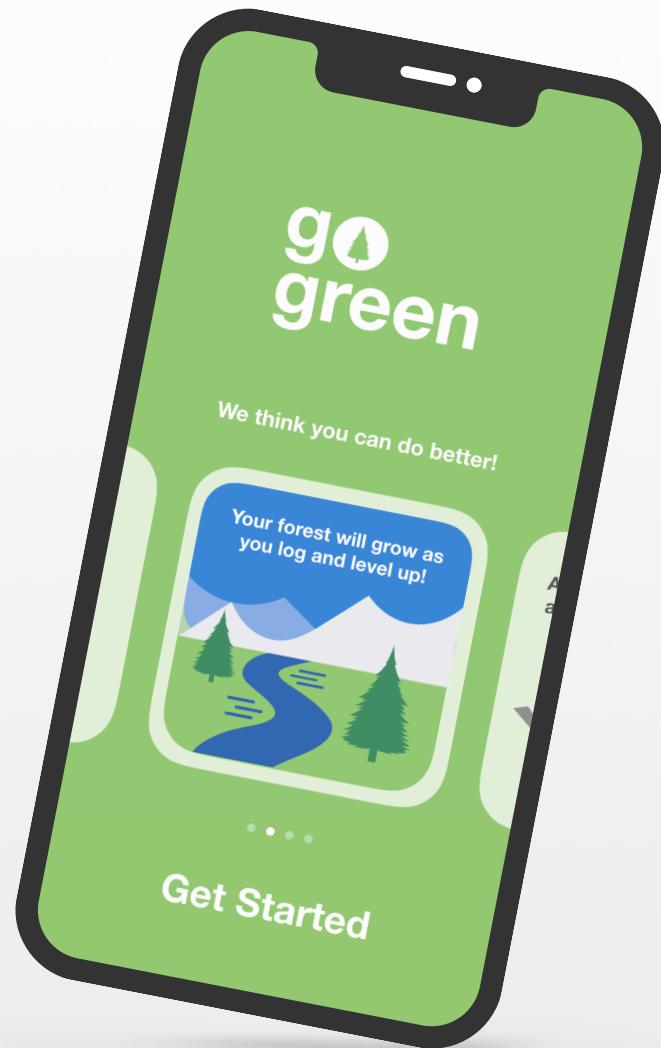
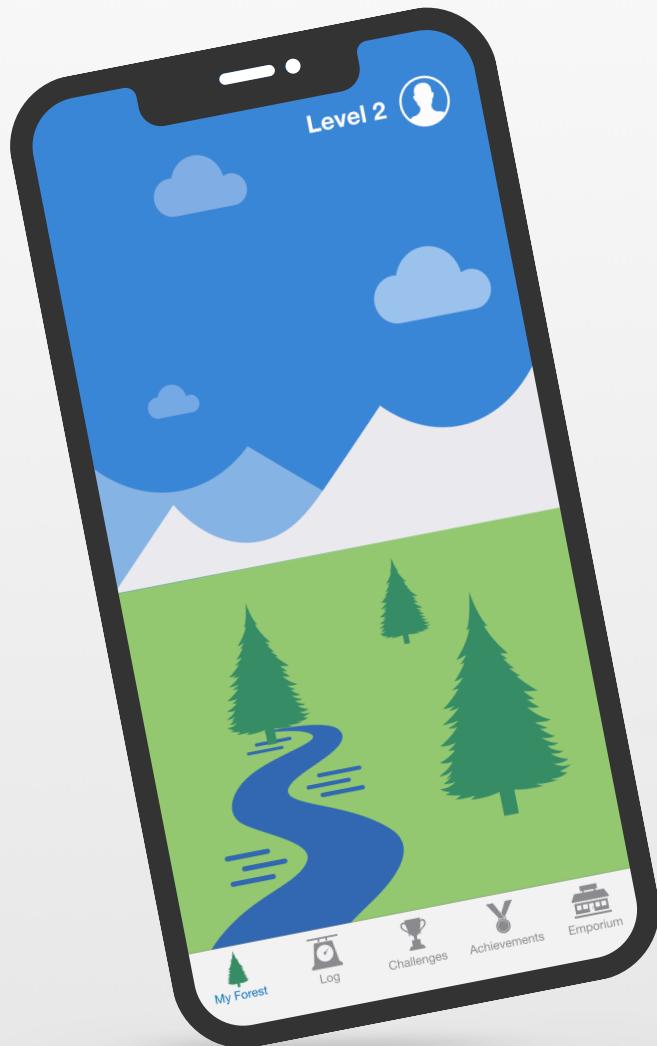




# Develop in Swift

## Swift Coding Club



# Welcome to the Swift Coding Club!

Learning to code teaches you how to solve problems and work with others in creative ways. And it helps you bring your ideas to life.

