for data-oriented systems. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Eloquent JavaScript, 3rd Edition

Functional Programming for Java Developers

Pragmatic Functional Programming

Steps for Transforming Into a Functional Programmer

Examples in C# and Python

Discover Functional JavaScript

Functional Programming, Simplified

Software -- Programming Techniques.

Completely revised and updated, this best-selling introduction to programming in JavaScript focuses on writing real applications. JavaScript lies at the heart of almost every modern web application, from social apps like Twitter to browser-based game frameworks like Phaser and Babylon. Though simple for beginners to pick up and play with, JavaScript is a flexible, complex language that you can use to build full-scale applications. This much anticipated and thoroughly revised third edition of Eloquent JavaScript dives deep into the JavaScript language to show you how to write beautiful, effective code. It has been updated to reflect the current state of JavaScript and web browsers and includes brand-new material on features like class notation, arrow functions, iterators, async functions, template strings, and block scope. A host of new exercises have also been added to test your skills and keep you on track. As with previous editions, Haverbeke continues to teach through extensive examples and immerses you in code from the start, while exercises and full-chapter projects give you hands-on experience with writing your own programs. You start by learning the basic structure of the JavaScript language as well as control structures, functions, and data structures to help you write basic programs. Then you'll learn about error handling and bug fixing, modularity, and asynchronous programming before moving on to web browsers and how JavaScript is used to program them. As you build projects such as an artificial life simulation, a simple programming language, and a paint program, you'll learn how to: - Understand the essential elements of programming, including syntax, control, and data - Organize and clarify your code with object-oriented and functional programming techniques - Script the browser and make basic web applications - Use the DOM effectively to interact with browsers - Harness Node.js to build servers and utilities Isn't it time you became fluent in the language of the Web? * All source code is available online in an inter¬active sandbox, where you can edit the code, run it, and see its output instantly.

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.

Save time and trouble building object-oriented, functional, and concurrent applications with Scala. The latest edition of this comprehensive cookbook is packed with more than 250 ready-to-use recipes and 1,000 code examples to help you solve the most common problems when working with Scala 3 and its popular libraries. Scala changes the way you think about programming--and that's a good thing. Whether you're working on web, big data, or distributed applications, this cookbook provides recipes based on real-world scenarios for both experienced Scala developers and programmers just learning to use this JVM language. Author Alvin Alexander includes practical solutions from his experience using Scala for component-based, highly scalable applications that support concurrency and distribution. Recipes cover: Strings, numbers, and control structures Classes, methods, objects, traits, packaging, and imports Functional programming techniques Scala's wealth of collections classes and methods Building and publishing Scala applications with sbt Actors and concurrency with Scala Future and Akka Typed Popular libraries, including Spark, Scala.js, Play Framework, and GraalVM Types, such as variance, givens, intersections, and unions Best practices, including pattern matching, modules, and functional error handling

For Teams, Testers, and You

Paradigm Over Syntax

Trends in Functional Programming

Harnessing the Power Of Java 8 Lambda Expressions

How to improve your JavaScript programs using functional techniques

An Overview of Functional and Object Oriented Programming in JavaScript

Functional Programming Patterns in Scala and Clojure

Master functions and discover how to write functional programs in R. In this concise book, you'll make your functions pure by avoiding side-effects; you'll write functions that manipulate other functions, and you'll construct complex functions using simpler functions as building blocks. In Functional Programming in R, you'll see how we can replace loops, which can have side-effects, with recursive functions that can more easily avoid them. In addition, the book covers why you shouldn't use recursion when loops are more efficient and how you can get the best of both worlds. Functional programming is a style of programming, like object-oriented programming, but one that focuses on data transformations and calculations rather than objects and state. Where in object-oriented programming you model your programs by describing which states an object can be in and how methods will reveal or modify that state, in functional programming you model programs by describing how functions translate input data to output data. Functions themselves are considered to be data you can manipulate and much of the strength of functional programming comes from manipulating functions; that is, building more complex functions by combining simpler functions. What You'll Learn Write functions in R including infix operators and replacement functions Create higher order functions Pass functions to other functions and start using functions as data you can manipulate Use Filer, Map and Reduce functions to express the intent behind code clearly and safely Build new functions from existing functions without necessarily writing any new functions, using point-free programming Create functions that carry data along with them Who This Book Is For Those with at least some experience with programming in R.

