Download Ebook Functional Programming In Swift Ebook Chris Eidhof

Data Structures & Algorithms in Swift (Fourth Edition)
Deep dive into the latest edition of the Swift programming language, 5th Edition
Updated for Swift 3
An In-Depth Guide for Programmers
Sams Teach Yourself Swift in 24 Hours
The Bulgarian C# Book
Master the fundamentals of programming in Swift 4
Beginning iPhone and iPad programming
Optimize Your Code for Better Apps
Thinking in SwiftUI
Functional Programming in Java
App Architecture
Familiarize Yourself with POP to Fully Unleash the Power of Swift 5 and Protocols
Domain Modeling Made Functional
Swift, Xcode, and Cocoa Basics
Change the way you approach your applications using functional programming in Go
Tackle Software Complexity with Domain-Driven Design and F#
Exploring Clojure, Elixir, Haskell, Scala, and Swift
Swift for Beginners
High Performance IOS Apps
An Advanced Exploration of the Swift Language
RxSwift (Fourth Edition)
iOS 12 Programming Fundamentals with Swift
Reactive Programming with Swift
Expert Swift (First Edition)
Pro Swift - Swift 4. 1 Edition
How to improve your JavaScript programs using functional techniques
Swift Apprentice (Seventh Edition)
Learning Functional Programming in Go
Swift, Xcode, and Cocoa Basics
Verified Functional Programming in Agda
Implementing Practical Data Structures with Swift
Functional Programming: a PragPub Anthology
Functional Programming in JavaScript
Functional Programming: A PragPub Anthology
Application Development with Swift
Beginning Swift
The Big Nerd Ranch Guide
Mastering Swift 5

KARLEE VANESSA

Data Structures & Algorithms in Swift (Fourth Edition)
Genever Benning
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 books 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'l be well on your way to coming up with your own Rx patterns and solutions!
Deep dive into the latest edition of the Swift programming language, 5th Edition Pragmatic Bookshelf
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.
Updated for Swift 3 Independently Published
Understanding the Protocol-Oriented Programming (POP) paradigm is imperative if you plan on designing and implementing software using Swift 5. In this book, you'll learn how to work with POP to approach app development more efficiently. First, we review what POP is and how it differs from the classical object-oriented programming approach. Next, we discuss the pillars of this new paradigm: protocol extensions, protocol inheritance, and protocol composition. In the last part of this book, we're going to implement a fully functional app using the protocol-oriented approach. Topics include: What's protocol-oriented programming? The pillars of POP Defining method requirements Class-bound protocols Adopting a protocol Generics and protocols Implementing an app from scratch using POP Throughout the book, you'll acquire coding skills that can be applied in real-world situations. About the Author Karoly Nyisztor is a veteran software engineer and instructor. He has worked with large companies such as Apple, Siemens, and SAP. Karoly has designed and built several enterprise frameworks, and he holds twelve patents related to inventions in the field of mobile computing. After 18 years, he left the corporate world to start his own business. Since 2016, he's fully committed to teaching. As an instructor, he aims to share his 20+ years of software development expertise. Karoly teaches Software Architecture, Object-Oriented Programming and Design, Python, Swift and iOS Programming, and other, programming-related topics. You can find his courses and books on all major platforms including Amazon, LinkedIn Learning, Pluralsight, Udemy, and iTunes.
An In-Depth Guide for Programmers O'Reilly Media
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.
Sams Teach Yourself Swift in 24 Hours eBookFrenzy
Offers twenty-four lessons teaching how to build next-generation OS X and iOS apps using Apple's new programming language, with step-by-step instructions for such common tasks as using operators, iterating code with loops, and introducing generics.
The Bulgarian C# Book Simon and Schuster
Develop the skills required to create compelling, maintainable, and robust iOS and OS X apps with Swift About This Book Write expressive, understandable, and maintainable Swift 2 code with this hands-on tutorial Unveil the complex underpinnings of Swift to turn your app ideas into reality This book is packed with real-life examples to help you implement concepts as you learn Who This Book Is For If you are looking to build iOS or OS X apps using the most modern technology, this book is ideal for you. You will find this book especially useful if you are new to programming or if you are yet to develop for iOS or OS X. No prior programming exposure is required. What You Will Learn Form a solid understanding of the Swift 2 language Get to know the practical aspects of how a computer program actually works Understand the paradigms used by Apple's frameworks so you are not intimidated by them Utilize the vast resources written in Objective-C to better inform your Swift programming Develop a basic portfolio of Swift code by learning the critical concepts Experience both object-oriented and functional programming Get to know the new coding techniques made available by Swift 2 Discover resources to ensure you never stop becoming a better developer In Detail Swift is Apple's new programming language and the future of iOS and OS X app development. It is a high-performance language that feels like a modern scripting language. On the surface, Swift is easy to jump into, but it has complex underpinnings that are critical to becoming proficient at turning an idea into reality. This book is an approachable, step-by-step introduction into programming with Swift for everyone. It begins by giving you an overview of the key features through practical examples and progresses to more advanced topics that help differentiate the proficient developers from the mediocre

