Functional Programming For The Object Oriented Programmer Ebook Brian Marick

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!

Unlock the powers of functional programming hidden within JavaScript to build smarter, cleaner, and more reliable web apps About This Book Write powerful code with the high-level functions that JavaScript offers Discover what functional programming is, why it's effective, and how it's used in JavaScript Understand and optimize JavaScript's hidden potential as a true functional language Who This Book Is For If you are a JavaScript developer interested in learning functional programming, looking for the quantum leap toward mastering the JavaScript language, or just want to become a better programmer in general, then this book is ideal for you. This guide is aimed at programmers, involved in developing reactive frontend apps, server-side apps that wrangle with reliability and concurrency, and everything in between. What You Will Learn Get a run through of the basic JavaScript language constructs Code using the powerful object-oriented feature in JavaScript Master DOM manipulation, cross-browser strategies, and ES6 Understand the basic concurrency constructs in Javascript and best performance strategies Harness the power of patterns for tasks ranging from application building to code testing Build large-scale apps seamlessly with the help of reactive patterns Explore advanced design patterns, including dependency injection Develop more

powerful applications with currying and function composition Create more reliable code with closures and immutable data In Detail JavaScript is a high-level, dynamic, untyped, lightweight, and interpreted programming language and functional programming is a style that emphasizes and enables smarter code that minimizes complexity and increases modularity. It's a way of writing cleaner code through clever ways of mutating, combining, and using functions. And JavaScript provides an excellent medium for this approach. By learning how to expose JavaScript's true identity as a functional language, we can implement web apps that are more powerful, easier to maintain and more reliable. The java script: Functional Programming for JavaScript Developers course will take you on a journey to show how functional programming when combined with other techniques makes JavaScript programming more efficient. The first module Mastering JavaScript, stress on practical aspects of Javascript development like—Functions and Closures, Runtime debugging techniques, project layout, events and DOM processing, build tools, Object-oriented patterns, isomorphism—everything that a modern Javascript project would need. The second module, Mastering JavaScript Design Patterns - Second Edition, will explore how design patterns can help you improve and organize your JavaScript code. You'll get to grips with creational, structural, and behavioral patterns as you discover how to put them to work in different scenarios. This updated edition will also delve into reactive design patterns and microservices as they are a growing phenomenon in the world of

web development. It will also show you some advanced patterns, including dependency injection and live post processing. The third module, Functional Programming in JavaScript, will help you to write real-world applications by utilizing a wide range of functional techniques and styles. It explores the core concepts of functional programming common to all functional languages, with examples of their use in JavaScript. Style and approach This course will begin with providing insights and practical tips on advanced JavaScript features to build highly scalable web and mobile system and move on to some design patterns with JavaScript. Finally, the course ends with presenting the functional programming techniques and styles in JavaScript.

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 programmingObject compositionHow to work with composite data structuresClosuresHigher order 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 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

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

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.

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

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.

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

Save time and trouble building object-oriented, functional, and concurrent applications with Scala 3. The latest edition of this comprehensive cookbook is packed with more than 250 ready-to-use recipes and 700 code examples to help you solve the most common problems when working with Scala and its popular libraries. Whether you're working on web, big data, or distributed applications, this cookbook provides recipes based on real-world scenarios for experienced Scala developers and for programmers just learning to use this JVM language. Author Alvin Alexander includes practical solutions from his experience using Scala for highly scalable applications that support concurrency and distribution. Recipes cover: Strings, numbers, and control structures Classes, methods, objects, traits, packaging, and imports Functional programming in a variety of situations Building Scala applications with sbt Collections covering Scala's wealth of classes and methods Actors and concurrency List, array, map, set, and more Files, processes, and command-line tasks Web services and interacting with Java Databases and persistence, data types and idioms

JavaScript is at the heart of almost every modern Web application, whether it's Google Apps, Twitter, or the newest browser-based game. Though it's simple for beginners to pick up and play with, JavaScript is not a toy—it's a flexible and complex language that can be used to build full-scale applications. Eloquent JavaScript dives into this flourishing language and teaches

you to write code that's beautiful and effective. By immersing you in example code and encouraging experimentation right from the start, the author quickly gives you the tools you need to build your own programs. As you follow along with examples like an artificial life simulation and a version of the classic game Sokoban, you'll learn to: –Understand the essential elements of programming: syntax, control, and data –Use object-oriented and functional programming techniques to organize and clarify your programs –Script the browser and make basic Web applications –Work with tools like regular expressions and XMLHttpRequest objects And since programming is an art that's best learned by doing, all example code is available online in an interactive sandbox for you to experiment with. With Eloquent JavaScript as your guide, you can tweak, expand, and modify the author's code, or throw it away and build your own creations from scratch. Before you know it, you'll be fluent in the language of the Web.

