

Functional Programming In Swift Ebook Chris Eidhof

Haskell Programming makes Haskell as clear, painless, and practical as it can be, whether you're a beginner or an experienced hacker. Learning Haskell from the ground up is easier and works better. With our exercise-driven approach, you'll build on previous chapters such that by the time you reach the notorious Monad, it'll seem trivial.

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*.

THE #1 BESTSELLING BOOK ON OBJECTIVE-C 2.0 Programming in Objective-C 2.0 provides the new programmer a complete, step-by-step introduction to Objective-C, the primary language used to develop applications for the iPhone, iPad, and Mac OS X platforms. The book does not assume previous experience with either C or object-oriented programming languages, and it includes many detailed, practical examples of how to put Objective-C to use in your everyday iPhone/iPad or Mac OS X programming tasks. A powerful yet simple object-oriented programming language that's based on the C programming language, Objective-C is widely available not only on OS X and the iPhone/iPad platform but across many operating systems that support the gcc compiler, including Linux, Unix, and Windows systems. The second edition of this book thoroughly covers the latest version of the language, Objective-C 2.0. And it shows not only how to take advantage of the Foundation framework's rich built-in library of classes but also how to use the iPhone SDK to develop programs designed for the iPhone/iPad platform. Table of Contents 1 Introduction Part I: The Objective-C 2.0 Language 2 Programming in Objective-C 3 Classes, Objects, and Methods 4 Data Types and Expressions 5 Program Looping 6 Making Decisions 7 More on Classes 8 Inheritance 9 Polymorphism, Dynamic Typing, and Dynamic Binding 10 More on Variables and Data Types 11 Categories and Protocols 12 The Preprocessor 13 Underlying C Language Features Part II: The Foundation Framework 14 Introduction to the Foundation Framework 15 Numbers, Strings, and Collections 16 Working with Files 17 Memory Management 18 Copying Objects 19 Archiving Part III: Cocoa and the iPhone SDK 20 Introduction to Cocoa 21 Writing iPhone Applications Part IV: Appendixes A Glossary B Objective-C 2.0 Language Summary C Address Book Source Code D Resources

Elixir's straightforward syntax and this guided tour give you a clean, simple path to learn modern functional programming techniques. No previous functional programming experience required! This book walks you through the right concepts at the right pace, as you explore immutable values and explicit data transformation, functions, modules, recursive functions, pattern matching, high-order functions,

polymorphism, and failure handling, all while avoiding side effects. Don't board the Elixir train with an imperative mindset! To get the most out of functional languages, you need to think functionally. This book will get you there. Functional programming offers useful techniques for building maintainable and scalable software that solves today's difficult problems. The demand for software written in this way is increasing - you don't want to miss out. In this book, you'll not only learn Elixir and its features, you'll also learn the mindset required to program functionally. Elixir's clean syntax is excellent for exploring the critical skills of using functions and concurrency. Start with the basic techniques of the functional way: working with immutable data, transforming data in discrete steps, and avoiding side effects. Next, take a deep look at values, expressions, functions, and modules. Then extend your programming with pattern matching and flow control with case, if, cond, and functions. Use recursive functions to create iterations. Work with data types such as lists, tuples, and maps. Improve code reusability and readability with Elixir's most common high-order functions. Explore how to use lazy computation with streams, design your data, and take advantage of polymorphism with protocols. Combine functions and handle failures in a maintainable way using Elixir features and libraries. Learn techniques that matter to make code that lives harmoniously with the language. What You Need: You'll need a computer and Elixir 1.4 or newer version installed. No previous functional programming or Elixir experience is required. Some experience with any programming language is recommended.

If you've had trouble trying to learn Functional Programming (FP), you're not alone. In this book, Alvin Alexander -- author of the Scala Cookbook and former teacher of Java and Object-Oriented Programming (OOP) classes -- writes about his own problems in trying to understand FP, and how he finally conquered it. What he originally learned is that experienced FP developers are driven by two goals: to use only immutable values, and write only pure functions. What he later learned is that they have these goals as the result of another larger goal: they want all of their code to look and work just like algebra. While that sounds simple, it turns out that these goals require them to use many advanced Scala features -- which they often use all at the same time. As a result, their code can look completely foreign to novice FP developers. As Mr. Alexander writes, "When you first see their code it's easy to ask, 'Why would anyone write code like this?'" Mr. Alexander answers that "Why?" question by explaining the benefits of writing pure functional code. Once you understand those benefits -- your motivation for learning FP -- he shares five rules for programming in the book: All fields must be immutable ('val' fields). All functions must be pure functions. Null values are not allowed. Whenever you use an 'if' you must also use an 'else'. You won't create OOP classes that encapsulate data and behavior; instead you'll design data structures using Scala 'case' classes, and write pure functions that operate on those data structures. In the book you'll see how those five, simple rules naturally lead you to write pure, functional code that reads like algebra. He also shares one more Golden Rule for learning: Always ask "Why"? Lessons in the book include: How and why to write only pure functions Why pure function signatures are much more important than OOP method signatures Why recursion is a natural tool for functional programming, and how to write recursive algorithms Because the Scala 'for' expression is so important to FP, dozens of pages explain the details of how it works In the end you'll see that monads aren't that difficult because they're a natural extension of the Five Rules The book finishes with lessons on FP data modeling, and two main approaches for organizing your pure functions As Mr. Alexander writes, "In this book I take the time to explain all of the concepts that are used to write FP code in Scala. As I learned from my own experience, once you understand the Five Rules and the small concepts, you can understand Scala/FP." Please note that because of the limits on how large a printed book can be, the paperback version does not include all of the chapters that are in the Kindle eBook. The following lessons are not in the paperback version: Grandma's Cookies (a story about pure functions) The ScalaCheck lessons The Type Classes lessons The

appendices Because those lessons didn't fit in the print version, they have been made freely available online. (Alvin Alexander (alvinalexander.com) 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.)

