



Second Edition

# A Swift Kickstart

DANIEL H STEINBERG



Introducing the  
Swift Programming Language

Editors Cut

## Copyright

"A Swift Kickstart" Second Edition, by Daniel H Steinberg

Copyright © 2017 - 2020 Dim Sum Thinking, Inc. All rights reserved.

ISBN-13: 978-0-9830669-8-9

## Legal

Every precaution was taken in the preparation of this book. The publisher and author assume no responsibility for errors and omissions, or for damages resulting from the use of the information contained herein and in the accompanying code downloads.

The sample code is intended to be used to illustrate points made in the text. It is not intended to be used in production code.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks or service marks. Where those designations appear in this book, and Dim Sum Thinking, Inc. was aware of the trademark claim, the designations have been printed with initial capital letters or in all capitals.

This book uses terms that are registered trademarks of Apple Inc. for which the terms of use don't permit rendering them in all caps or initial caps. You can view a complete list of the trademarks and registered trademarks of Apple Inc at <http://www.apple.com/legal/trademark/appletmlist.html>.

The Editor's Cut name and logo are registered trademarks of Dim Sum Thinking, Inc.

This is version 1.0 for Swift 5.3, Xcode 12, and iOS 14 released September 2020.

# Let's Get Started

Sections:      Swift Playgrounds  
                  Xcode Playgrounds  
                  Code Conventions  
                  Links and Credits  
                  Version History  
                  Road Map

Welcome to the Swift 5.3 release of this introductory book about the Swift Programming Language.

I loved Swift from the moment Apple demonstrated it at WWDC 2014 and I've been a fan of the (sometimes painful to live through) changes that have been made to the language since.

This book has been updated with every release of Swift. I've gone back over the entire book and reconsidered the content in terms of the latest release. This forces me to improve my code in ways that now feel natural.

Apple open-sourced Swift in December 2015 and involved the community in decisions of how to stabilize and expand the language. The conversations were amazing. We got to hear what

Apple had intended with their language and the folks at Apple working on the language got to hear how we had used it.

It took a while for the language to settle down. We saw quite a few changes in Swift 3 and 4 and then ABI and module stability in Swift 5. These improvements led most of us to write code that's both more concise and more expressive than what we were able to write before.

The additions in Swift 5.x have been written by a core team and community that better understands the language that they're now evolving. As a result not only has all of the code been updated and checked for compliance with the latest Swift and Xcode, but I've also tried to embrace the best practices that came out of the community discussions.

This isn't a comprehensive book covering every nook and cranny of Swift. This is a kickstart. It's designed to get you up and coding in Swift. That said, there's a lot of material in this short book. By the end, you'll know enough to go to the docs, forums, and other resources to learn the finer points.

This definitely isn't a book that you read and nod your head to. Code along with me either on an iPad or a Mac. Try things out and see what works and what doesn't work. Join in the fun and you'll be thinking in Swift in no time at all.

We'll start playing with code right away, working through examples using playgrounds. One of the changes in this edition is that you can use Swift Playgrounds on iOS or the Mac or Xcode Playgrounds

on the Mac if you prefer. I prefer using Xcode playgrounds on the Mac while reading this book in Books on an iPad in landscape using the scrolling mode.

Whichever platform you use for your coding must be updated to use the latest tools from Apple.

"Really," you ask, "Do I really need a Mac or an iPad? Isn't Swift open source? Couldn't I just use Linux or something?"

You can. But in this book, all of my code, descriptions, and screenshots are for the latest shipping version of the OS. So that means the screenshots are taken for and the code is verified for macOS 11 Big Sur, iPadOS 14, and Xcode 12. You can download and install Swift Playgrounds from the App Store for macOS or for iOS.

Please check <http://developer.apple.com> and Apple's Developer app in the app store for additional resources including videos, sample code, documentation, and forums. You'll also find information on what is required to take advantage of these resources.

This book is for experienced developers who are either new to Swift or who haven't been keeping up with all of the changes to Swift. So much of the code we wrote in the past couple of years looks different when we update it to the latest version of Swift. More than that, the way we think about our code has changed. We're beginning to understand what we mean when we say "That code is more Swiftly."

I'm sure you'll enjoy this journey as much as I.

Let's start by downloading the necessary files and establishing some code conventions for this book.

If you're anxious to start coding, skip ahead to the next chapter.

# Road Map

Here's a quick overview of the book. You can think of the book as falling into roughly two parts.

## Part One: Functions and Variables

This part describes the basic components of any Swift application: functions, variables, and collections. As Swift has matured, even these fundamental building blocks have changed quite a bit.

### Chapter 2: Functions

We introduce the basic syntax of Swift functions. We use playgrounds to explore how we create and call functions that may take zero or more parameters and return zero or more values.

### Chapter 3: Variables and Constants

Sometimes we need to store state. Constants are more than just a way to store magic numbers. In Swift, we think of constants as being immutable variables and use them to make it clear when we have a

variable whose value will never change. You'll learn to create and use variables and constants.

## Chapter 4: Collections

There are two fundamental collections in Swift: Array and Dictionary. These collections must be homogeneous. In other words, they must contain elements that all have the same type. We look at how to create arrays and dictionaries and how to modify them. You'll learn to enumerate arrays and dictionaries and see how to use optional types. At the end we take a brief look at a third type of collections: Set.

## Part Two: Types

In Part One, we mainly used the types provided to us in the Swift Standard Library and in the Cocoa and Cocoa Touch libraries. In Part Two, we learn to create our own types. Even the simplest of Swift applications requires that we create our own types and teach them to play nicely together.

## Chapter 5: Enumerations

There are a surprising range of applications for this simplest of Swift constructs. An enumeration is essentially just a collection of options in a given category. Enumeration members can have associated values for storing simple data or raw values for distinguishing the



members. We can even add computed properties and methods to enumerations.

## Chapter 6: Structs

Swift offers a powerful solution for encapsulating small bundles of data: structs. A struct can have both stored and computed properties. We look at how to create an instance of a struct and how to add methods to make structs easier to work with. Structs have a lot more power in Swift than they do in other languages. Often, we'll reach for them when instead you may be tempted to use a class.

## Chapter 7: Classes

The third type is Swift's reference type: classes. We create a base class and subclasses and walk through the complexities of initializing stored properties up and down a hierarchy. We can add and override methods and properties in subclasses and add observers that respond when a property value is set. In this chapter, we work a bit more with optionals and look at how to use optional chaining.

## Chapter 8: Protocols

Swift endeavors to change your mind about inheritance. Mostly, we want to inherit the signature of methods and not their implementations. Protocols collect method signatures and property

declarations into convenient packages that enumerations, structs, and classes can choose to conform to. Swift also gives us the ability to specify behavior in a protocol extension.

## Part Three: Thinking in Swift

### Chapter 9: Errors

Of course we all try to write code that "just works" but often things go wrong. In this final chapter, we look at using `Optionals`, asserts, and Swift `Errors` to handle issues. As with so many other parts of the language, Swift's `Error` mechanism is simple, clean, and powerful.

### Chapter 10: Flexible Functions

A recurring theme with Swift is that it makes code easier to reason about. In this chapter you see that by default a function makes immutable copies of the parameters that are passed to it. We then focus on creating non-mutating versions of functions that create a new instance of the type rather than changing its value.

### Chapter 11: Higher-Order Functions

This chapter is so much fun! First we learn about functions that can return or consume other functions. We then study four canonical examples from the Swift Standard Library: `map()`, `filter()`, `reduce()`, and `flatMap()`.

Wow. That's a lot to learn.

Let's get started with functions.