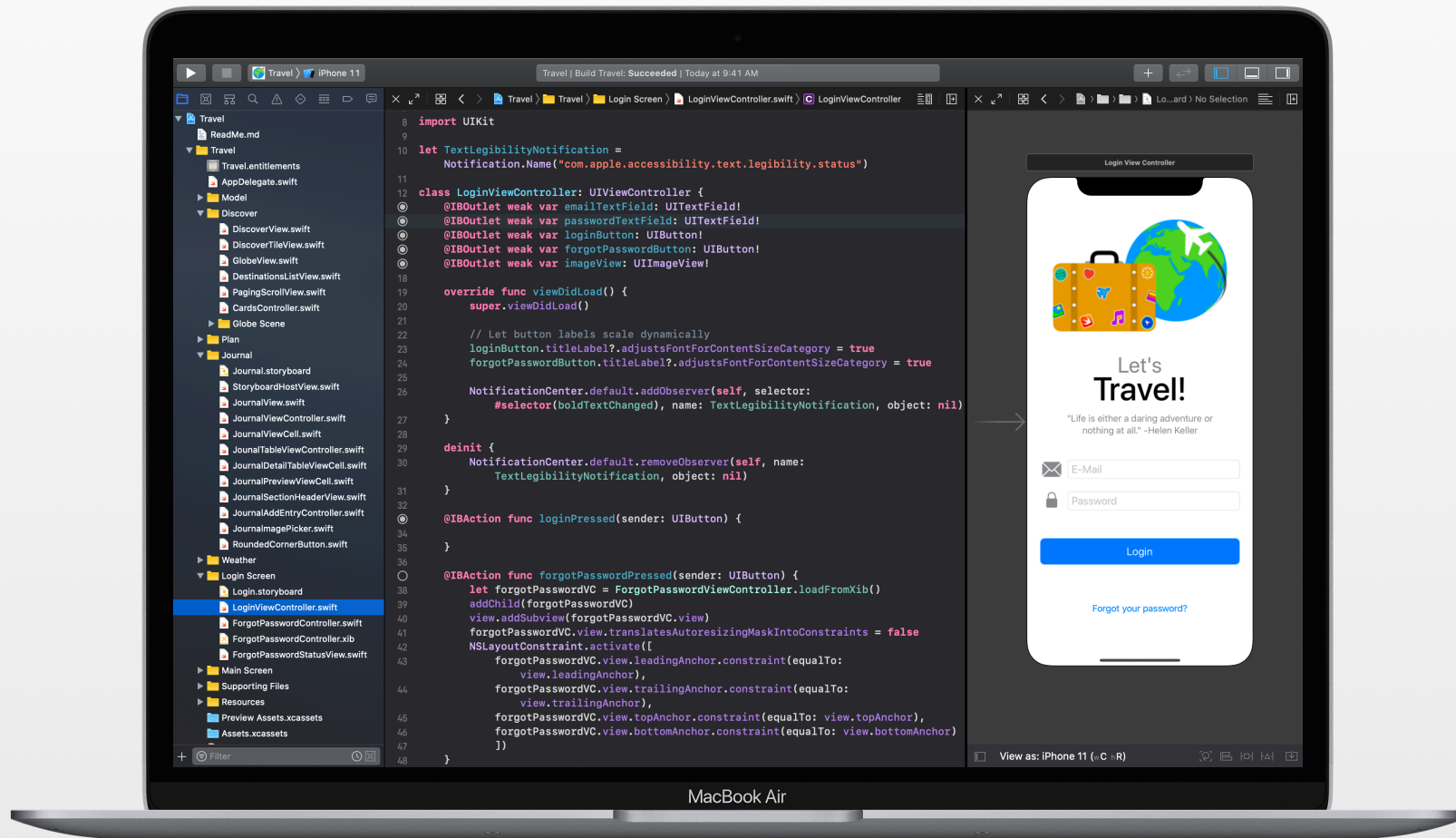


Apple Develop in Swift

Curriculum Guide



Develop in Swift

Develop in Swift is a curriculum that teaches students to use Xcode on Mac and learn Swift—a powerful and intuitive open source programming language designed by Apple. It's the same language professional developers are using in the fast-growing app economy to make apps for iOS, iPadOS, macOS, tvOS, watchOS, and beyond. The Develop in Swift curriculum is great for teaching both students who are new to coding and those with experience.