ones. It covers important concepts such as Variables, Optionals, Closures, Generics, and Memory Management. Mixed in with those concepts, it also helps you learn the art of programming such as maintainability, useful design patterns, and resources to further your knowledge. This all culminates in writing a basic iOS app that will get you well on your way to turning your own app ideas into reality. Style and approach This is an approachable, step-by-step guide to programming in Swift 2. Each topic is separated into compressible sections that are full of practical examples and easy-to-understand explanations. Each section builds on the previous topics so you can develop a proficient and comprehensive understanding of app development in Swift 2.

Master the fundamentals of programming in Swift 4 Swift Functional Programming

Implement object-oriented programming paradigms with Swift 3.0 and mix them with modern functional programming techniques to build powerful real-world applications About This Book Leverage the most efficient object-oriented design patterns in your Swift applications Write robust, safer, and better code using the blueprints that generate objects Build a platform with object-oriented code using real-world elements and represent them in your apps Who This Book Is For This book is for iOS and macOS developers who want to get a detailed practical understanding of object-oriented programming with the latest version of Swift: 3.0. What You Will Learn Write high-quality and easy-to-maintain reusable object-oriented code to build applications for iOS, macOS, and Linux Work with encapsulation, abstraction, and polymorphism using Swift 3.0 Work with classes, instances, properties, and methods in Swift 3.0 Take advantage of inheritance, specialization, and the possibility to overload or override members Implement encapsulation, abstraction, and polymorphism Explore functional programming techniques mixed with object-oriented code in Swift 3.0 Understand the differences between Swift 3.0, previous Swift versions, and Objective-C code In Detail Swift has quickly become one of the most-liked languages and developers' de-facto choice when building applications that target iOS and macOS. In the new version, the Swift team wants to take its adoption to the next level by making it available for new platforms and audiences. This book introduces the object-oriented paradigm and its implementation in the Swift 3 programming language to help you understand how real-world objects can become part of fundamental reusable elements in the code. This book is developed with XCode 8.x and covers all the enhancements included in Swift 3.0. In addition, we teach you to run most of the examples with the Swift REPL available on macOS and Linux, and with a Web-based Swift sandbox developed by IBM capable of running on any web browser, including Windows and mobile devices. You will organize data in blueprints that generate instances. You'll work with examples so you understand how to encapsulate and hide data by working with properties and access control. Then, you'll get to grips with complex scenarios where you use instances that belong to more than one blueprint. You'll discover the power of contract programming and parametric polymorphism. You'll combine generic code with inheritance and multiple inheritance. Later, you'll see how to combine functional programming with object-oriented programming and find out how to refactor your existing code for easy maintenance. Style and approach This simple guide is packed with practical examples of solutions to common problems. Each chapter includes exercises and the possibility for you to test your progress by answering a quiz

Beginning iPhone and iPad programming Pragmatic Bookshelf 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.

Optimize Your Code for Better Apps Packt Publishing Swift greatly simplifies the process of developing applications for Apple devices. This book provides you with the essential skills to help you get started with developing applications using Swift. Key Features Teaches you how to correctly structure and architect software using Swift Uses real-world examples to connect the theory to a professional setting Imparts expertise in the core Swift standard library Book Description Take your first foray into programming for Apple devices with Swift.Swift is fundamentally different from Objective-C, as it is a protocol-oriented language. While you can still write normal object-oriented code in Swift, it requires a new way of thinking to take advantage of its powerful features and a solid understanding of the basics to become productive. What you will learn Explore the fundamental Swift programming concepts, language structure, and the Swift programming syntax Learn how Swift compares to other computer languages and how to transform your thinking to leverage new concepts such as optionals and protocols Master how to use key language elements, such as strings and collections Grasp how Swift supports modern application development using advanced features, such as built-in Unicode support and higher-order functions Who this book is for If you are seeking fundamental Swift programming skills, in preparation for learning to develop native applications for iOS or macOS, this book is the best for you. You don't need to have any prior Swift knowledge; however, object-oriented programming experience is desired.