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 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.

This book provides a distinct way to teach discrete mathematics. Since discrete mathematics is crucial for rigorous study in computer science, many texts include applications of mathematical topics to computer science or have selected topics of particular interest to computer science. This text fully integrates discrete mathematics with .....

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 RCRC Press

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.

Practical FP in Scala: A hands-on approach, is a book for intermediate to advanced Scala developers. Aimed at those who understand functional effects, referential transparency and the benefits of functional programming to some extent but who are missing some pieces to put all

these concepts together to build a large application in a time-constrained manner.Throughout the chapters we will design, architect and develop a complete stateful application serving an API via HTTP, accessing a database and dealing with cached data, using the best practices and best functional libraries available in the Cats ecosystem.You will also learn about common design patterns such as managing state, error handling and anti-patterns, all accompanied by clear examples. Furthermore, at the end of the book, we will dive into some advanced concepts such as MTL, Classy Optics and Typeclass derivation.

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

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 techniques 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

Software -- Programming Techniques.

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.

Scala is now an established programming language developed by Martin Oderskey 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.

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.
Learn how to write scalable and concurrent programs in Scala, a language that grows with you. Key Features Get a grip on the functional features of the Scala programming language Understand and develop optimal applications using object-oriented and functional Scala constructs Learn reactive principles with Scala and work with the Akka framework Book Description Scala is a general-purpose programming language that supports both functional and object-oriented programming paradigms. Due to its concise design and versatility, Scala's applications have been extended to a wide variety of fields such as data science and cluster computing. You will learn to write highly scalable, concurrent, and testable programs to meet everyday software requirements. We will begin by understanding the language basics, syntax, core data types, literals, variables, and more. From here you will be introduced to data structures with Scala and you will learn to work with higher-order functions. Scala's powerful collections framework will help you get the best out of immutable data structures and utilize them effectively. You will then be introduced to concepts such as pattern matching, case classes, and functional programming features. From here, you will learn to work with Scala's object-oriented features. Going forward, you will learn about asynchronous and reactive programming with Scala, where you will be introduced to the Akka framework. Finally, you will learn the interoperability of Scala and Java. After reading this book, you'll be well versed with this language and its features, and you will be able to write scalable, concurrent, and reactive programs in Scala. What you will learn Get to know the reasons for choosing Scala: its use and the advantages it provides over other languages Bring together functional and object-oriented

programming constructs to make a manageable application Master basic to advanced Scala constructs Test your applications using advanced testing methodologies such as TDD Select preferred language constructs from the wide variety of constructs provided by Scala Make the transition from the object-oriented paradigm to the functional programming paradigm Write clean, concise, and powerful code with a functional mindset Create concurrent, scalable, and reactive applications utilizing the advantages of Scala Who this book is for This book is for programmers who choose to get a grip over Scala to write concurrent, scalable, and reactive programs. No prior experience with any programming language is required to learn the concepts explained in this book. Knowledge of any programming language would help the reader understanding concepts faster though.

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 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.

If your application source code is overly verbose, it can be a nightmare to maintain. Write concise and expressive, type-safe code in an environment that lets you build for the JVM, browser, and more. Key Features Expert guidance that shows you to efficiently use both object-oriented and functional programming techniques Understand functional programming libraries, such as Cats and

Scalaz, and use them to augment your Scala development Perfectly balances theory and hands-on exercises, assessments, and activities Book Description This book teaches you how to build and contribute to Scala programs, recognizing common patterns and techniques used with the language. You'll learn how to write concise, functional code with Scala. After an introduction to core concepts, syntax, and writing example applications with scalac, you'll learn about the Scala Collections API and how the language handles type safety via static types out-of-the-box. You'll then learn about advanced functional programming patterns, and how you can write your own Domain Specific Languages (DSLs). By the end of the book, you'll be equipped with the skills you need to successfully build smart, efficient applications in Scala that can be compiled to the JVM. What you will learn Understand the key language syntax and core concepts for application development Master the type system to create scalable type-safe applications while cutting down your time spent debugging Understand how you can work with advanced data structures via built-in features such as the Collections library Use classes, objects, and traits to transform a trivial chatbot program into a useful assistant Understand what are pure functions, immutability, and higher-order functions Recognize and implement popular functional programming design patterns Who this book is for This is an

ideal book for developers who are looking to learn Scala, and is particularly well suited for Java developers looking to migrate across to Scala for application development on the JVM.