High School Curriculum Pathway

The 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—whether they’re experienced in Swift or other programming languages.

AP® CS Principles*

180 hours

Students learn key computing concepts, building a solid foundation in programming with Swift as they prepare for the AP® Computer Science Principles exam. They’ll learn about the impact of computing and apps on society, economies, and our culture while exploring iOS app development. Lessons take students through the app design process: brainstorming, planning, prototyping, and evaluating an app of their own.

Unit 1: Values

Episode 1: The TV Club

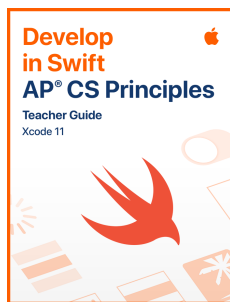
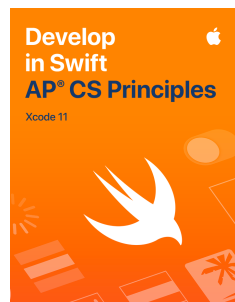
Unit 2: Algorithms

Episode 2: The Viewing Party

Unit 3: Organizing Data

Episode 3: Sharing Photos

Unit 4: Building Apps



Fundamentals

180 hours

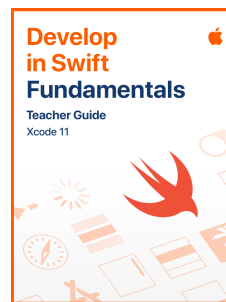
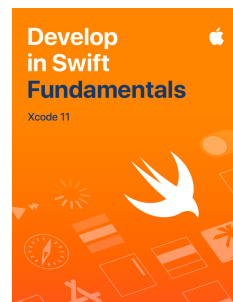
Students build fundamental iOS app development skills with Swift. They’ll master the core concepts and practices that Swift programmers use daily and build a basic fluency in Xcode source and UI editors. Students will be able to create iOS apps that adhere to standard practices, including the use of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

Unit 1: Getting Started with App Development

Unit 2: Introduction to UIKit

Unit 3: Navigation and Workflows

Unit 4: Build Your App



Data Collections

180 hours

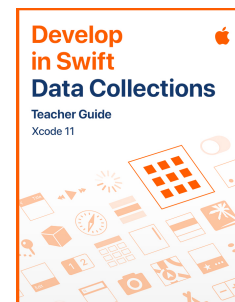
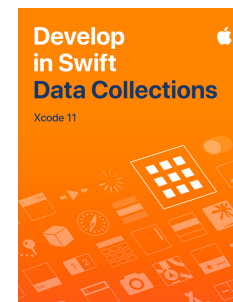
Students expand on the knowledge and skills they developed in Fundamentals by extending their work in iOS app development, creating more complex and capable apps. They’ll work with data from a server and explore new iOS APIs that allow for much richer app experiences—including displaying large collections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

Unit 1: Tables and Persistence

Unit 2: Working with the Web

Unit 3: Advanced Data Display

Unit 4: Build Your App



*For schools not teaching AP® CSP, the Develop in Swift Explorations course is available.

Coming soon

Higher Education 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—whether they have experience with 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, the economy, and cultures while exploring iOS app development. Lessons take students through the app design process: brainstorming, planning, prototyping, and evaluating an app of their own.

Unit 1: Values

Episode 1: The TV Club

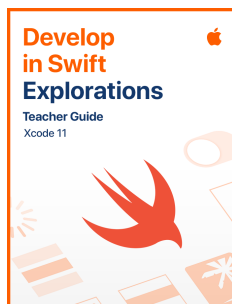
Unit 2: Algorithms

Episode 2: The Viewing Party

Unit 3: Organizing Data

Episode 3: Sharing Photos

Unit 4: Building Apps



Fundamentals

One semester

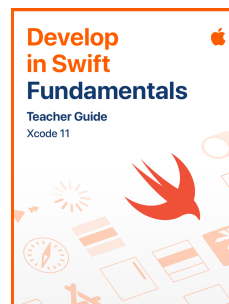
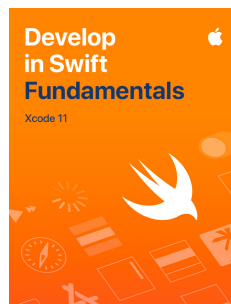
Students build fundamental iOS app development skills with Swift. They'll master the core concepts and practices that Swift programmers use daily and build a basic fluency in Xcode source and UI editors. Students will be able to create iOS apps that adhere to standard practices, including the use of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

