THE

A BEGINNER'S GUIDE

SWIFT

FROM BASICS TO BEST PRACTICES

CODEBOOK

WRITTEN BY

GRACE HUANG

AUTHOR OF “THE SWIFTUI WAY”

The Swift Codebook

A Beginner’s Guide from Basics to Best Practices

Grace Huang

Table of Contents

Copyright © 2024 by Grace Huang

For inquiries regarding the rights to this book, please contact Grace Huang at higracehuang@gmail.com.

First Edition: 2024-05-15

v1.1.3

Introduction

Why this book?

Over the years I have been teaching about building apps with SwiftUI through books and courses. The most common questions I've received from students are not about SwiftUI, but rather about the specifics of the programming language Swift itself:

- "What is the difference between `let` and `var`?"
- "What does the word `in` mean when declaring a function?"
- "Can you use an `async` function instead of a completion handler?"
- "What is the purpose of the underscore `_` when used in an argument name?"
- "Can you explain the differences between `struct` and `class`?"

Then, I realized that people find it easier to conceptualize building, but often find the language to be a hurdle. That's why I wrote this book: to answer their questions and show them that learning the language is easier than they expected.

I wanted learning Swift to be easy and less daunting, so I used examples, stories, and analogies to make the concepts easy to understand. As Morgan Housel said in his book *Same As Ever*, "When a topic is complex, stories are like leverage."

I wanted this book to be a reference that you can hold in your hands and reach for when you are coding, so I designed it in a way that the table of contents can be a quick way to find what you want to check. This book is also published in paperback, so you can easily keep it by your side and refer to it whenever needed.

I wanted this book to be practical, so I included plenty of best practices in the form of tips and notes throughout. Even if you are just starting to learn Swift, you can get insights from seasoned developers in the industry.

I wanted this book for people from diverse backgrounds. Whether you're in college, sales, nursing, design, finance, or law, I want this book to speak a common language that everyone can understand and enjoy. To avoid jargon, I invited a true beginner to review and edit the book. The programming language will not be a hurdle but a stepping stone, enabling you to transform your ideas into real products.

Prerequisites

In addition to this book, you will also need the following to follow along:

- A Mac computer
- Install Swift Playgrounds or Xcode

Swift is primarily used for developing software within the Apple ecosystem, and the majority of Swift development occurs on Apple platforms. In this book, we will utilize Swift Playgrounds, an app developed by Apple, on a Mac computer to learn Swift.

No prior programming experience is required.

End Goal

By finishing this book, you will be proficient in the programming language Swift and can use it freely for your projects.

If you cannot finish the book in one sitting, you can treat this book as a reference. It is there whenever you need to look up a specific usage for Swift.

Structure

The book is structured into two main parts, each consisting of multiple chapters.

Part 1

In the first part, we lay the foundation by exploring the building blocks of the Swift language, much like learning the words when starting to learn a new natural language.

This part covers:

- Data types, both simple (integers, floats, strings, etc.) and complex (collections, enumerations, optionals)
- Variables and constants
- Operators
- Control flows
- Functions
- Error handling

These concepts weave together to form the logic of your code, allowing you to express your ideas and solutions through programming.

Part 2

The second part covers the advanced aspects of Swift. Like forming sentences that make sense in a new natural language, we will learn some key concepts built upon the basics of Swift we've learned in Part 1.

We will explore:

- Complex string manipulation techniques in Swift
- Object-Oriented Programming (OOP), one of the most important programming paradigms
- Asynchronous programming, essential for handling tasks that don't happen at the same time in the real world

Assumptions

Focus on the Language

This book focuses on the Swift programming language itself, but not the projects. The basics of Swift are the foundation for building anything in the Apple ecosystem.

Swift 5

This book is based on Swift 5.10, the latest version at the time of writing, which was released in March 2024.

If you use an earlier or later version, please expect differences. However, I will do my best to keep this book up to date as much as possible.

Get Book Updates

I update my book with the latest versions of Swift.

To receive the latest updates on the book, subscribe to my mailing list by sending an email to higracehuang@gmail.com, including:

- The book name
- The platform where you purchased the book

Feedback

Your feedback will always be appreciated. Please feel free to contact me via any of the following ways:

- Email: higracehuang@gmail.com
- X/Twitter: https://x.com/imgracehuang

Later, if you like the book, a review on the platform where you purchased it would be greatly appreciated.

A word before reading

In case you're wondering why you should still learn about a programming language, given that AI writes code 10 times faster than humans, I hope you can always keep this comic in mind.

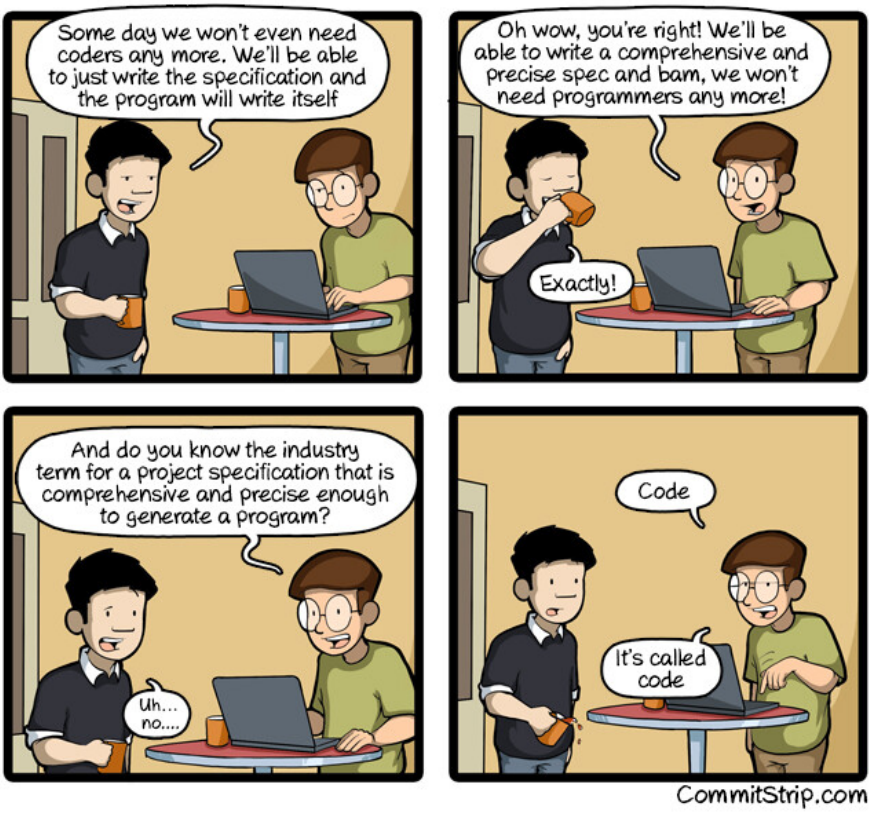
Some day we won't even need coders any more. We'll be able to just write the specification and the program will write itself
Oh wow, you're right! We'll be able to write a comprehensive and precise spec and bam, we won't need programmers any more!
Exactly!
And do you know the industry term for a project specification that is comprehensive and precise enough to generate a program?
Uh... no....
Code
It's called code
CommitStrip.com

Figure 1. Comic: A very comprehensive and precise spec, published on August 25th, 2016

AI makes coding 10x efficient but it does not make coding obsolete. Rather, it requires us to be able to recognize mistakes, with knowledge in languages.

Just like a chef, you don’t need to do everything in the kitchen, but you still need to understand the fundamentals of cooking, to recognize when a dish needs adjustments or corrections.

Have fun with Swift!

-Grace