Learn the basics of reactive programming and how it makes apps more responsive. This book shows you how to incorporate reactive programming into existing development products and cycles using RXSwift and RxCocoa on iOS and Mac. As we move away from the traditional paradigm of typing or touching one step at a time to interact with programs, users expect apps to adapt and not need constant hand-holding. People today expect their devices to do much more than just follow commands. They expect devices to react and adapt. Reactive programming, a new term for asynchronous processing, requires new app architectures, and you'll learn how these are already built into iOS and macOS in many places. As part of this more complex environment, you'll move beyond Cocoa and Cocoa Touch to incorporate data from Amazon Web Services (AWS), JavaScript Object Notation (JSON), and other formats, and standards. Together with the concepts of reactive programming and RxSwift, these tools help you build more powerful and useful apps that have wide appeal and use. What You'll Learn Work with tools such as Darwin microkernel, RxSwift, and RxCocoa Use Git repositories and other resources to get into coding Create apps that adapt to gestures and UI interaction as well as what's happening in and around the environment of the app itself. Who This Book Is For This book is for Swift programmers interested in learning to create reactive apps with RxSwift.

Swift Functional ProgrammingPackt Publishing Ltd

You want increased customer satisfaction, faster development cycles, and less wasted work. Domain-driven design (DDD) combined with functional programming is the innovative combo that will get you there. In this pragmatic, down-to-earth guide, you'll see how applying the core principles of functional programming can result in software designs that model real-world requirements both elegantly and concisely - often more so than an object-oriented approach. Practical examples in the open-source F# functional language, and examples from familiar business domains, show you how to apply these techniques to build software that is business-focused, flexible, and high quality. Domain-driven design is a well-established approach to designing software that ensures that domain experts and developers work together effectively to create high-quality software. This book is the first to combine DDD with techniques from statically typed functional programming. This book is perfect for newcomers to DDD or functional programming - all the techniques you need will be introduced and explained. Model a complex domain accurately using the F# type system, creating compilable code that is also readable documentation---ensuring that the code and design never get out of sync. Encode business rules in the design so that you have "compile-time unit tests," and eliminate many potential bugs by making illegal states unrepresentable. Assemble a series of small, testable functions into a complete use case, and compose these individual scenarios into a large-scale design. Discover why the combination of functional programming and DDD leads naturally to service-oriented and hexagonal architectures. Finally, create a functional domain model that works with traditional databases, NoSQL, and event stores, and safely expose your domain via a website or API. Solve real problems by focusing on real-world requirements for your software. What You Need: The code in this book is designed to be run interactively on Windows, Mac and Linux. You will need a recent version of F# (4.0 or greater), and the appropriate .NET runtime for your platform. Full installation instructions for all platforms at fsharp.org.

Deep Dive Into Swift! Swift is a rich language with a plethora of features to offer. Reading the official documentation or entry-level books is important, but it's not enough to grasp the true power of the language. Expert Swift is here to help, by showing you how to

harness the full power of Swift. You'll learn about advanced usages of protocols, generics, functional reactive programming, API design and more. Who This Book is For This book is for intermediate Swift developers who already know the basics of Swift and are looking to deepen their knowledge and understanding of the language. Topics Covered in Expert Swift

Protocols and Generics: Learn how protocols and generics work, and how you can leverage them in your code to produce clean, long-lasting and easy-to-refactor APIs.

Sequences and Collections: Learn how to use Sequences and Collections to write generic algorithms that operate across type families.

Unsafe: Understand the memory layout of types and how to use typed and untyped pointers.

Functional Reactive Programming: Explore the most important and refined concepts of functional reactive programming and how you can apply these concepts to your apps.

Objective-C Interoperability: Learn how to expose Objective-C code to Swift and vice versa.

Library and API Design: Enhancing your skill set and intuition for designing great APIs.

One thing you can count on: after reading this book, you'll be prepared to use the advanced features of Swift and improve your existing code with the knowledge you'll acquire.

Presents a guide to the features of C?, covering such topics as functions, generics, iterators, currying, caching, order functions, sequences, monads, and MapReduce.

Explore functional programming and discover new ways of thinking about code. You know you need to master functional programming, but learning one functional language is only the start. In this book, through articles drawn from PragPub magazine and articles written specifically for this book, you'll explore functional thinking and functional style and idioms across languages. Led by expert guides, you'll discover the distinct strengths and approaches of Clojure, Elixir, Haskell, Scala, and Swift and learn which best suits your needs. Contributing authors: Rich Hickey, Stuart Halloway, Aaron Bedra, Michael Bevilacqua-Linn, Venkat Subramaniam, Paul Callaghan, Jose Valim, Dave Thomas, Natasha Murashev, Tony Hillerson, Josh Chisholm, and Bruce Tate. Functional programming is on the rise because it lets you write simpler, cleaner code, and its emphasis on immutability makes it ideal for maximizing the benefits of multiple cores and distributed solutions. So far nobody's invented the perfect functional language - each has its unique strengths. In *Functional Programming: A PragPub Anthology*, you'll investigate the philosophies, tools, and idioms of five different functional programming languages. See how Swift, the development language for iOS, encourages you to build highly scalable apps using functional techniques like map and reduce. Discover how Scala allows you to transition gently but deeply into functional programming without losing the benefits of the JVM, while with Lisp-based Clojure, you can plunge fully into the functional style. Learn about advanced functional concepts in Haskell, a pure functional language making powerful use of the type system with type inference and type classes. And see how functional programming is becoming more elegant and friendly with Elixir, a new functional language built on the powerful Erlang base. The industry has been embracing functional programming more and more, driven by the need for concurrency and parallelism. This collection of articles will lead you to mastering the functional approach to problem solving. So put on your explorer's hat and prepare to be surprised. The goal of exploration is always discovery. What You Need: Familiarity with one or more programming languages.