Swift Coding Clubs are an excellent way to learn to code and design apps. Activities built around Swift, Apple’s coding language, help you collaborate as you learn to code, prototype apps, and think about how code can make a difference in the world around you.

You don’t have to be an educator or a programmer to run a Swift Coding Club. The materials are self-paced, so you can even learn alongside your club members. And you can all celebrate your club’s ideas and designs with an app showcase event for your community.

---

This guide is arranged in three sections:



## Get Started

Everything you need to launch a Swift Coding Club.



## Learn and Apply

Modules and activities for club sessions.



## Celebrate

Helpful resources to plan and host a community event.

## Coding Resources

Swift Coding Clubs are built around a variety of resources for learning to code. Apple takes coders from learning the basics on iPad to building real apps on Mac.



### Everyone Can Code | Ages 10+

Use Swift code to learn coding fundamentals with Swift Playgrounds on iPad. Find out more about the [Everyone Can Code curriculum >](#)



### Develop in Swift | Ages 14+

Learn to develop apps in Xcode on Mac. Find out more about the [Develop in Swift offering >](#)

### Learn to Code with Apple

You don’t need prior experience to dive right into creating apps for Apple platforms. Apple’s app development curriculum makes it easy for anyone to code in Swift just like the pros—whether it’s for a semester in school, for professional certification, or to advance your skills. [Learn more >](#)

# Get Started

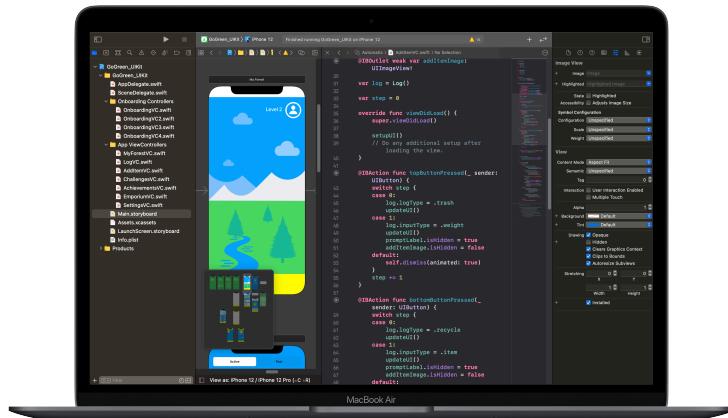


## 1. Explore Develop in Swift resources

Develop in Swift materials teach you how to use Xcode on Mac and 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, macOS, tvOS, iPadOS, watchOS, and beyond. And it's great for inspiring club members who are both new to coding and those with experience. Before you start designing your club experience, explore these Develop in Swift resources.

### Xcode

Xcode is an integrated development environment professional developers use to build real apps. It gives you the tools you need to create a complete app—from designing your user interface (UI) and implementing code to testing and debugging your app and provisioning it for distribution on the App Store.

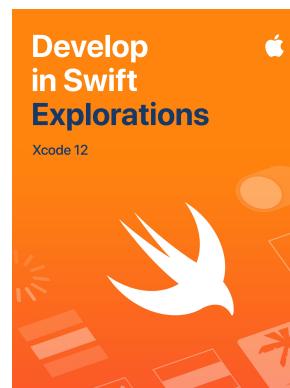


[Download Xcode >](#)

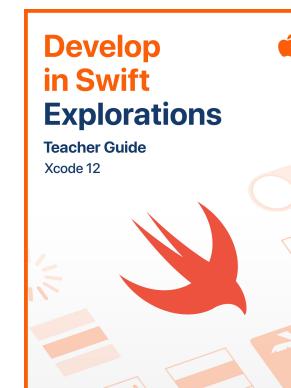
Get Started

### Develop in Swift Explorations

Learn key computing concepts for building a solid foundation in programming with Swift. Learn about the impact of computing and apps on society, economies, and cultures while exploring iOS app development. Lessons take you through the app design process: brainstorming, planning, prototyping, and evaluating an app of your own.



[Download Develop in Swift Explorations >](#)



[Download Develop in Swift Explorations Teacher Guide >](#)



## 2. Check your tech

Be sure you have the following before your first meeting:

- **Mac.** You'll need a Mac running macOS Big Sur or later. It's best if each club participant has their own Mac. You can also share and code together.
- **Xcode 12.** This free Mac app from Apple is used to build every Mac app and every iOS app, too. Xcode has all the tools needed for creating an amazing app.
- **Develop in Swift Explorations.** This free resource from Apple guides new coders through key computing concepts to build a solid foundation in programming with Swift.
- **Keynote.** You'll use this app on Mac for your designing your app prototypes.