Unit 1: Getting Started with App Development

Unit 2: Introduction to UIKit

Unit 3: Navigation and Workflows

Unit 4: Build Your App



Data Collections

One semester

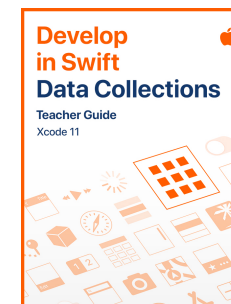
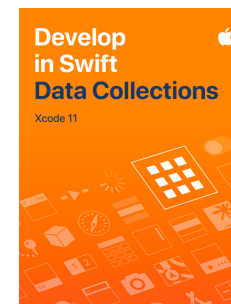
Students expand on the knowledge and skills they developed in Fundamentals by extending their work in iOS app development, creating more complex and capable apps. They'll work with data from a server and explore new iOS APIs that allow for much richer app experiences—including displaying large collections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

Unit 1: Tables and Persistence

Unit 2: Working with the Web

Unit 3: Advanced Data Display

Unit 4: Build Your App



Coming soon

Curriculum Overview

The Develop in Swift curriculum is intended for high school and higher education students to explore designing and building a fully functioning app of their own. As they develop new skills, students will master key coding concepts and can even earn AP® credit or an industry-recognized certification for their knowledge of Swift and Xcode. And students who earn certification can share a digital badge with their professional networks to convey their preparedness for industry.

Supporting teacher guides provide educators with tools to deepen engagement with aspiring app developers, regardless of experience teaching Swift or other programming languages. For after-school or summer learning programs, there are complimentary Swift Coding Club materials. And app showcases give aspiring coders a chance to celebrate their ingenuity with the community, whether they're learning in or out of the classroom.

Values | Play

Strings

What You'll Build

- A program to build a fill-in-the-blanks game.
- A lyrics-builder for a rallying song.

What You'll Learn

- How to use text in programming.
- How to use a program to change and combine words.
- What counts as text to a Swift program.

Key Vocabulary

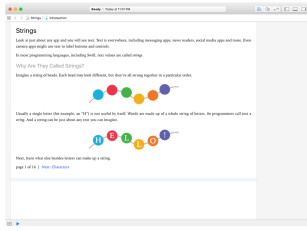
- [Character](#)
- [Escape character](#)
- [Escape sequence](#)
- [Quick Look button](#)
- [String](#)
- [String interpolation](#)
- [String concatenation](#)
- [Unicode](#)

Introduction

Numbers are wonderful, especially 0 and 1. But for the most part, people spend a lot more time interacting with text—whether it's sending a message, jotting down a reminder, or reading a post online. In this lesson, you'll learn how computers handle text, and you'll think up some creative string combinations along the way.

Go Build

Open the `Strings.playground` file in the folder you downloaded and follow the instructions.



Reflection Questions

What apps do you use that show text?

What apps do you enter text into by typing or dictating?

Do apps you use behave differently depending on what text you enter?

40

Values | Design

1.4 Design For People

Analyze Technology


As you progress through this course, you'll develop a deeper understanding of the role computers play in modern life. In fact, you'll quickly see that computing innovations touch just about every aspect of our lives. Computing innovations can be physical devices, software, or non-physical computing concepts. All computing innovations include a computer program as part of the whole.

At the same time, you'll come to appreciate the fact that every digital technology has been designed by people, for people. It's also been designed with a particular purpose in mind, such as to solve a problem, to simplify a task, or to provide for self-expression.

Analyze This

Ask your classmates how many communication apps they use in a day.

From messaging to social media posts to comments on a shared document, people are in a near-constant state of conversation. You might take this steady state for granted, but do some research into the past. Investigate how communication technology has developed over time. Create an infographic showing the evolution of communication technologies.