Bring the power of functional programming to Swift to develop clean, smart, scalable and reliable applications. About This Book Written for the latest version of Swift, this is a comprehensive guide that introduces iOS, Web and macOS developers to the all-new world of functional programming that has so far been alien to them Get familiar with using functional programming alongside existing OOP techniques so you can get the best of both worlds and develop clean, robust, and scalable code Develop a case study on example backend API with Swift and Vapor Framework and an iOS application with Functional Programming, Protocol-Oriented Programming, Functional Reactive Programming, and Object-Oriented Programming techniques Who This Book Is For Meant for a reader who knows object-oriented programming, has some experience with Objective-C/Swift programming languages and wants to further enhance his skills with functional programming techniques with Swift 3.x. What You Will Learn Understand what functional programming is and why it matters Understand custom operators, function composition, currying, recursion, and memoization Explore algebraic data types, pattern matching, generics, associated type protocols, and type erasure Get acquainted with higher-kinded types and higher-order functions using practical examples Get familiar with functional and non-functional ways to deal with optionals Make use of functional data structures such as semigroup, monoid, binary search tree, linked list, stack, and lazy list Understand the importance of immutability, copy constructors, and lenses Develop a backend API with Vapor Create an iOS app by combining FP, OOP, FRP, and POP paradigms In Detail Swift is a multi-paradigm programming language enabling you to tackle different problems in various ways. Understanding each paradigm and knowing when and how to utilize and combine them can lead to a better code base. Functional programming (FP) is an important paradigm that empowers us with declarative development and makes applications more suitable for testing, as well as performant and elegant. This book aims to simplify the FP paradigms, making them easily understandable and usable, by showing you how to solve many of your day-to-day development problems using Swift FP. It starts with the basics of FP, and you will go through all the core concepts of Swift and the building blocks of FP. You will also go through important aspects, such as function composition and currying, custom operator definition, monads, functors, applicative functors, memoization, lenses, algebraic data types, type erasure, functional data structures, functional reactive programming (FRP), and protocol-oriented programming (POP). You will then learn to combine those techniques to develop a fully functional iOS application from scratch Style and approach An easy-to-follow guide that is full of hands-on coding examples of real-world applications. Each topic is explained sequentially and placed in context, and for the more inquisitive, there are more details of the concepts used. It introduces the Swift language basics and functional programming techniques in simple, non-mathematical vocabulary with examples in Swift.

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

references, and functional interfaces. You'll design and write applications better using the new standards in Java 8 and the JDK. Lambda expressions are lightweight, highly concise anonymous methods backed by functional interfaces in Java 8. You can use them to leap forward into a whole new world of programming in Java. With functional programming capabilities, which have been around for decades in other languages, you can now write elegant, concise, less error-prone code using standard Java. This book will guide you through the paradigm change, offer the essential details about the new features, and show you how to transition from your old way of coding to an improved style. In this book you'll see popular design patterns, such as decorator, builder, and strategy, come to life to solve common design problems, but with little ceremony and effort. With these new capabilities in hand, Functional Programming in Java will help you pick up techniques to implement designs that were beyond easy reach in earlier versions of Java. You'll see how you can reap the benefits of tail call optimization, memoization, and effortless parallelization techniques. Java 8 will change the way you write applications. If you're eager to take advantage of the new features in the language, this is the book for you. What you need: Java 8 with support for lambda expressions and the JDK is required to make use of the concepts and the examples in this book.

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Through the authors' carefully constructed explanations and examples, you will develop an understanding of Swift grammar and the elements of effective Swift style. This book is written for Swift 3.0 and will also show you how to navigate Xcode 8 and get the most out of Apple's documentation. Throughout the book, the authors share their insights into Swift to ensure that you understand the hows and whys of Swift and can put that understanding to use in different contexts. After working through the book, you will have the knowledge and confidence to develop your own solutions to a wide range of programming challenges using Swift.

For weeks, months—nay!—from the very moment you were born, you've felt it calling to you. At long last you'll be united with the programming language you've been longing for: Clojure! As a Lisp-style functional programming language, Clojure lets you write robust and elegant code, and because it runs on the Java Virtual Machine, you can take advantage of the vast Java ecosystem. Clojure for the Brave and True offers a "dessert-first" approach: you'll start playing with real programs immediately, as you steadily acclimate to the abstract but powerful features of Lisp and functional programming. Inside you'll find an offbeat, practical guide to Clojure, filled with quirky sample programs that catch cheese thieves and track glittery vampires. Learn how to: –Wield Clojure's core functions –Use Emacs for Clojure development –Write macros to modify Clojure itself –Use Clojure's tools to simplify concurrency and parallel programming Clojure for the Brave and True assumes no prior experience with Clojure, the Java Virtual Machine, or functional programming. Are you ready, brave reader, to meet your true destiny? Grab your best pair of parentheses—you're about to embark on an epic journey into the world of Clojure!

Summary Now updated for Swift 5! Swift is more than just a fun language to build iOS applications with. It features a host of powerful tools that, if effectively used, can help you create even better apps with clean, crystal-clear code and awesome features.

