

Have you ever had an idea for an app and wondered how to make it happen? If so, this course was designed for you. You'll start by focusing on iOS development tools, basic programming concepts, and industry best practices. Building on this foundation, you'll work through practical exercises, creating apps from scratch, and building the mindset of an app developer.

You'll build three projects—a simple flashlight app that changes the background color of the screen, a fill-in-the-blanks word guessing game, and a personality quiz. Throughout the course, you'll have the chance to design, prototype, and test an app of your own. As you build up your coding skills, you'll make connections to how you can apply what you are learning to bring your app idea to life.

You can work through Develop in Swift Fundamentals on your own, or you may be in a class with a teacher. If you're on your own, we recommend completing every lesson, lab, and guided project, to make sure you're building all the skills. If you have a teacher guiding you, keep in mind that they may use different parts of the course in different ways.

[This course was designed for students with no prior programming experience. Prior experience with languages other than Swift will definitely give you a boost when learning the basics. And if you already know something about Swift, Xcode, and iOS development, you might want to jump straight into the labs and guided projects to practice your skills.](#)

[Course Structure and Content](#)

[At the core of Develop in Swift Fundamentals are three progressively challenging guided projects, each preceded by multiple lessons that cover the concepts and skills required to build the app. App design lessons are integrated throughout the course that explore how to develop and iterate on your own app ideas as well as how to create a prototype that can serve as a compelling demo and launch your project toward a successful 1.0 release.](#)

[About the Lessons](#)

[This course features 33 lessons that help you learn a specific skill related to Swift or app development. Each lesson starts with a brief introduction to the concept, a set of learning objectives, new vocabulary terms, and references to documentation used to build the lesson. The body of the lesson includes concept explanations, sample code, and screencasts. At the end of each lesson, a lab and review questions allow you to apply the concepts you've just learned and check your understanding.](#)

Since Develop in Swift Fundamentals covers three very different types of content—Swift, app development, and app design—you'll see three different approaches to the lessons.

Swift lessons focus on specific concepts, and the labs for these are presented in playgrounds - an interactive coding environment that lets you experiment with code and see the results immediately.

App development lessons cover the Software Development Kit, or SDK. These lessons focus on building specific features for iOS apps, usually guiding you through a mini-project. The labs for these guide you to apply what you learned in a new scenario.

App design lessons use the app design cycle to build design skills to turn an idea into an app prototype in Keynote. You'll refine and improve your ideas over time, connecting what you are learning about code to bring your app idea to life.

[About the Projects](#)

Each guided project includes a description of user-centered features, a project plan, and step-by-step instructions that lead to a fully functioning app. Through these guided projects - as well as through labs sprinkled throughout the course - you will be able to customize features according to your interests. At the same time, you'll be performing the kind of work you can expect in an app development workplace.

The first project is Light, a simple flashlight app. You'll learn the basics of data, operators, and control flow in the Swift programming language. You'll also learn about Xcode, Interface Builder, building and running an app, debugging, and documentation.

The second project is Apple Pie, a word-guessing game. You'll learn about Swift strings, functions, structures, collections, and loops. You'll also learn about UIKit, the system views and controls that make up a user interface, and how to display data using Auto Layout and stack views.

The third project is Personality Quiz, a personalized survey that reveals a fun response to the user. You'll learn how to build simple workflows and navigation hierarchies using navigation controllers, tab bar controllers, and segues. You'll also learn about optionals and enumerations, two powerful tools in Swift.

After you've built the guided projects, you'll learn how to design, prototype, and architect an app of your own.

[Curriculum Pathway](#)

Develop in Swift curriculum encourages students to solve real world challenges creatively through app development. Students build foundational knowledge with Explorations or Fundamentals courses then progress to more advanced concepts in Data Collections. All courses include free teacher guides to support educators—regardless of experience teaching Swift or other programming languages.

[Explorations \(One semester\)](#)

Students learn key computing concepts, building a solid foundation in programming with Swift. They'll learn about the impact of computing and apps on society, economies, and cultures while exploring iOS app development.