Visit [Apple Support](#) to get help with Apple products.

## 3. Make a plan

Here are some things to consider:

- Who are your club members? What are their interests? Do they have experience with coding, or are they brand-new to it?
- How often will your club meet? How many hours of coding activities will you have?
- What technology is available for the club?
- What are the goals of your club?





## 4. Spread the word

Let people know about your Swift Coding Club. Here are some ideas and resources to attract new members to your club:

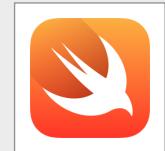
- **Announce your club.** Use email, social media, the web, flyers, or word of mouth to let your community know about your club.
  - **Host an informational meeting.** Ask potential club members about their interests and the types of projects they'd want to create. Talk about ideas for holding community events and how they can get involved.
- You can also share a short video about the club online or on social media.

The items below can help you promote and personalize your Swift Coding Club:

- **Posters.** [Download this free template](#), then personalize it to create your own poster. Print and display it, or make a digital poster to share online. Be sure to include details for when and where the club will meet and how to join.
- **Stickers and T-shirts.** Use these [Swift Coding Club stickers](#) to help promote your club. T-shirts are a great way to recognize members who participate in app showcase events. Download the [Swift Coding Club T-shirt template](#) to make shirts for your members.



Swift Coding Club poster



Swift Coding Club sticker



Swift Coding Club T-shirt

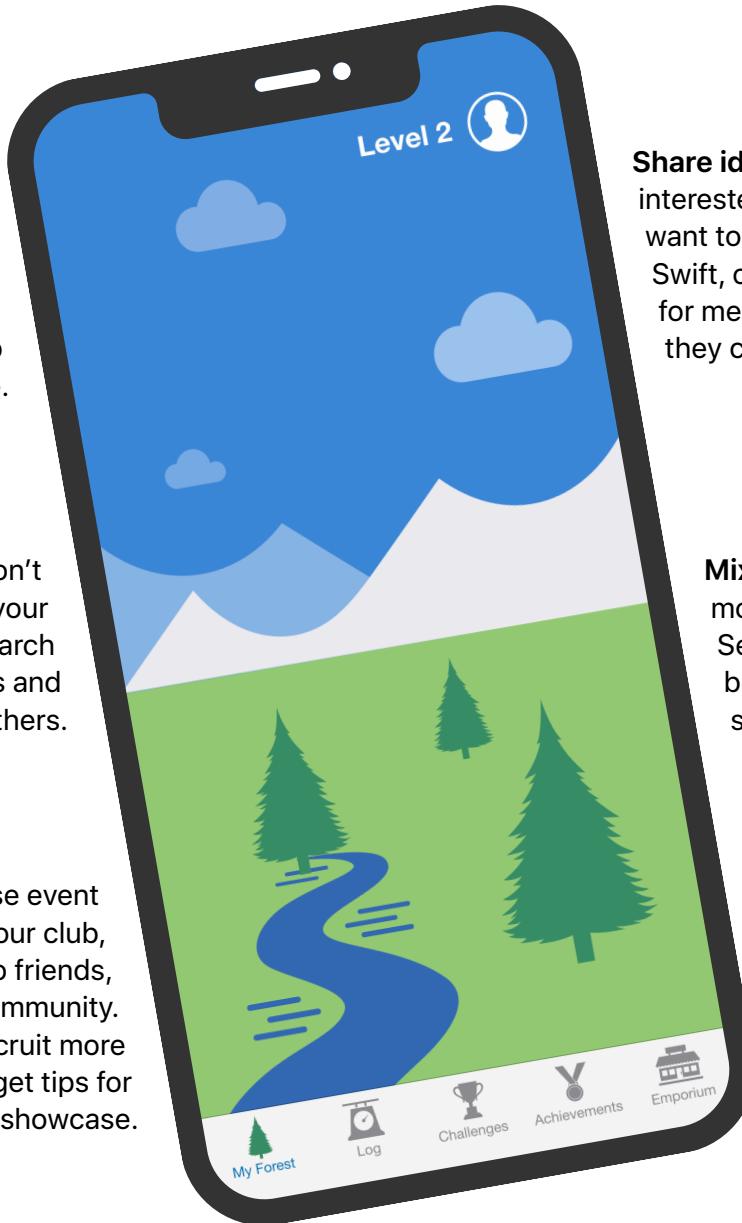
# Tips for club leaders



**Build a leadership team.** Having a group of members who help with leading the club can make it much easier and more fun. Which club members have leadership potential? Think about adding officers to your club for events, coding, app design, and more.