Swift in Depth is designed to help you unlock these tools and quirks and get developing next-gen apps, web services, and more! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology It's fun to create your first toy iOS or Mac app in Swift. Writing secure, reliable, professional-grade software is a different animal altogether. The Swift language includes an amazing set of high-powered features, and it supports a wide range of programming styles and techniques. You just have to roll up your sleeves and learn Swift in depth. About the Book Swift in Depth guides you concept by concept through the skills you need to build professional software for Apple platforms, such as iOS and Mac; also on the server with Linux. By following the numerous concrete examples, enlightening explanations, and engaging exercises, you'll finally grok powerful techniques like generics, efficient error handling, protocol-oriented programming, and advanced Swift patterns. Author Tjeerd in 't Veen reveals the high-value, difficult-to-discover Swift techniques he's learned through his own hard-won experience. What's inside Covers Swift 5 Writing reusable code with generics Iterators, sequences, and collections Protocol-oriented programming Understanding map, flatMap, and compactMap Asynchronous error handling with ResultBest practices in Swift About the Reader Written for advanced-beginner and intermediate-level Swift programmers. About the Author Tjeerd in 't Veen is a senior software engineer and architect in the mobile division of a large international banking firm. Table of Contents Introducing Swift in depth Modeling data with enums Writing cleaner properties Making optionals second nature Demystifying initializers Effortless error handling Generics Putting the pro in protocol-oriented programming Iterators, sequences, and collections Understanding map, flatMap, and compactMap Asynchronous error handling with Result Protocol extensions Swift patterns Delivering quality Swift code Where to Swift from here

Learn Reactive Programming in Swift with RxSwift!The popularity of reactive programming continues to grow on an every-increasing number of platforms and languages. Rx lets developers easily and quickly build apps with code that can be understood by other Rx developers - even over different platforms. Not only will you learn how to use the RxSwift port to create complex reactive applications on iOS, you'll also see how to easily solve common application design issues by using RxSwift. Finally you'll discover how to exercise full control over the library and leverage the full power of reactive programming in your apps. This book is for iOS developers who already feel comfortable with iOS and Swift, and want to dive deep into development with RxSwift. Topics Covered in RxSwift:- Getting Started: Get an introduction to the reactive programming paradigm, learn the terminology involved and see how to begin using RxSwift in your projects.- Event Management: Learn how to handle asynchronous event sequences via two key concepts in Rx - Observables and Observers.- Being Selective: See how to work with various events using concepts such as filtering, transforming, combining, and time operators.- UI Development: RxSwift makes it easy to work with UI of your apps using RxCocoa, which provides integration of both UIKit and Cocoa.- Intermediate Topics: Level up your RxSwift knowledge with chapters on reactive networking, multi-threading, and error handling. And much, much more! By the end of this book, you'll have hands-on experience solving common issues in a reactive paradigm - and you'll be well on your way to coming up with your own Rx patterns and solutions!

Agda is an advanced programming language based on Type Theory. Agda's type system is expressive enough to support full functional verification of programs, in two styles. In external verification, we write pure functional programs and then write proofs of properties about

them. The proofs are separate external artifacts, typically using structural induction. In internal verification, we specify properties of programs through rich types for the programs themselves. This often necessitates including proofs inside code, to show the type checker that the specified properties hold. The power to prove properties of programs in these two styles is a profound addition to the practice of programming, giving programmers the power to guarantee the absence of bugs, and thus improve the quality of software more than previously possible. Verified Functional Programming in Agda is the first book to provide a systematic exposition of external and internal verification in Agda, suitable for undergraduate students of Computer Science. No familiarity with functional programming or computer-checked proofs is presupposed. The book begins with an introduction to functional programming through familiar examples like booleans, natural numbers, and lists, and techniques for external verification. Internal verification is considered through the examples of vectors, binary search trees, and Braun trees. More advanced material on type-level computation, explicit reasoning about termination, and normalization by evaluation is also included. The book also includes a medium-sized case study on Huffman encoding and decoding.

Take your macOS Sierra to the next level using the latest tools, designs, and best coding practices while developing with Swift 3.0 About This Book Learn to harness the power of macOS with the elegance of the Swift programming language Become highly competent in building apps on the macOS platform Get the most in-depth guide with a hands-on approach on the latest version of macOS Who This Book Is For This book is for developers who have some experience with macOS and want to take their skills to next level by unlocking the full potential of latest version of macOS with Swift 3 to build impressive applications. Basic knowledge of Swift will be beneficial but is not required. What You Will Learn Combine beautiful design with robust code for the very best user experience Bring the best coding practices to the new macOS Sierra See what's new in Swift 3.0 and how best to leverage the Swift language Master Apple's tools, including Xcode, Interface Builder, and Instruments Use Unix and other common command-line tools to increase productivity Explore the essential Cocoa frameworks, including networking, animation, audio, and video In Detail macOS continues to lead the way in desktop operating systems, with its tight integration across the Apple ecosystem of platforms and devices. With this book, you will get an in-depth knowledge of working on macOS, enabling you to unleash the full potential of the latest version using Swift 3 to build applications. This book will help you broaden your horizons by taking your programming skills to next level. The initial chapters will show you all about the environment that surrounds a developer at the start of a project. It introduces you to the new features that Swift 3 and Xcode 8 offers and also covers the common design patterns that you need to know for planning anything more than trivial projects. You will then learn the advanced Swift programming concepts, including memory management, generics, protocol orientated and functional programming and with this knowledge you will be able to tackle the next several chapters that deal with Apple's own Cocoa frameworks. It also covers AppKit, Foundation, and Core Data in detail which is a part of the Cocoa umbrella framework. The rest of the book will cover the challenges posed by asynchronous programming, error handling, debugging, and many other areas that are an indispensable part of producing software in a professional environment. By the end of this book, you will be well acquainted with Swift, Cocoa, and AppKit, as well as a plethora of other essential tools, and you will be ready to tackle much more complex and advanced software projects. Style and approach This comprehensive guide takes a hands-on practical approach incorporating a visually-rich format rather than a text heavy format. The focus is on teaching the core concepts through a series of small projects and standalone examples so you gain expertise with various aspects of macOS application development.