Create succinct and expressive implementations with functional programming in Python Key Features Learn how to choose between imperative and functional approaches based on expressiveness, clarity, and performance Get familiar with complex concepts such as monads, concurrency, and immutability Apply functional Python to common

Exploratory Data Analysis (EDA) programming problems Book Description If you're a Python developer who wants to discover how to take the power of functional programming (FP) and bring it into your own programs, then this book is essential for you, even if you know next to nothing about the paradigm. Starting with a general overview of functional concepts, you'll explore common functional features such as first-class and higher-order functions, pure functions, and more. You'll see how these are accomplished in Python 3.6 to give you the core foundations you'll build upon. After that, you'll discover common functional optimizations for Python to help your apps reach even higher speeds. You'll learn FP concepts such as lazy evaluation using Python's generator functions and expressions. Moving forward, you'll learn to design and implement decorators to create composite functions. You'll also explore data preparation techniques and data exploration in depth, and see how the Python standard library fits the functional programming model. Finally, to top off your journey into the world of functional Python, you'll at look at the PyMonad project and some larger examples to put everything into perspective. What you will learn Use Python's generator functions and generator expressions to work with collections in a non-strict (or lazy) manner Utilize Python library modules including itertools, functools, multiprocessing, and concurrent features to ensure efficient functional programs Use Python strings with object-oriented suffix notation and prefix notation Avoid stateful classes with families of tuples Design and implement decorators to create composite functions Use functions such as max(), min(), map(), filter(), and sorted() Write higher-order functions Who this book is for This book is for Python developers who would like to perform Functional programming with Python. Python Programming knowledge is assumed.

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 ...

This book gives an introduction to the programming language Scala. It presents it from a functional programming perspective. The book explains with detail functional programming and recursivity, and includes chapters on lazy and eager evaluation, streams, higher-order functions (including map, fold, reduce, and aggregate), and algebraic data types. The book also describes the object-oriented aspects of Scala, as they are a fundamental part of the language. In addition, the book includes a chapter on parallelism in Scala, giving an overview of the actor model.

Functional Programming in R
An Exploration of Functional Programming and Object Composition in JavaScript
An Introduction to Object-Functional Programming
Introducing Functional Programming with Underscore.js
Elements of Functional Programming
Becoming Functional
With examples in F# and C#

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.

JavaScript is the first language to bring Functional Programming to the mainstream. At the same time, it offers a new way of doing Object Oriented Programming without classes and prototypes. Programming in a functional style means to use concepts such as first-class functions, closures, higher-order functions, partial application, currying, immutability or pure functions. Pure Functional Programming promises to make code easier to read, understand, test, debug or compose. Can it deliver its promise? If it can, can we build an application using only pure functions? Decorators are a tool for reusing common logic and creating variations of existing functions. Closure can encapsulate state. Multiple closures sharing the same private state can create flexible and encapsulated objects. "One of the best new Functional Programming books" - BookAuthority

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

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.

Programming Scala
Data-Oriented Programming
Grokking Functional Programming
Advanced R
Bridging the Divide Between Opposing Paradigms
Functional Thinking

Scala is now an established programming language developed by Martin Odersky and his team at the EPFL. The name Scala is derived from Sca(lable) La(nguage). Scala is a multi-paradigm language, incorporating object oriented approaches with functional programming. Although some familiarity with standard computing concepts is assumed (such as the idea of compiling a program and executing this compiled from etc.) and with basic procedural language concepts (such as variables and allocation of values to these variables) the early chapters of the book do not assume any familiarity with object orientation nor with functional programming These chapters also step through other concepts with which the reader may not be familiar (such as list processing). From this background, the book provides a practical introduction to both object and functional approaches using Scala. These concepts are introduced through practical experience taking the reader beyond the level of the language syntax to the philosophy and practice of object oriented development and functional programming. Students and those actively involved in the software industry will find this comprehensive introduction to Scala invaluable.

An Essential Reference for Intermediate and Advanced R Programmers Advanced R presents useful tools and techniques for attacking many types of R programming problems, helping you avoid mistakes and dead ends. With more than ten years of experience programming in R, the author illustrates the elegance, beauty, and flexibility at the heart of R. The book develops the necessary skills to produce quality code that can be used in a variety of circumstances. You will learn: The fundamentals of R, including standard data types and functions Functional programming as a useful framework for solving wide classes of problems The positives and negatives of metaprogramming How to write fast, memory-efficient code This book not only helps current R users become R programmers but also shows existing programmers what's special about R. Intermediate R programmers can dive deeper into R and learn new strategies for solving diverse problems while programmers from other languages can learn the details of R and understand why R works the way it does.