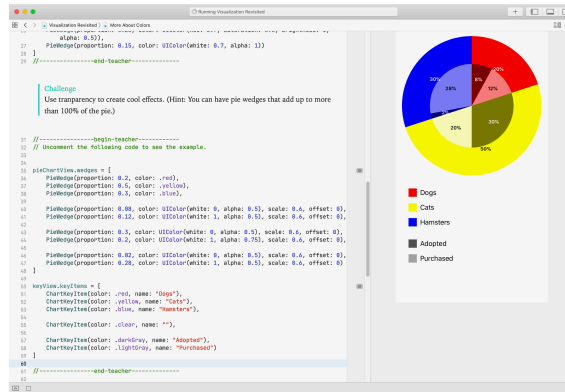


72

Key Features

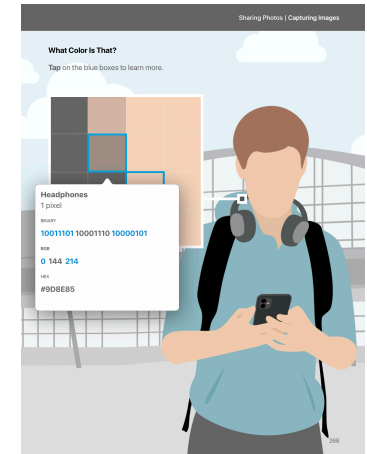
Xcode playgrounds

Students learn programming concepts as they write code in playgrounds—interactive coding environments that let them experiment with code and see results immediately.



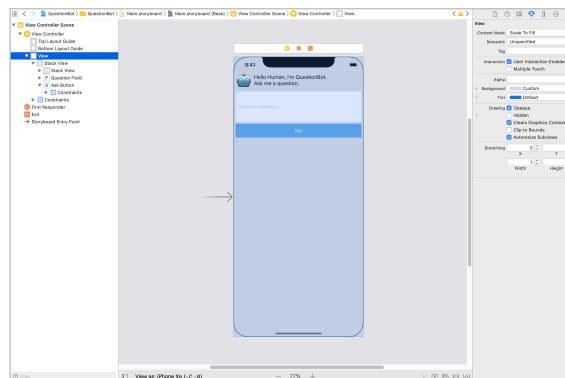
Interactive widgets*

Using photo galleries, interactive images, and Keynote files, students examine everyday activities and tools—from searching on the web and taking photos to interacting on social media—while exploring the technology behind them and their impact on society.



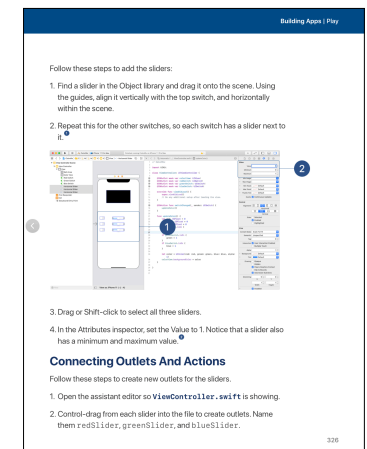
Guided app projects

Using the included project files, students can try out certain parts of code without having to build an app from scratch. Supporting images and videos challenge them to apply their knowledge.



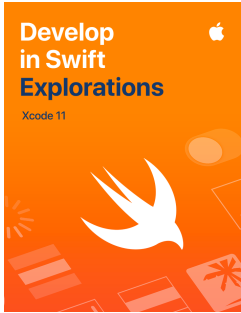
Step-by-step instructions

Detailed instructions with images and videos guide students through all the steps of building an app in Xcode.



*Available in Develop in Swift AP® CS Principles and Develop in Swift Explorations courses only.

Develop in Swift Explorations



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.

Unit 1: Values. Students learn about the fundamental units of Swift—the values that flow through their code, including text and numbers. They explore how to associate names with values using variables. The unit culminates in an app project to display a photo.

Episode 1: The TV Club. Students follow members of a TV club as they anticipate the new season of their favorite show. They learn how searching on the web and signing up for accounts relates to their personal information, as well as how to think about their privacy while using apps.