Enter the Swift future of iOS and OS X programming Beginning Swift Programming is your ideal starting point for creating Mac, iPhone, and iPad apps using Apple's new Swift programming language. Written by an experienced Apple developer and trainer, this comprehensive guide

explains everything you need to know to jumpstart the creation of your app idea. Coverage includes data types, strings and characters, operators and functions, arrays and dictionaries, control flow, and looping, with expert guidance on classes, objects, class inheritance, closures, protocols, and generics. This succinct — yet complete — overview provides a detailed introduction to the core features of Swift. Apple developed Swift to address the limitations of Objective-C, and add features found in more complex languages like Python. The result is simpler, cleaner, more expressive code with automatic memory management, functional programming patterns, and more, including built-in features that make Swift apps faster, scalable, and more secure. This book explains it all, helping developers master Apple's new language. Become fluent with syntax that's easier to read and maintain Understand inferred types for cleaner, less mistake-prone code Learn the key features that make Swift more expressive than Objective-C Learn the new optional types in Swift that make your code more resilient Understand the key design patterns in iOS and Mac OS programming using protocols and delegates Learn how to use generics to create highly reusable code Learn the new access controls mechanism in Swift Get up to speed quickly to remain relevant and ahead of the curve. Programming Language Pragmatics, Fourth Edition, is the most comprehensive programming language textbook available today. It is distinguished and acclaimed for its integrated treatment of language design and implementation, with an emphasis on the fundamental tradeoffs that continue to drive software development. The book provides readers with a solid foundation in the syntax, semantics, and pragmatics of the full range of programming languages, from traditional languages like C to the latest in functional, scripting, and object-oriented programming. This fourth edition has been heavily revised throughout, with expanded coverage of type systems and functional programming, a unified treatment of polymorphism, highlights of the newest language standards, and examples featuring the ARM and x86 64-bit architectures. Updated coverage of the latest developments in programming language design, including C & C++11, Java 8, C# 5, Scala, Go, Swift, Python 3, and HTML 5 Updated treatment of functional programming, with extensive coverage of OCaml New chapters devoted to type systems and composite types Unified and updated treatment of polymorphism in all its forms New examples featuring the ARM and x86 64-bit architectures

Learn how to solve blocking user experience and build event based reactive applications with Swift. Key Features Build fast and scalable apps with RxSwift Apply reactive programming to solve complex problems and build efficient programs with reactive user interfaces Take expressiveness, scalability, and maintainability of your Swift code to the next level with this practical guide Book Description RxSwift belongs to a large family of Rx implementations in different programming languages that share almost identical syntax and semantics. Reactive approach will help you to write clean, cohesive, resilient, scalable, and maintainable code with highly configurable behavior. This book will introduce you to the world of reactive programming, primarily focusing on mobile platforms. It will tell how you can benefit from using RxSwift in your projects, existing or new. Further on, the book will demonstrate the unbelievable ease of configuring asynchronous behavior and other aspects of the app that are traditionally considered to be hard to implement and maintain. It will explain what Rx is made of, and how to switch to reactive way of thinking to get the most out of it. Also, test production code using RxTest and the red/ green approach. Finally, the book will dive into real-world recipes and show you how to build a real-world app by applying the reactive paradigm. By the end of the book, you'll be able to build a reactive swift application by leveraging all the concepts this book takes you through. What you will learn Understand the practical benefits of Rx on a mobile platform Explore the building blocks of Rx, and Rx data flows with marble diagrams Learn how to convert an existing code base into RxSwift code base Learn how to debug and test your Rx Code Work with Playgrounds to transform sequences by filtering them using map, flatmap and other operators Learn how to combine different operators to work with Events in a more

controlled manner. Discover RxCocoa and convert your simple UI elements to Reactive components Build a complete RxSwift app using MVVM as design pattern Who this book is for This book is for the developers who are familiar with Swift and iOS application development and are looking out to reduce the complexity of their apps. Prior experience of reactive programming is not necessary.

Advanced Swift takes you through Swift's features, from low-level programming to high-level abstractions. In this book, we'll write about advanced concepts in Swift programming. If you have read the Swift Programming Guide, and want to explore more, this book is for you. Swift is a great language for systems programming, but also lends itself for very high-level programming. We'll explore both high-level topics (for example, programming with generics and protocols), as well as low-level topics (for example, wrapping a C library and string internals). This book explains a range of application design patterns and their implementation techniques using a single example app, fully implemented in five design patterns. Instead of advocating for any particular pattern, we lay out the problems all architectures are trying to address: constructing the app's components, communicating between the view and the model, and handling non-model state. We show high-level solutions to these problems and break them down to the level of implementation for five different design patterns - two commonly used and three more experimental. The common architectures are Model-View-Controller and Model-View-ViewModel + Coordinator. In addition to explaining these patterns conceptually and on the implementation level, we discuss solutions to commonly encountered problems, like massive view controllers. On the experimental side we explain View-State-Driven Model-View-Controller, ModelAdapter-ViewBinder, and The Elm Architecture. By examining these experimental patterns, we extract valuable lessons that can be applied to other patterns and to existing code bases.