**Learn together.** Club leaders don't have to know everything. Help your members work on their own research and problem-solving skills and encourage them to help others.

**Show off.** An app showcase event is a great way to promote your club, app ideas, and coding skills to friends, families, educators, and the community. It might even help you recruit more members. See page 12 to get tips for holding your own app showcase.



**Share ideas.** Some members will be interested in making games. Others might want to create apps to help people, learn Swift, or control robots. Think about ways for members to work together on projects they care about.

**Mix it up.** Sometimes members who are more advanced can leave others behind. See if those members can partner up with beginners for pair programming. Teaching someone else is a great way to learn!

# Learn and Apply



## 1. Learn about Swift

Swift is a powerful and intuitive programming language created by Apple for building apps. It's the same language professional developers are using in the fast-growing app economy to make apps for iPad, Mac, Apple TV, Apple Watch, and more. Swift makes programming easier, more flexible, and more fun.

To find out more about Swift, visit [swift.org](https://swift.org).

## 2. Dive into Xcode and Develop in Swift Explorations

The club materials are built around app projects for Xcode, an integrated development environment professional developers use to build real apps. It includes a source code editor for writing and managing code, a debugger for diagnosing problems, and a user interface editor—called Interface Builder—for laying out the visual elements of the app and connecting them to the code.

To find out more about Xcode, visit the [Xcode Support](#) page.

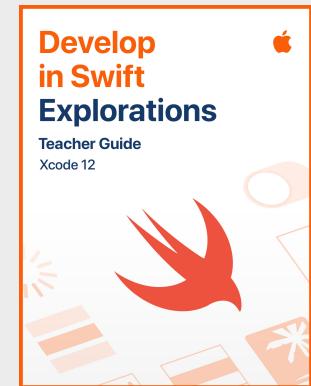
*Develop in Swift Explorations* guides learners through playground activities in Xcode as they learn coding basics. Xcode playgrounds allow you to write Swift code and immediately see the results in a live preview. Playing with code—and seeing what it does—is a great way to get started coding and to experiment with new ideas.

*Develop in Swift Explorations Teacher Guide* includes additional activities you can use with club members to spark their interest, support their understanding, and inspire them to follow and develop their passions.

Download [Develop in Swift Explorations Teacher Guide >](#)



[Download the Xcode app >](#)



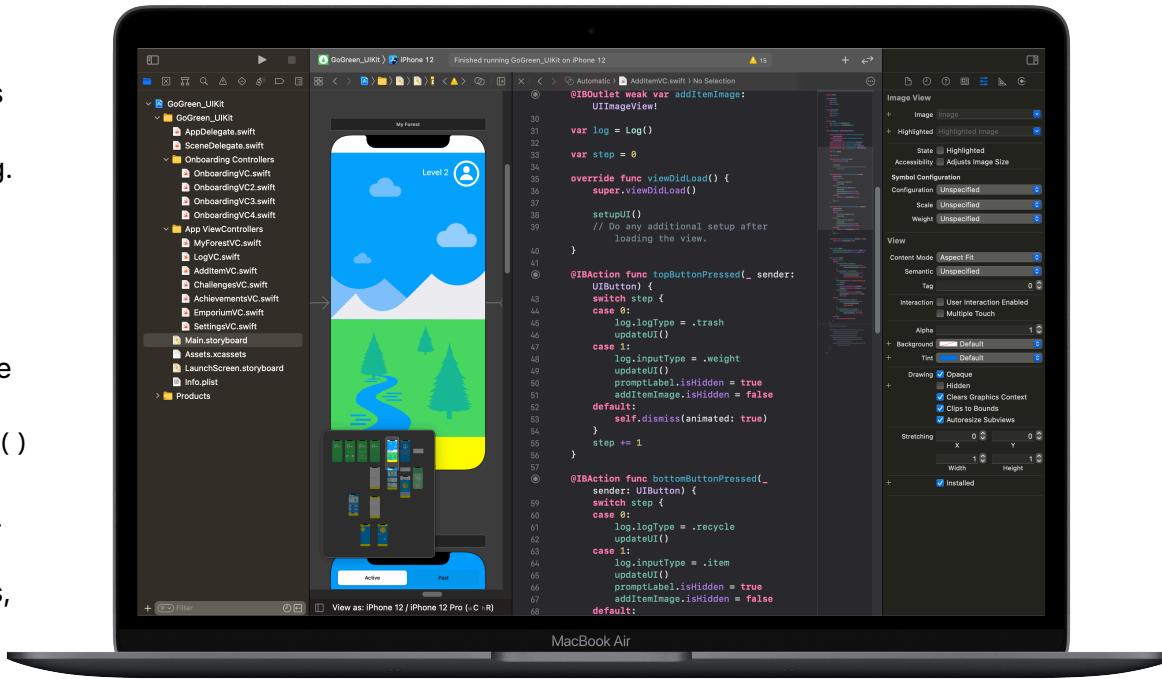
[Download the teacher guide >](#)