Unit 2: Algorithms. Students learn how to structure their code using functions to encapsulate repetitive tasks, use if/else statements to represent decisions, and explore how Swift uses types to distinguish different kinds of data. The culminating project is a QuestionBot app that responds to user input from the keyboard.

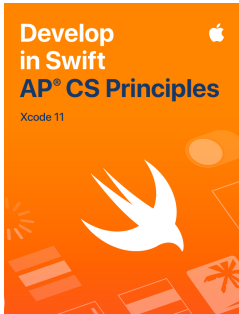
Episode 2: The Viewing Party. The TV club story continues as its members stream the episode while texting each other. Students explore how data is represented inside their devices at the lowest level and how it flows across the internet. They also learn more about security and privacy of data.

Unit 3: Organizing Data. Students explore how to create custom types using structs, group large quantities of items into arrays, and process them using loops. They also learn how enums represent a set of related values, and in the app project at the end of the unit, they build an interactive game with colorful shapes.

Episode 3: Sharing Photos. The TV club concludes as its members share pictures of the viewing party on social media. Students learn about digitizing analog data and parallel computing, and they explore some consequences of sharing data online.

Unit 4: Building Apps. Students deepen their skills in Xcode and Interface Builder in guided projects to build apps from the ground up. They learn how to add user interface elements to a screen, connect those elements to their code, and respond to the events generated by user interaction. They use the incremental development process to build their apps one piece at a time, testing as they go. The culmination of the unit is a study app with flash card and quiz modes.

Develop in Swift AP[®] CS Principles



As a College Board–endorsed provider for the 2020–2021 school year, Apple created Develop in Swift AP[®] CS Principles to prepare students for the AP[®] Computer Science Principles exam. This course is based on the Develop in Swift Explorations course and helps students learn key computing concepts and build a solid foundation in programming with Swift.

The Develop in Swift AP[®] CS Principles course aligns with the College Board curricular requirements, with pedagogy that supports the five Big Ideas to encompass foundational computer science concepts. These ideas include Creative Development, Data, Algorithms and Programming, Computing Systems and Networks, and Impact of Computing. The course also contains six Computational Thinking Practices that describe how students will explore the learning objectives, including Computational Solution Design, Algorithms and Program Development, Abstraction in Program Development, Code Analysis, Computing Innovations, and Responsible Computing.

Unit 1: Values. Students learn about the fundamental units of Swift—the values that flow through their code, including text and numbers. They explore how to associate names with values using variables. The unit culminates in an app project to display a photo.

Episode 1: The TV Club. Students follow members of a TV club as they anticipate the new season of their favorite show. They learn how searching on the web and signing up for accounts relates to their personal information, as well as how to think about their privacy while using apps.

Unit 2: Algorithms. Students learn how to structure their code using functions to encapsulate repetitive tasks, use if/else statements to represent decisions, and explore how Swift uses types to distinguish different kinds of data. The culminating project is a QuestionBot app that responds to user input from the keyboard.

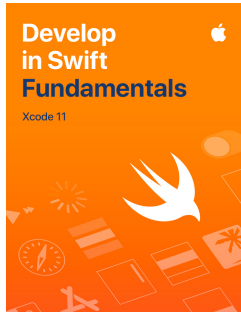
Episode 2: The Viewing Party. The TV club story continues as its members stream the episode while texting each other. Students explore how data is represented inside their devices at the lowest level and how it flows across the internet. They also learn more about security and privacy of data.

Unit 3: Organizing Data. Students explore how to create custom types using structs, group large quantities of items into arrays, and process them using loops. They also learn how enums represent a set of related values, and in the app project at the end of the unit, they build an interactive game with colorful shapes.

Episode 3: Sharing Photos. The TV club concludes as its members share pictures of the viewing party on social media. Students learn about digitizing analog data and parallel computing, and they explore some consequences of sharing data online.

Unit 4: Building Apps. Students deepen their skills in Xcode and Interface Builder in guided projects to build apps from the ground up. They learn how to add user interface elements to a screen, connect those elements to their code, and respond to the events generated by user interaction. They use the incremental development process to build their apps one piece at a time, testing as they go. The culmination of the unit is a study app with flash card and quiz modes.