Ready to build mobile apps that out-perform the rest? If you're an iOS developer with app-building experience, this practical guide provides tips and best practices to help you solve many common performance issues. You'll learn how to design and optimize iOS apps that deliver a smooth experience even when the network is poor and memory is low. Today's picky users want fast and responsive apps that don't hog resources. In this book, author Gaurav Vaish demonstrates methods for writing optimal code from an engineering perspective, using reusable Objective-C code that you can use right away. Up your game and create high-performance native iOS apps that truly stand out from the crowd. Measure key performance indicators—attributes that constitute and affect app performance Write efficient apps by minimizing memory and power consumption, and explore options for using available CPU cores Optimize your app's lifecycle and UI, as well as its networking, data sharing, and security features Learn about application testing, debugging and analysis tools, and monitoring your app in the wild Collect data from real users to analyze app usage, identify bottlenecks, and provide fixes Use iOS 9 upgrades to improve your app's performance

Like it or not, JavaScript is everywhere these days—from browser to server to mobile—and now you, too, need to learn the language or dive deeper than you have. This concise book guides you into and through JavaScript, written by a veteran programmer who once found himself in the same position. Speaking JavaScript helps you approach the language with four standalone sections. First, a quick-start guide teaches you just enough of the language to help you be productive

right away. More experienced JavaScript programmers will find a complete and easy-to-read reference that covers each language feature in depth. Complete contents include: JavaScript quick start: Familiar with object-oriented programming? This part helps you learn JavaScript quickly and properly. JavaScript in depth: Learn details of ECMAScript 5, from syntax, variables, functions, and object-oriented programming to regular expressions and JSON with lots of examples. Pick a topic and jump in. Background: Understand JavaScript's history and its relationship with other programming languages. Tips, tools, and libraries: Survey existing style guides, best practices, advanced techniques, module systems, package managers, build tools, and learning resources.

The morning after his six year old daughter died suddenly and unexpectedly, her father began a blog. The entries from the first five months make up the chapters of this book. This is a raw mixture of memories and self-examination.

Advanced text on how to program in the functional way; has exercises, solutions and code.

Dive into Combine! Writing asynchronous code can be challenging, with a variety of possible interfaces to represent, perform, and consume asynchronous work - delegates, notification center, KVO, closures, etc. Juggling all of these different mechanisms can be somewhat overwhelming. Does it have to be this hard? Not anymore! In this book, you'll learn about Combine - Apple's framework to work with asynchronous events in a unified and reactive way that ensures your app is always up to date based on the latest state of its data. Who This Book Is For This book is for intermediate iOS developers who already know the basics of iOS and Swift development but are interested in learning declarative/reactive programming and take their app and state management to the next level. You'll also find this book interesting if you're interested in SwiftUI - as many of the reactive capabilities keeping your SwiftUI views up-to-date are built on top of Combine. Topics Covered in Combine: Asynchronous Programming with Swift What & Why: Learn what is Combine and reactive programming and the problems they solve, and how you can unify all of your asynchronous piece of work. Operators: Learn how to compose, transform, filter and otherwise manipulate different pieces of asynchronous work using operators. In Practice: You'll gain knowledge on various topics and techniques you'll leverage when writing your own real-life apps, as well as practice these techniques with actual hands-on apps and projects. SwiftUI: You'll learn about how Combine is deeply rooted within SwiftUI and provides it with the ability to reactively update its views based on the state of your app. Advanced Combine: Once you've got a handle on the basics, you'll dive into advanced Combine topics such as Error Handling, Schedulers, and Custom Publishers. By the end of this book, you'll be a pro in building full-fledged applications using Combine's various abilities.

The free book "Fundamentals of Computer Programming with C#" is a comprehensive computer programming tutorial that teaches programming, logical thinking, data structures and algorithms, problem solving and high quality code with

lots of examples in C#. It starts with the first steps in programming and software development like variables, data types, conditional statements, loops and arrays and continues with other basic topics like methods, numeral systems, strings and string processing, exceptions, classes and objects. After the basics this fundamental programming book enters into more advanced programming topics like recursion, data structures (lists, trees, hash-tables and graphs), high-quality code, unit testing and refactoring, object-oriented principles (inheritance, abstraction, encapsulation and polymorphism) and their implementation the C# language. It also covers fundamental topics that each good developer should know like algorithm design, complexity of algorithms and problem solving. The book uses C# language and Visual Studio to illustrate the programming concepts and explains some C# / .NET specific technologies like lambda expressions, extension methods and LINQ. The book is written by a team of developers lead by Svetlin Nakov who has 20+ years practical software development experience. It teaches the major programming concepts and way of thinking needed to become a good software engineer and the C# language in the meantime. It is a great start for anyone who wants to become a skillful software engineer. The books does not teach technologies like databases, mobile and web development, but shows the true way to master the basics of programming regardless of the languages, technologies and tools. It is good for beginners and intermediate developers who want to put a solid base for a successful career in the software engineering industry. The book is accompanied by free video lessons, presentation slides and mind maps, as well as hundreds of exercises and live examples. Download the free C# programming book, videos, presentations and other resources from <http://introprogramming.info>. Title: Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) ISBN: 9789544007737 ISBN-13: 978-954-400-773-7 (9789544007737) ISBN-10: 954-400-773-3 (9544007733) Author: Svetlin Nakov & Co. Pages: 1132 Language: English Published: Sofia, 2013 Publisher: Faber Publishing, Bulgaria Web site: <http://www.introprogramming.info> License: CC-Attribution-Share-Alike Tags: free, programming, book, computer programming, programming fundamentals, ebook, book programming, C#, CSharp, C# book, tutorial, C# tutorial; programming concepts, programming fundamentals, compiler, Visual Studio, .NET, .NET Framework, data types, variables, expressions, statements, console, conditional statements, control-flow logic, loops, arrays, numeral systems, methods, strings, text processing, StringBuilder, exceptions, exception handling, stack trace, streams, files, text files, linear data structures, list, linked list, stack, queue, tree, balanced tree, graph, depth-first search, DFS, breadth-first search, BFS, dictionaries, hash tables, associative arrays, sets, algorithms, sorting algorithm, searching algorithms, recursion, combinatorial algorithms, algorithm complexity, OOP, object-oriented programming, classes, objects, constructors, fields, properties, static members, abstraction, interfaces, encapsulation, inheritance, virtual methods, polymorphism, cohesion, coupling, enumerations, generics, namespaces, UML, design patterns,