# Tips for learning with Xcode



**There's no one way to write code.** Members should review each other's code, provide feedback, and help each other debug.

**Use debugging tools.** When your app crashes, a red highlight shows the line in your code where the error occurred. Use `print()` in your code to log helpful information to the console. Set breakpoints to pause your app, inspect variables, and step through code line by line.



**Stop and think.** Bugs are inevitable. Stop and think about the problem. What are its symptoms? Was everything working fine until [x] occurred?

**Take it further.** Advanced club members can work through the Develop in Swift Explorations course and focus more on coding aspects of their Xcode app prototypes.

**Explore Xcode preferences.** Set your text editing and other preferences by choosing Xcode > Preferences from the menu bar. You can add developer accounts, customize navigation or fonts, choose certain behaviors when events occur, and more.

**Set up a help desk.** Maintain a space where club experts can provide support to their peers.

**Master keyboard shortcuts.**  
Build and run the project: ⌘ R  
Comment or uncomment selected code: ⌘ /  
Reindent selected code: ⌘ I  
Show the inspectors: ⌘ ⌂ O  
Show documentation: ⌘ ⌄ O



### 3. Choose your projects

Swift Coding Club materials are organized around app project modules and an app design challenge. Club members learn programming concepts as they explore Xcode by completing a series of playground activities and guided app development projects. The Develop in Swift Explorations course provides everything they'll need to complete each module.

The first few app projects require no prior knowledge and then begin to increase in difficulty. Check the complexity of each project, and choose a starting point that's suited to your club members' coding experience.

The app design challenge is designed to be completed either concurrently with another module or as a standalone challenge.

**Module 1: PhotoFrame App**

**Module 2: QuestionBot App**

**Module 3: ColorMix App**

**Module 4: ElementQuiz App**

**Module 5: App Design Challenge**



[Download the course >](#)



## 4. App design challenge

While club members are learning to build apps in Xcode, they can also begin to concept and share ideas for apps they'll want to build and help prototype ideas, test the app with peers, and refine user experiences. The app design challenge lets them work beyond their programming skills, unleashing their creativity and ingenuity.

Club members will learn about the features of successful apps and what to consider when designing their own. They'll work through the Develop in Swift App Design Workbook to engage in the app design process and develop a working app prototype that they can share through an app showcase. Club members can work on an app project for part of each session and their own app idea for the remainder of the sessions, or alternate sessions between projects.





## 5. Take it further

You can also add sessions that support your members' interests. You can expand on design and coding activities by investigating apps for connected devices or for watchOS. Or go deeper into topics like machine learning and augmented reality to explore the future of app design.

To prompt design brainstorming, you might even add guest speakers or field trips to help club members better understand a project's audience and design requirements.



# Celebrate



## Community event or virtual app showcase

Host a community event or virtual app showcase to involve the broader community and explore the potential of code for solving challenges they care about. These events are also the perfect way to show off your club members' talents!

**1. Plan the big event.** Set a date and invite peers, educators, families, and community members.

Allow time for each team to present their project and hold a short Q&A session either in person or virtually. If the group is large, you can split the event into two rounds and have members watch each other's presentations.

Consider finishing the event with a fun slideshow of photos taken throughout club sessions.



**2. Design awards.** Friendly competition can be a great motivator. Inspire club members by offering awards that recognize specific strengths in coding and design—for example:

- Best Engineering
- Best Innovation
- Best Design
- Best Presentation

You can also encourage audience participation with a People's Choice award.



Download and modify this [certificate](#) for different awards.



**3. Recruit judges and mentors.** They can be educators or staff, peers with expertise in coding, experts from the developer or design industry, local community leaders, or individuals who would benefit from project ideas.

Judges don't have to wait until the showcase to meet the club. Consider inviting them as guest speakers to share their knowledge while learners are in the brainstorming or planning phase of their projects.