Develop in Swift Fundamentals



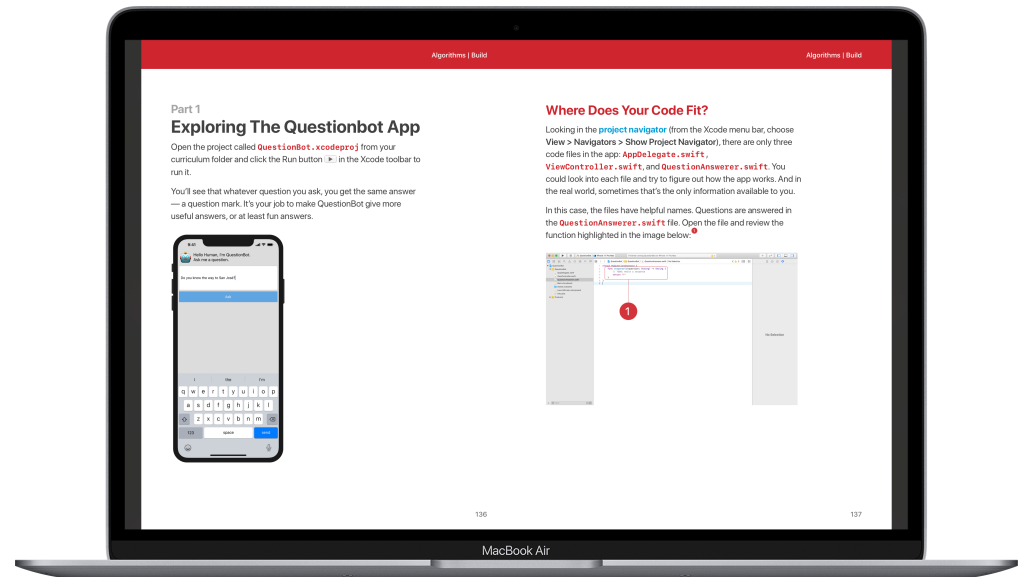
Students build fundamental iOS app development skills with Swift. They'll master the core concepts and practices that Swift programmers use daily and build a basic fluency in Xcode source and UI editors. Students will be able to create iOS apps that adhere to standard practices, including the use of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

Unit 1: Getting Started with App Development. Students find out about the basics of data, operators, and control flow in Swift, as well as documentation, debugging, Xcode, building and running an app, and Interface Builder. They then apply this knowledge to a guided project called Light in which they create a simple flashlight app.

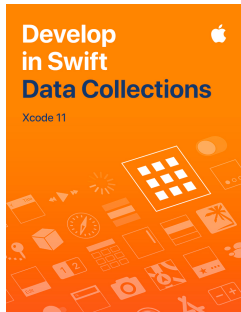
Unit 2: Introduction to UIKit. Students explore Swift strings, functions, structures, collections, and loops. They also learn about UIKit—the system views and controls that make up a user interface—and how to display data using Auto Layout and stack views. They put this knowledge to practice in a guided project called Apple Pie, where they build a word-guessing game app.

Unit 3: Navigation and Workflows. Students discover how to build simple workflows and navigation hierarchies using navigation controllers, tab bar controllers, and segues. They also examine two powerful tools in Swift: optionals and enumerations. They put this knowledge into practice with a guided project called Personality Quiz—a personalized survey that reveals a fun response to the user.

Unit 4: Build Your App. Students learn about the design cycle and use it to design an app of their own. They explore how to develop and iterate on their designs, as well as to create a prototype that can serve as a compelling demo and launch their project toward a successful 1.0 release.



Develop in Swift Data Collections



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. Students learn new features of the iOS SDK to continue their app developer journey.