Fundamentals (One semester)

Students build fundamental iOS app development skills with Swift. They'll master the core concepts and practices that Swift programmers use daily and build a basic fluency in Xcode source and UI editors. Students will be able to create iOS apps that adhere to standard practices, including the use of stock UI elements, layout techniques, and common navigation interfaces.

Data Collections (One semester)

Students expand on the knowledge and skills they developed in Fundamentals by extending their work in iOS app development, creating more complex and capable apps. They'll work with data from a server and explore new iOS APIs that allow for much richer app experiences—including displaying large collections of data in multiple formats.

Set Up Your Learning Environment

Learning to build apps involves many tools and many resources. At any given time, you may have multiple projects and playgrounds open in Xcode—as well as this book, Xcode documentation, Safari, and some number of assets on your desktop. As you start to build apps, you'll discover it's important to keep your workspace organized.

It's up to you how to navigate between applications. Some students like to use split-screen mode so they can keep all their tools in one single view. Others prefer to run each application (including this book) in full-screen mode and switch between applications as necessary.

To enter full-screen mode, click the green circle in the top left of the window or use the keyboard shortcut, Control-Command-F. You can then navigate between the full-screen applications using Mission Control, by swiping left or right with four fingers on the trackpad, or using the keyboard shortcuts, Control-Left Arrow and Control-Right Arrow.

Gather Your Materials

To complete the lessons in this guide, you'll need the following:

- A Mac running macOS Big Sur or Monterey.
- Xcode 13, available on the Mac App Store.
- Project files for the course, which you can download [here](#).

To access these materials in Xcode, you might need to enter the administrator name and password for your Mac.

[A Note About the App Design Workbook](#)

[As you work through the App Design Workbook, you'll find embedded coding activities focused on using SwiftUI in Swift Playgrounds. SwiftUI is another framework for building great apps. While this course focuses on the UIKit framework, some app developers choose to use a bit of both frameworks, since they have different strengths and weaknesses. While the App Design Workbook SwiftUI coding exercises are optional, if you choose to complete the activities they will help you expand your coding knowledge and skills.](#)

[A Word of Advice](#)

[Develop in Swift Fundamentals is designed to make Swift and iOS development approachable. But you will get stuck. All programmers get stuck.](#)

[Learning to program is hard. And building apps is hard. You'll feel discouraged when you can't get something to work just right. You'll feel frustrated when you've been stuck for hours on the same problem. And you may want to quit when you don't understand something.](#)

[But it gets easier. It turns into a puzzle. You'll experience a rush of adrenaline when you hit the Run button and your app works, especially after you've spent hours or days trying to get one little thing to work just right. You'll smile when you write code that runs perfectly on the first try. And you'll celebrate when your first app goes live on the App Store.](#)

[We're excited to see what you come up with.](#)

[Introduction](#)

[Introduction](#)

[Develop](#)

[in Swift](#)

[Fundamentals](#)

[Introduction 2](#)

[Introduction](#)

[Introduction 3](#)

[Introduction 4](#)

[Swift](#)

[App development](#)

[App■
design](#)

[Introduction 5](#)

[Light](#)

-

[Apple Pie](#)

-

[Personality](#)

[Quiz](#)

[Introduction 6](#)

[Unit 1: Values](#)

[Episode 1: The TV Club](#)

[Unit 2: Algorithms](#)

[Episode 2: The Viewing Party](#)

[Unit 3: Organizing Data](#)

[Episode 3: Sharing Photos](#)

[Unit 4: Building Apps](#)

[Introduction 7](#)

[Unit 1: Getting Started with App Development](#)

[Unit 2: Introduction to UIKit](#)

[Unit 3: Navigation and Workflows](#)

-

[Introduction 8](#)

[Unit 1: Tables and Persistence](#)

[Unit 2: Working with the Web](#)

[Unit 3: Advanced Data Display](#)

-

[Download student materials](#)

-

[Introduction 9](#)

<https://developer.apple.com/go/?id=app-d>

[Introduction 10](#)