**4. Share and inspire.** You may want to record the presentations. Share them with the broader community and create a highlight reel to inspire future club members.





**Develop in Swift**  
Swift Coding Club

# Certificate of Achievement

Awarded to

For

---

Signature

---

Date

# Swift Coding Club Modules

Module 1: PhotoFrame App

Module 2: QuestionBot App

Module 3: ColorMix App

Module 4: ElementQuiz App

Module 5: App Design Challenge



# PhotoFrame App

Module 1



# PhotoFrame App

## Module 1 Overview

Find out just how easy it is to build your very first app. In this module, you'll learn the key concepts and skills you need to build an app that can display basic UI components such as a photo. Understanding the basics of UI components is critical to creating any app, so it will come in handy as you expand your coding and app development skills. With this project, you'll also familiarize yourself with Xcode, Interface Builder, and Simulator, and you'll learn how to use them together to build your own apps.

### Sessions 1–7

Learn about values and experiment with values, variables, and constants in Xcode playgrounds.

- Playground Basics
- Naming and Identifiers
- Constants and Variables
- Strings

### Sessions 8–9

Apply new skills and concepts to create a word game playground.

### Sessions 10–12

Build a PhotoFrame app in Xcode and Interface Builder.



# PhotoFrame App

## 1 Get Started with Xcode

Find your way around Xcode playgrounds, and learn how to enter and change code in basic programs.

**Get Started:** Explore programming basics and the roles of data inputs and outputs.

What is Programming? (page 15)  
Values (page 16)

**Play:** Learn how to enter and change code.

Playground Basics playground  
(pages 27–29)

## 2–3 Naming and Identifiers

Learn about the importance of naming in programming, and build some simple programs to solve problems.

**Get Started:** Find out why naming and identifiers are important, and name key components for a new game as you sketch your design for it.

Naming and Identifiers (pages 18–19)

**Play:** Practice your naming skills as you build programs to solve simple problems.

Naming and Identifiers playground  
(pages 30–33)

## 4–5 Constants and Variables

Learn how to declare variables and constants, and build a program to keep track of a score.

**Get Started:** Compare variables and constants, and imagine your life as a program.

Constants and Variables (page 20)

**Play:** Build a program to help you track a game score.

Constants and Variables playground  
(pages 38–41)

# PhotoFrame App

## 6–7 Strings

Learn about strings and how to use them in your code to create a simple game.

**Get Started:** Identify key properties of strings, and create your own chatbot responses.

Strings (pages 23–24)

**Play:** Build a fill-in-the-blanks game.

Strings playground (pages 42–44)

## 8–9 Word Games

Use your knowledge of values, constants, and strings to create word games in playgrounds, then have club members try them out.

**Apply:** Build a word-substitution game that results in funny stories.

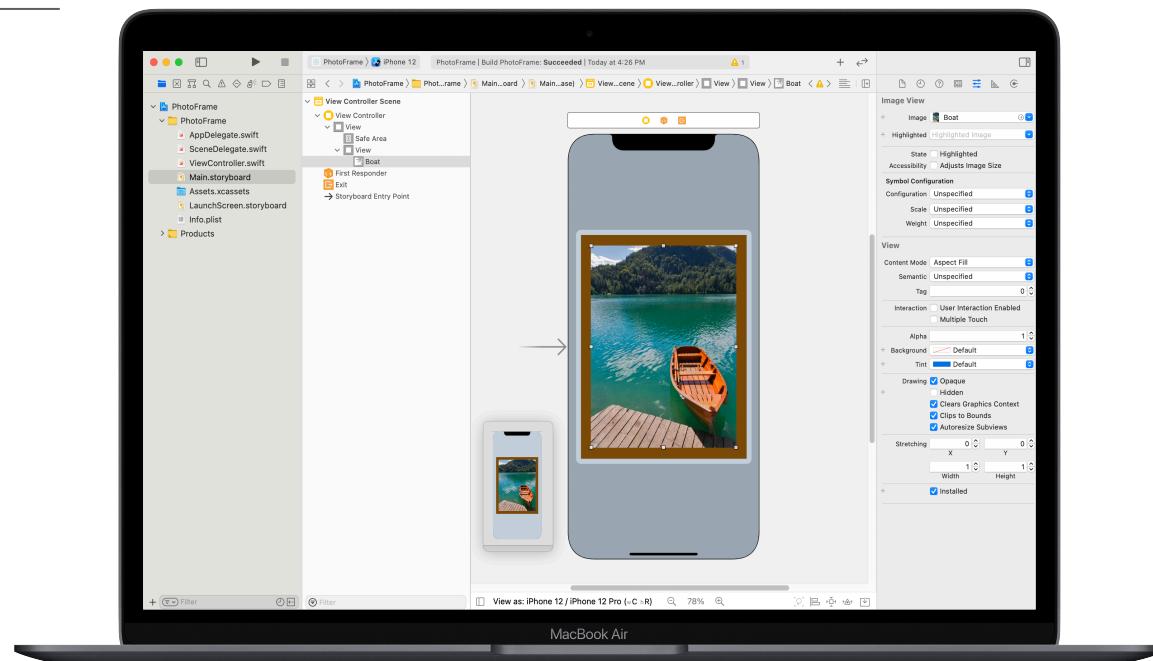
Word Games playground (pages 45–46)