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.

Functional programming is a very powerful programming paradigm that can help us to write better code. This book presents essential functional and reactive

programming concepts in a simplified manner using Typescript.

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.

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

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' fit in the print version, they have been made freely available online. (Alvin Alexander (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.)

Create robust and maintainable Java applications using the functional style of programmingAbout 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 applicationsWho This Book Is ForIf 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 programsIn DetailFunctional 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 approachIn 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.

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 ?uki? 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 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

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

Solve practical real-world problems using JavaScript and Node.js About This Book Learn the concepts of Node.js to gain a high-level understanding of the Node.js execution model Build an interactive web application with MongoDB and Redis and create your own JavaScript modules that work both on the client side and server side Familiarize yourself with the new features of Node.js and JavaScript with this exclusive step-by-step guide Who This Book Is For This book is for developers who want to learn JavaScript and Node.js. Previous experience with programming is desired, but no JavaScript or Node.js knowledge is required. The book focuses

mostly on web development, such as networking, serving dynamic pages, and real-time client-server communication. What You Will Learn Understand which problems Node.js best solves Write idiomatic JavaScript and Node.js code Build web applications and command-line tools Minimise complexity and efficiently solve difficult problems Test and deploy Node.js applications Work with persistent data Implement real-time client-server applications Integrate .NET and Node.js code In Detail Node.js is an open source, cross-platform runtime environment that allows you to use JavaScript to develop server-side web applications. This short guide will help you develop applications using JavaScript and Node.js, leverage your existing programming skills from .NET or Java, and make the most of these other platforms through understanding the Node.js programming model. You will learn how to build web applications and APIs in Node, discover packages in the Node.js ecosystem, test and deploy your Node.js code, and more. Finally, you will discover how to integrate Node.js and .NET code. Style and approach This is a step-by-step and practical guide to Node.js for .Net developers. It covers the fundamentals relating to typical applications. The focus is on providing the practical skills required to develop applications, with a summary of the key concepts covered.

Save time and trouble when using Scala to build object-oriented, functional, and concurrent applications. With more than 250 ready-to-use recipes and 700 code examples, this comprehensive cookbook covers the most common problems you'll encounter when using the Scala language, libraries, and tools. It's ideal not only for experienced Scala developers, but also for programmers learning to use this JVM language. Author Alvin Alexander (creator of DevDaily.com) provides solutions based on his experience using Scala for highly scalable,

component-based applications that support concurrency and distribution. Packed with real-world scenarios, this book provides recipes for: Strings, numeric types, and control structures Classes, methods, objects, traits, and packaging Functional programming in a variety of situations Collections covering Scala's wealth of classes and methods Concurrency, using the Akka Actors library Using the Scala REPL and the Simple Build Tool (SBT) Web services on both the client and server sides Interacting with SQL and NoSQL databases Best practices in Scala development

It's all in the name: Learn You a Haskell for Great Good! is a hilarious, illustrated guide to this complex functional language. Packed with the author's original artwork, pop culture references, and most importantly, useful example code, this book teaches functional fundamentals in a way you never thought possible. You'll start with the kid stuff: basic syntax, recursion, types and type classes. Then once you've got the basics down, the real black belt master-class begins: you'll learn to use applicative functors, monads, zippers, and all the other mythical Haskell constructs you've only read about in storybooks. As you work your way through the author's imaginative (and occasionally insane) examples, you'll learn to: –Laugh in the face of side effects as you wield purely functional programming techniques –Use the magic of Haskell's "laziness" to play with infinite sets of data –Organize your programs by creating your own types, type classes, and modules –Use Haskell's elegant input/output system to share the genius of your programs with the outside world Short of eating the author's brain, you will not find a better way to learn this powerful language than reading Learn You a Haskell for Great Good!