extension methods, anonymous types, lambda expressions, LINQ, code quality, high-quality code, high-quality classes, high-quality methods, code formatting, self-documenting code, code refactoring, problem solving, problem solving methodology, 9789544007737, 9544007733

This book will teach you how to use Swift to apply functional programming techniques to your iOS or OS X projects. These techniques complement object-oriented programming that most Objective-C developers will already be familiar with, providing you with a valuable new tool in your developer's toolbox. We will start by taking a look at Swift's new language features, such as higher-order functions, generics, optionals, enumerations, and pattern matching. Mastering these new features will enable you to write functional code effectively. After that, we will provide several examples of how to use functional programming patterns to solve real-world problems. These examples include a compositional and type-safe API around Core Image, a library for diagrams built on Core Graphics, and a small spreadsheet application built from scratch.

Annotation Code first is an additional means of building a model to be used with the Entity Framework and is creating a lot of excitement in the .NET development community. The reader will begin with an overview of what code first is, why it was created, how it fits into the Entity Framework and when to use it over the alternatives.

Stop trying to write Swift as if it were Objective-C, and start using powerful, modern technologies such as functional programming, protocol-oriented programming, lazy variables, enum associated values, operator overloading and more. 100% ADVANCED: You'll learn key features such as @autoclosure, rethrows, variadic functions, generics, lazy variables, operator overloading, and more.

POP READY: Dive into protocol-oriented programming with real-world examples that let you see for yourself why it's such a revolutionary approach to development. MONADS EXPLAINED: Struggling with functional programming? Pro Swift explains map(), flatMap(), reduce() and more, using practical examples you can apply immediately. Pro Swift teaches you to write faster, more efficient Swift with techniques you can apply in your own code immediately - upgrade your skills today!

The Swift standard library is developed using protocol-oriented programming techniques, generics, and first-class value semantics; therefore it is important that every Swift developer understand these powerful concepts and how to take advantage of them. This book will demonstrate how to use protocol-oriented programming techniques to build ...

SwiftUI is radically different from UIKit. So in this short book, we will help you build a mental model of how SwiftUI works. We explain the most important concepts in detail, and we follow them up with exercises to give you hands-on experience. SwiftUI is still a young framework, and as such, we don't believe it's appropriate to write a complete reference. Instead, this book focuses on transitioning your way of thinking from the object-oriented style of UIKit to the declarative style of SwiftUI. Thinking in SwiftUI is geared toward readers who are familiar with Swift and who have experience building apps in frameworks like UIKit.

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

Leverage the power of the Functional Reactive Programming paradigm with Swift to develop robust iOS applications About This Book- Build highly responsive applications with this practical introduction to Reactive programming- This book uses ReactiveCocoa, the most popular solution for Reactive Programming on iOS to install, debug, and develop a framework with Swift-Switch from the traditional programming style to the reactive paradigm to code your first reactive applications with ease Who This Book Is For Reactive Programming with Swift is for Swift developers who want to start making more powerful and efficient applications. You need a basic understanding of Swift to follow along. This book takes a first-principles approach to what Reactive Programming is and how you can start implementing it in your next iOS applications. What You Will Learn- Switch your programming concepts from imperative to Functional reactive programming- Improve your app's maintenance by developing with a different paradigm- Create unit tests and automation tests using the ReactiveCocoa framework- Create clear code that is very easy to read- Get accustomed to migrating mobile apps to the Reactive way of programming- Perform asynchronous calls and join them later In Detail Reactive programming helps you write applications that are more powerful and efficient. You can write more software, help more people, and create applications that scale. Reactive programming is a growing paradigm that we will help you set to work in Swift. Reactive Programming with Swift guides you through migrating from the traditional way of developing to the new ReactiveCocoa framework, which uses Swift as its main programming language. You will learn how to develop with this framework, debug code, create unit tests, use additional frameworks, and convert a traditional framework into a ReactiveCocoa one. Starting with a crash course on the fundamental concepts of Reactive programming, we'll set you up so you're ready to create reactive applications. We'll then move on to topics such as Graphical events, Streaming, and Core data, which will help you dive deeper with advanced programming. The concept of switching your programming concepts from imperative to functional reactive programming will also be covered. By the end of this book, you will be able to successfully create highly functional apps using Swift. Style and approach This book is a fast-paced, practical guide compiled with ample images and screenshots that explain how to create apps and demonstrate their logic.