Thinking in SwiftUI Pearson Technology Group
Learn Data Structures & Algorithms in Swift!Data structures and algorithms form the basis of computer programming and are the starting point for anyone looking to become a software engineer. Choosing the proper data structure and algorithm involves understanding the many details and trade-offs of using them, which can be time-consuming to learn - and confusing.This is where this book, Data Structures & Algorithms in Swift, comes to the rescue! In this book, you'll learn the nuts and bolts of how fundamental data structures and algorithms work by using easy-to-follow tutorials loaded with illustrations; you'll also learn by working in Swift playground code.Who This Book Is ForThis book is for developers who know the basics of Swift syntax and want a better theoretical understanding of what data structures and algorithms are to build more complex programs or ace a whiteboard interview.Topics Covered in Data Structures & Algorithms in Swift*Basic data structures and algorithms, including stacks, queues and linked lists. *How protocols can be used to generalize algorithms. *How to leverage the algorithms of the Swift standard library with your own data structures. *Trees, tries and graphs. *Building algorithms on top of other primitives. *A complete spectrum of sorting algorithms from simple to advanced. *How to think about algorithmic complexity. *Finding shortest paths, traversals, subgraphs and much more.After reading this book, you'll have a solid foundation on data structures and algorithms and be ready to solve more complex problems in your apps elegantly.

Functional Programming in Java Simon and Schuster Summary Hello Swift! is a how-to guide to programming iOS Apps with the Swift language, written from a kid's perspective. This approachable, well-illustrated, step-by-step guide takes you from beginning programming concepts all the way through developing complete apps. (Adults will like it too!) 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 play games and explore new things on your iPhone. How amazing would it be to create your own apps? With a little practice, you can! Apple's Swift language, along with special coding playgrounds and an easy-to-use programming environment, make it easier than ever. Take it from author Tanmay Bakshi, who started programming when he was just five years old. About the Book His book, Hello Swift! iOS app programming for kids and other beginners, teaches you how to write apps for iPhones and iOS devices step by step, starting with your first line of Swift code. Packed with dozens of apps and special exercises, the book will teach you how to program by writing games, solving puzzles, and exploring what your iPhone can do. Hello Swift! gets you started. Where you go next is up to you! What's inside Crystal-clear explanations anyone can understand Kid-friendly examples, including games and puzzles Learn by doing—you'll build dozens of small apps Exercises that encourage critical thinking About the Reader Written for kids who want to learn how to program. (Psst! Adults like it, too.) About the Author Tanmay Bakshi had his first

app on the iOS App Store at the age of nine. He's now the youngest IBM Champion, a Cloud Advisor, Watson Developer, TED Speaker, and Manning author! Table of Contents Get ready to build apps with Swift! Create your first app Your first real Swift code using variables I/O laboratory Computers make decisions, too! Let computers do repetitive work Knitting variables into arrays and dictionaries Reuse your code: Clean it with function detergent Reduce your code: Use less, do more with class detergent Reading and writing files Frameworks: Bookshelves of classes SpriteKit: Fun animation time Time to watch your WatchKit code Continuing your journey with Swift

App Architecture Packt Publishing Ltd
And ConclusionChapter 2. Functions; Function Parameters and Return Value; Void Return Type and Parameters; Function Signature; External Parameter Names; Overloading; Default Parameter Values; Variadic Parameters; Ignored Parameters; Modifiable Parameters; Function In Function; Recursion; Function As Value; Anonymous Functions; Define-and-Call; Closures; How Closures Improve Code; Function Returning Function; Closure Setting a Captured Variable; Closure Preserving Its Captured Environment; Curried Functions; Chapter 3. Variables and Simple Types; Variable Scope and Lifetime.

Familiarize Yourself with POP to Fully Unleash the Power of Swift 5 and Protocols Createspace Independent Publishing Platform 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.

