

HACKING WITH SWIFT



SWIFT

FOR COMPLETE BEGINNERS

YOUR HANDS-ON GUIDE

Get started with the
fundamentals of Swift
programming

FREE SAMPLE

Paul Hudson

Chapter 1

Introduction

Why Swift?

There are lots of programming languages out there, but I think you're going to really enjoy learning Swift. This is partly for practical reasons – you can make a lot of money on the App Store! – but there are lots of technical reasons too.

You see, Swift is a relatively young language, having launched only in 2014. This means it doesn't have a lot of the language cruft that old languages can suffer from, and usually means there is only one way to solve a particular problem.

At the same time, being such a new programming language means that Swift leverages all sorts of new ideas built upon the successes – and sometimes mistakes – of older languages. For example, it makes it hard to accidentally write unsafe code, it makes it very *easy* to write code that is clear and understandable, and it supports all the world languages so you'll never see those strange character errors that plague other languages.

Swift itself is just the language, and isn't designed to draw anything on the screen. When it comes to building software with Swift, you'll be using SwiftUI: Apple's powerful framework that creates text, buttons, images, user interaction, and much more. As the name suggests, SwiftUI was built for Swift – it's literally designed to leverage the power and safety offered by the language, which makes it remarkably quick to build really powerful apps.

So, you should learn Swift because you can make a lot of money with it, but also because it does so many things really well. No cruft, no confusion, just lots of power at your fingertips. What's not to like?

About this course

I've been teaching folks to write Swift since 2014, the same year Swift launched, and at this point Hacking with Swift is the world's largest site dedicated to teaching Swift.

Along the way I learned a huge amount about what topics matter the most, how to structure topics into a smooth and consistent flow, and most importantly how to help learners remember topics they've learned. This course is the product of all that learning.

Unlike my previous work this does not strive to teach you every aspect of Swift, but instead it spends more time on the subset of features that matter the most – the ones you'll use in every app you build, time and time again. Yes, there are some advanced language features covered, but I've cherry-picked them based on usefulness. When you've finished the book you might want to carry on learning some of the more advanced features, but I suspect you'd much rather get busy learning how to use SwiftUI.

Each chapter of this book is available as both text and video, but they cover exactly the same material so you're welcome to learn whichever way suits you best. If you're using the videos you'll notice that I sometimes introduce topics using slides and sometimes demonstrate them in Xcode. It might feel repetitive, but it's intentional – there's a lot of things to learn, and if you saw each one only once it just wouldn't stay in your memory!

There's one last thing: you might notice how many chapters start with “How to...”, and that's intentional – this book is here to show you how to do things in a hands-on way, as opposed to delving into theory. Theory *is* important, and you'll come across a lot of it as you can continue to learn, but here the focus is relentlessly practical because I believe the best way to learn something new is to try it yourself.

Programming is an art: don't spend all your time sharpening your pencil when you should be drawing.

How to follow along

There's a lot of code shown off in this book, and I really want to encourage you to try it all yourself – type the code into your computer, run it and see the output, then experiment a little to make sure you understand it.

To run the code in this book you should have installed Xcode 13.0 or later from the Mac App Store. It's free, and includes everything you need to follow along.

We'll be using a Swift Playground for all the code in this book. You can create one by launching Xcode, then going to the File menu and choosing New > Playground. When you're asked what kind of playground to create, choose Blank from the macOS tab, then save it somewhere you can get to easily.

Playgrounds are like little sandboxes where you can try out Swift code easily, seeing the result of your work side by side with the code. You can use one playground for all the work you'll be doing, or create new a playground for each chapter – do whatever works best for you.

Tip: The first time you run code in a playground it might take a few seconds to start, but subsequent runs will be fast.