Function literals, Monads, Lazy evaluation, Currying, and more About This Book Write concise and maintainable code with

streams and high-order functions Understand the benefits of currying your Golang functions Learn the most effective design patterns for functional programming and learn when to apply each of them Build distributed MapReduce solutions using Go Who This Book Is For This book is for Golang developers comfortable with OOP and interested in learning how to apply the functional paradigm to create robust and testable apps. Prior programming experience with Go would be helpful, but not mandatory. What You Will Learn Learn how to compose reliable applications using high-order functions Explore techniques to eliminate side-effects using FP techniques such as currying Use first-class functions to implement pure functions Understand how to implement a lambda expression in Go Compose a working application using the decorator pattern Create faster programs using lazy evaluation Use Go concurrency constructs to compose a functionality pipeline Understand category theory and what it has to do with FP In Detail Functional programming is a popular programming paradigm that is used to simplify many tasks and will help you write flexible and succinct code. It allows you to decompose your programs into smaller, highly reusable components, without applying conceptual restraints on how the software should be modularized. This book bridges the language gap for Golang developers by showing you how to create and consume functional constructs in Golang. The book is divided into four modules. The first module explains the functional style of programming; pure functional programming (FP), manipulating collections, and using high-order functions. In the second module, you will learn design patterns that you can use to build FP-style applications. In the next module, you will learn FP techniques that you can use to improve your API signatures, to increase performance, and to build better Cloud-native applications. The last module delves into the underpinnings of FP with an introduction to category theory for software developers to give you a real understanding of what pure functional programming is all about, along with applicable code examples. By the end of the book, you will be adept at building applications the functional way. Style and approach This book takes a pragmatic approach and shows you techniques to write better functional constructs in Golang. We'll also show you how use these concepts to build robust and testable apps.

Summary Manning's bestselling Java 8 book has been revised for Java 9! In *Modern Java in Action*, you'll build on your existing Java language skills with the newest features and techniques. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Modern applications take advantage of innovative designs, including microservices, reactive architectures, and streaming data. Modern Java features like lambdas, streams, and the long-awaited Java Module System make implementing these designs significantly easier. It's time to upgrade your skills and meet these challenges head on! About the Book *Modern Java in Action* connects new features of the Java language with their practical applications. Using crystal-clear examples and careful attention to detail, this book respects your time. It will help you expand your existing knowledge of core Java as you master modern additions like the Streams API and the Java Module System, explore new approaches to concurrency, and learn how functional concepts can help you write code that's easier to read and maintain. What's inside Thoroughly revised edition of Manning's bestselling *Java 8 in Action* New features in Java 8, Java 9, and beyond Streaming data and reactive programming The Java Module System About the Reader Written for developers familiar with core Java

features. About the Author Raoul-Gabriel Urma is CEO of Cambridge Spark. Mario Fusco is a senior software engineer at Red Hat. Alan Mycroft is a University of Cambridge computer science professor; he cofounded the Raspberry Pi Foundation. Table of Contents PART 1 - FUNDAMENTALS Java 8, 9, 10, and 11: what's happening? Passing code with behavior parameterization Lambda expressions PART 2 - FUNCTIONAL-STYLE DATA PROCESSING WITH STREAMS Introducing streams Working with streams Collecting data with streams Parallel data processing and performance PART 3 - EFFECTIVE PROGRAMMING WITH STREAMS AND LAMBDA Collection API enhancements Refactoring, testing, and debugging Domain-specific languages using lambdas PART 4 - EVERYDAY JAVA Using Optional as a better alternative to null New Date and Time API Default methods The Java Module System PART 5 - ENHANCED JAVA CONCURRENCY Concepts behind CompletableFuture and reactive programming CompletableFuture: composable asynchronous programming Reactive programming PART 6 - FUNCTIONAL PROGRAMMING AND FUTURE JAVA EVOLUTION Thinking functionally Functional programming techniques Blending OOP and FP: Comparing Java and Scala Conclusions and where next for Java

The goal of this book is to teach the skills necessary to build iOS 14 applications using SwiftUI, Xcode 12 and the Swift 5.3 programming language. Beginning with the basics, this book provides an outline of the steps necessary to set up an iOS development environment together with an introduction to the use of Swift Playgrounds to learn and experiment with Swift. The book also includes in-depth chapters introducing the Swift 5.3 programming language including data types, control flow, functions, object-oriented programming, property wrappers and error handling. An introduction to the key concepts of SwiftUI and project architecture is followed by a guided tour of Xcode in SwiftUI development mode. The book also covers the creation of custom SwiftUI views and explains how these views are combined to create user interface layouts including the use of stacks, frames and forms. Other topics covered include data handling using state properties in addition to observable, state and environment objects, as are key user interface design concepts such as modifiers, lists, tabbed views, context menus, user interface navigation, and outline groups. The book also includes chapters covering graphics drawing, user interface animation, view transitions and gesture handling, WidgetKit, document-based apps and SiriKit integration. Chapters are also provided explaining how to integrate SwiftUI views into existing UIKit-based projects and explains the integration of UIKit code into SwiftUI. Finally, the book explains how to package up a completed app and upload it to the App Store for publication. Along the way, the topics covered in the book are put into practice through detailed tutorials, the source code for which is also available for download. The aim of this book, therefore, is to teach you the skills necessary to build your own apps for iOS 14 using SwiftUI. Assuming you are ready to download the iOS 14 SDK and Xcode 12 and have an Apple Mac system you are ready to get started.

[Copyright: e59c56480976f79a56a2400d7224293c](https://www.pdfdrive.com/functional-programming-in-swift-ebook-by-raoul-gabriel-urma-mario-fusco-alan-mycroft.html)