Domain Modeling Made Functional Pearson Education 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

Swift, Xcode, and Cocoa Basics Razeware LLC
Despite using them every day, most software engineers know little about how programming languages are designed and implemented. For many, their only experience with that corner of computer science was a terrifying "compilers" class that they suffered through in undergrad and tried to blot from their memory as soon as they had scribbled their last NFA to DFA conversion on the final exam. That fearsome reputation belies a field that is rich with useful techniques and not so difficult as some of its practitioners might have you believe. A better understanding of how programming languages are built will make you a stronger software engineer and teach you concepts and data structures you'll use the rest of your coding days. You might even have fun. This book teaches you everything you need to know to implement a full-featured, efficient scripting language. You'll learn both high-

level concepts around parsing and semantics and gritty details like bytecode representation and garbage collection. Your brain will light up with new ideas, and your hands will get dirty and calloused. Starting from main(), you will build a language that features rich syntax, dynamic typing, garbage collection, lexical scope, first-class functions, closures, classes, and inheritance. All packed into a few thousand lines of clean, fast code that you thoroughly understand because you wrote each one yourself. Change the way you approach your applications using functional programming in Go Pearson Education

Develop highly efficient and appealing iOS applications by using the Swift language About This Book Develop a series of applications with Swift using the development kits and new/updated APIs Use the new features of iOS 8 to add new flavor to your applications A hands-on guide with detailed code snippets to aid you in developing powerful Swift applications Who This Book Is For If you are an iOS developer with experience in Objective-C, and wish to develop applications with Swift, then this book is ideal for you. Familiarity with the fundamentals of Swift is an added advantage but not a necessity. What You Will Learn Use playgrounds in Xcode to make the writing of Swift code productive and easy Get acquainted with the advanced features of Swift and make complete use of them in your code Add a new method for authentication to your app using Touch ID Develop health-related apps using HealthKit Take your apps to the next level of performance and capability using Metal Develop applications for wearables using WatchKit Use Notification Center to easily access all your notifications Make your users devices more stylish by using Apple's built-in Quick Type keyboard, instead of the native one In Detail After years of using Objective-C for developing apps for iOS/Mac OS, Apple now offers a new, creative, easy, and innovative programming language for application development, called Swift. Swift makes iOS application development a breeze by offering speed, security and power to your application development process. Swift is easy to learn and has awesome features such as being open source, debugging,interactive playgrounds, error handling model, and so on. Swift has simplified its memory management with Automatic Reference Counting (ARC) and it is compatible with Objective-C. This book has been created to provide you with the information and skills you need to use the new programming language Swift. The book starts with an introduction to Swift and code structure. Following this, you will use playgrounds to become familiar with the language in no time. Then the book takes you through the advanced features offered by Swift and how to use them with your old Objective-C code or projects. You will then learn to use Swift in real projects by covering APIs such as HealthKit, Metal, WatchKit, and Touch ID in each chapter. The book's easy to follow structure ensures you get the best start to developing applications with Swift. Style and approach The book achieves its end goal by dividing its content into two parts. Part 1 will take the readers, who are new to Swift, through its architecture and basics. Part 2 of the book will cover content on application

development with Swift.
"O'Reilly Media, Inc."
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.
Tackle Software Complexity with Domain-Driven Design and F# Morgan & Claypool
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 Exploring Clojure, Elixir, Haskell, Scala, and Swift Packt Publishing Ltd

Summary Functional Programming in JavaScript teaches JavaScript developers functional techniques that will improve extensibility, modularity, reusability, testability, and performance. Through concrete examples and jargon-free explanations, this book teaches you how to apply functional programming to real-life development tasks Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology In complex web applications, the low-level details of your JavaScript code can obscure the workings of the system as a whole. As a coding style, functional programming (FP) promotes loosely coupled relationships among the components of your application, making the big picture easier to design, communicate, and maintain. About the Book Functional Programming in JavaScript teaches you techniques to improve your web applications - their extensibility, modularity, reusability, and testability, as well as their performance. This easy-to-read book uses concrete examples and clear explanations to show you how to use functional programming in real life. If you're new to functional programming, you'll appreciate this guide's many insightful comparisons to imperative or object-oriented programming that help you understand functional design. By the end, you'll think about application design in a fresh new way, and you may even grow to appreciate monads! What's Inside High-value FP techniques for real-world uses Using FP where it makes the most sense Separating the logic of your system from implementation details FP-style error handling, testing, and debugging All code samples use JavaScript ES6 (ES 2015) About the Reader Written for developers with a solid grasp of JavaScript fundamentals and web application design. About the Author Luis Atencio is a software engineer and architect building enterprise applications in Java, PHP, and JavaScript. Table of Contents PART 1 THINK FUNCTIONALLY Becoming functional Higher-order JavaScript PART 2 GET FUNCTIONAL Few data structures, many operations Toward modular, reusable code Design patterns against complexity PART 3 ENHANCING YOUR FUNCTIONAL SKILLS Bulletproofing your code Functional optimizations Managing asynchronous events and data

Swift for Beginners Simon and Schuster
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.