## 10–12 PhotoFrame App

Find your way around Xcode Interface Builder, and build and run a simple app.

**Apply:** Build and view an app that displays a custom-framed photo.

PhotoFrame app project (pages 48–73)



# QuestionBot App

## Module 2



# QuestionBot App

## Module 2 Overview

Have you ever used a quiz app or wondered how Siri works? All apps have internal logic that defines their behavior. With QuestionBot, you'll build an app containing the "brain" of a bot that responds differently to different questions. To do that, you'll learn how to design algorithms, group code in functions, use different types, and more. This module helps you focus on understanding how apps work, learn about creating the code that controls the interface of an app, and explore logic you can use to make an app mimic human intelligence.

### Sessions 1–10

Learn about algorithms as the cornerstone of programming, and experiment with functions, types, and parameters in Xcode playgrounds.

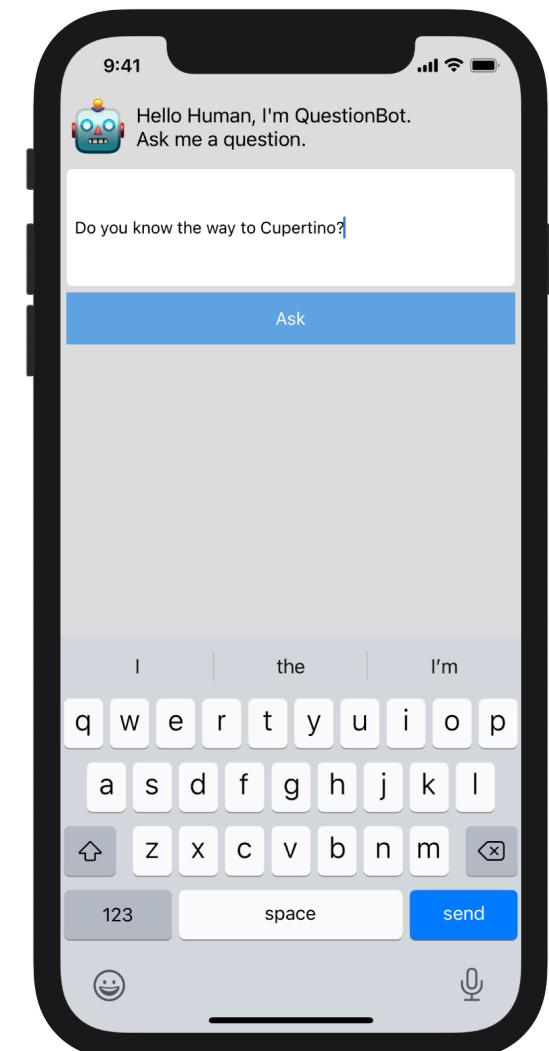
- Algorithms
- Functions
- Types
- Parameters
- Making Decisions with Booleans

### Sessions 11–12

Apply new skills and concepts to create dance routines in the BoogieBot playground.

### Sessions 13–14

In Xcode, add functionality to program the "brain" of a QuestionBot app to answer questions.



# QuestionBot App

## 1–2 Algorithms

Learn about algorithms as a key programming tool, and practice designing algorithms to solve everyday problems.

**Get Started:** Use sequencing and selection in algorithms to solve simple problems, and design an algorithm for a program that matches music to your mood.

Algorithms (page 109)

Sequencing (page 110)

Selection (page 111)

## 3–4 Functions

Learn about functions for creating reusable sections of code, and build a song maker.

**Get Started:** Practice grouping instructions under functions in a familiar context, like preparing dinner.

Functions (pages 112–114)

**Play:** Build a program that makes a repetitive song.

Functions playground (pages 121–124)

## 5–6 Types

Learn about types as a way to distinguish different types of data, and build a program to perform simple calculations.