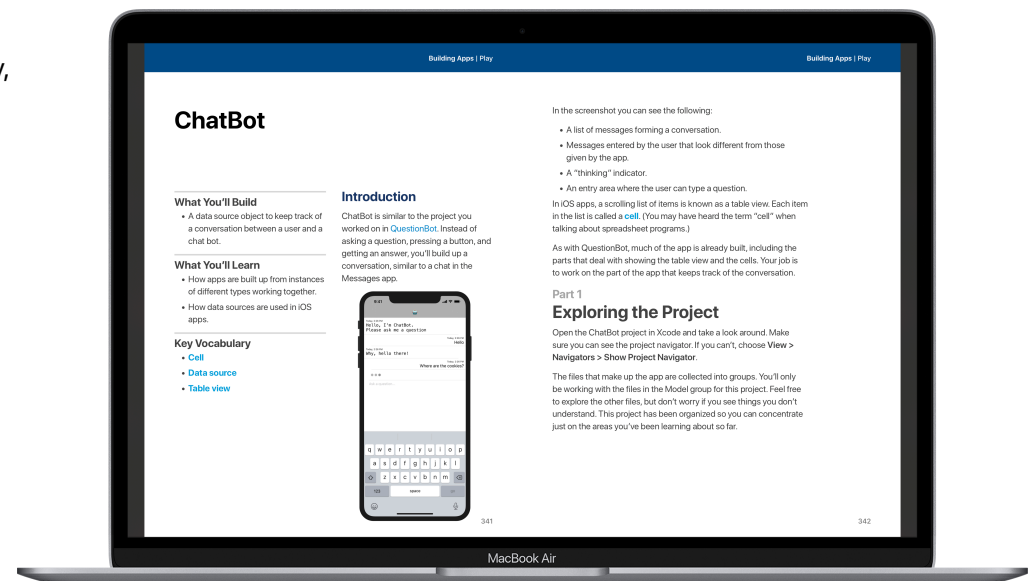
Coming soon

Unit 1: Tables and Persistence. Students find out about scroll views, table views, and building complex input screens. They also explore how to save data, share data to other apps, and work with images in a user's photo library. They use their new skills in a guided project called List, a task-tracking app that allows the user to add, edit, and delete items in a familiar table-based interface.

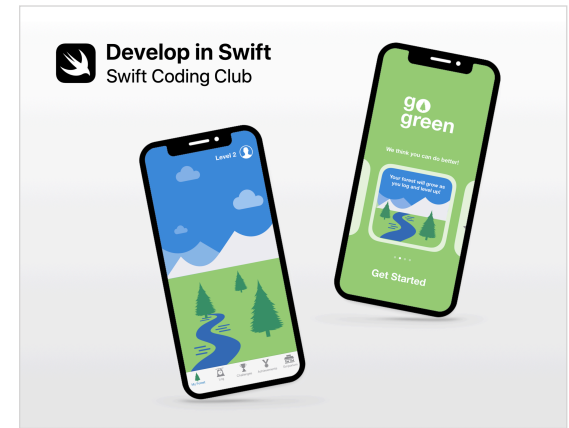
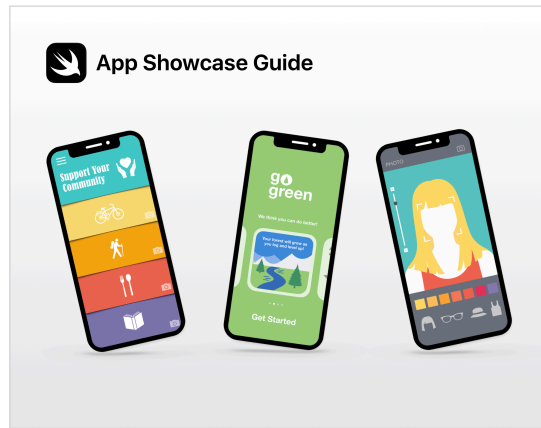
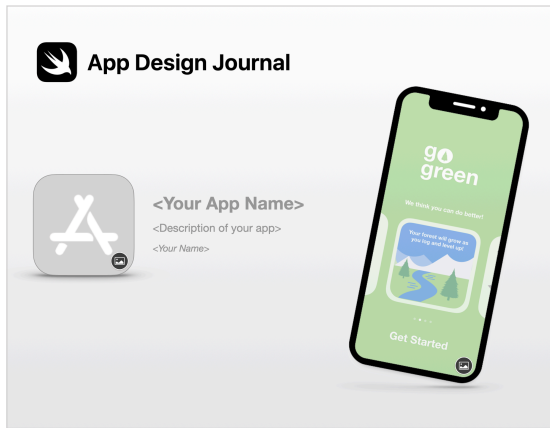
Unit 2: Working with the Web. Students learn about animations, concurrency, and working with the web. They apply what they've learned in a guided project called Restaurant—a customizable menu app that displays a restaurant's available dishes and allows the user to submit an order. The app uses a web service that lets students set up the menu with their own menu items and photos.

Unit 3: Advanced Data Display. Students learn about how to use collection views to display data in a highly customizable, two-dimensional layout. They also discover the power of Swift generics and bring all their skills together in an app that manages a complex data set and presents a customizable interface.

Unit 4: Build Your App. Students learn about the app design cycle and use it to design an app of their own. They explore how to develop and iterate on their designs, as well as to create a prototype that can serve as a compelling demo and launch their project toward a successful 1.0 release.



Supplemental Resources



App Design Journal

Even before they can begin to develop apps in Swift, students can think about the types of apps they might want to design and build. The App Design Journal guides students with design thinking project to help them prototype their ideas, test the app with their peers, and refining the user experience.

App Showcase Guide

Celebrate student ingenuity by encouraging students to share their coding achievements with community events, such as project demonstration events or app showcases. The App Showcase Guide provides practical support to help you plan and execute a showcase event.

Swift Coding Club

Swift Coding Clubs are a fun way to design apps. Activities are built on learning Swift programming concepts in Xcode playgrounds on Mac. Students collaborate with peers to prototype apps and think about how code can make a difference in the world around them.

Apple Professional Learning

The Develop in Swift curriculum is supported with a free online professional learning course hosted on Canvas to prepare educators to teach Develop in Swift Explorations, with optional content for Develop in Swift AP® CS Principles. Participants will learn the foundational knowledge needed to teach Swift and Xcode directly from Apple experts, making this an ideal introductory course for teaching Develop in Swift in any educational environment.

Help students stand out in the app economy with Swift certification

Educators who are teaching app development with Swift can help their students earn recognition for their knowledge of Swift and Xcode. The app development with Swift certification is based on Fundamentals and Data Collections courses to help students stand out and compete for high-demand jobs in iOS app development. Available through an exam administered by Certiport, app development with Swift certification shows that students are ready to take the next step in becoming app developers. Learn more: www.certiport.com/apple

Additional Resources

Apple Professional Learning

The Develop in Swift curriculum is supported with a free online professional learning course available on Canvas to prepare educators to teach Develop in Swift Explorations with optional content for Develop in Swift AP® CS Principles. Participants will learn the foundational knowledge needed to teach Swift and Xcode directly from Apple experts, making this an ideal introductory course for teaching Develop in Swift in any educational environment.

For educators interested in going further, Apple Professional Learning Specialists help provide faculty with personal support for innovating their instructional practices in ways that will engage students and enable personalized learning.

Apple Professional Learning Specialists organize multiple-day training engagements designed to provide:

- Leadership visioning and planning to help administrators prioritize learning objectives.
- Guidance from an Apple Professional Learning Specialist who's dedicated to your project.
- Customized, research-based professional learning plans to match learning goals.
- Hands-on, immersive learning experiences to help faculty develop innovative instructional practices that engage students.
- An opportunity to work with your professional learning leaders to build sustainability.

To learn more about the Apple Professional Learning course visit apple.co/developinswiftexplorationspl or email contact your Apple account executive or AppleProfessionalLearning@apple.com.

Develop in Swift

- [Develop in Swift program](#)
- [App Development with Swift Level 1 certification](#)
- [Develop in Swift Explorations Professional Learning Course](#)
- [Develop in Swift on Canvas Commons](#)

Everyone Can Code

- [Everyone Can Code program](#)
- [Everyone Can Code Puzzles](#)
- [Everyone Can Code Puzzles Teacher Guide](#)
- [Swift Playgrounds app](#)

About Swift

Swift is the powerful and intuitive programming language created by Apple for building apps. Swift is not only great for getting you started with coding, it's also super powerful. It's designed to scale from writing the simplest program, like "Hello, world!", to the world's most advanced software. Learn more about [Swift](#).

About Xcode

Xcode is the Mac app used to build every other Mac app and every iOS app, too. It has all the tools you need to create an amazing app experience. And it's available as a [free download](#) from the Mac App Store. Learn more about [Xcode](#).