**Get Started:** Explore types as a way of describing values and consider the types of parts you'd use in a construction task.

Types (pages 115–116)

**Play:** Build a program to complete simple calculations.

Types playground (pages 125–127)

# QuestionBot App

## 7–8 Parameters

Learn about using parameters to define inputs to functions, and build a program that outputs different sentences based on the values you enter.

**Get Started:** Use parameters to make your functions more flexible, and refine your dinner preparation functions to accommodate specific requirements.

Parameters (pages 116–117)

**Play:** Build a program that uses functions to output a different sentence based on the values you pass into it.

Parameters and Results playground (pages 128–130)

## 9–10

## Making Decisions with Booleans

Learn about how powerful the Boolean type can be in programming, and build a program to determine if a given year will be a leap year.

**Get Started:** Explore Booleans and use them to help a robot out of a tricky situation.

Making Decisions with Booleans (page 118)

**Play:** Build a program that determines if a year is a leap year.

Making Decisions playground (pages 131–134)

## 11–12

## BoogieBot

Use functions to build complex dance routines out of smaller parts, and share your choreography with others as animated images.

**Apply:** Create dance routines for BoogieBot and save your creations as animated images.

BoogieBot playground (page 135)

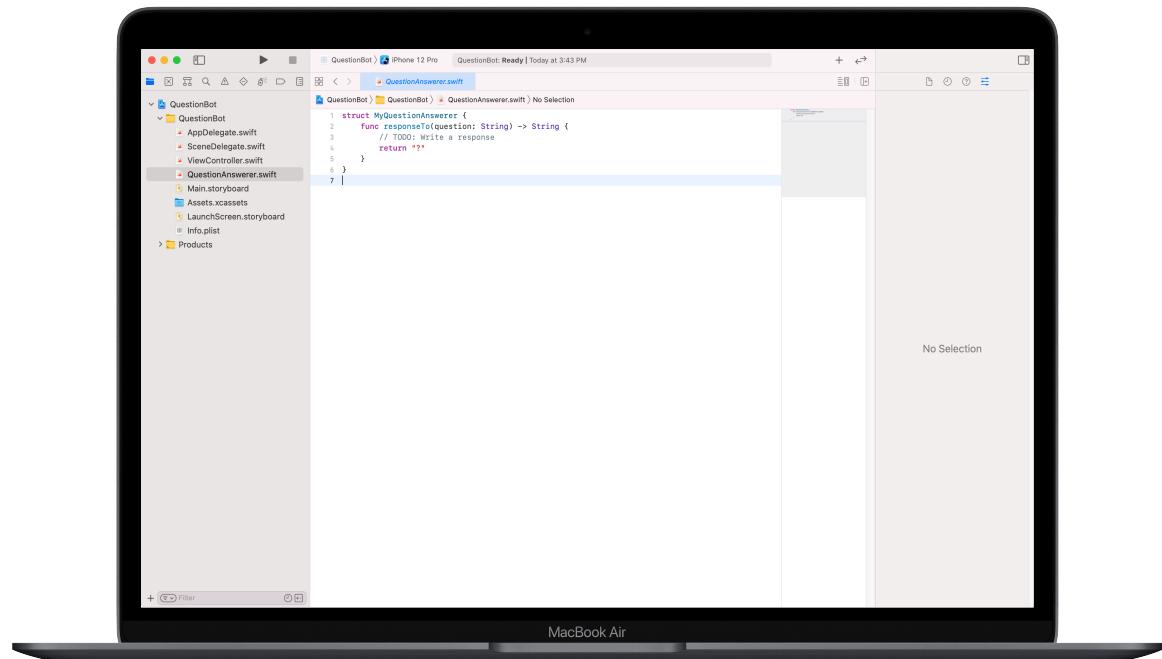
# QuestionBot App

## 13–14 QuestionBot

Build the logic of a QuestionBot app so that it responds differently to different questions.

**Apply:** Program the “brain” of a QuestionBot app to decide how it will answer questions, and learn how to test and troubleshoot your code.

QuestionBot app project (pages 138–150)



# ColorMix App

## Module 3



# ColorMix App

## Module 3 Overview

Think about the iPhone user interface (UI). So far, you've built apps from basic UI elements and learned how to create the logic behind the UI. With ColorMix, you'll learn how to build an interactive app that has controls such as buttons and switches. More importantly, you'll learn how to connect those visual UI elements to Swift code so they work the way you want them to. To do this, you'll learn about defining your own custom types with properties and methods, using instances of types, and collecting data in arrays. In the end, you'll have ColorMix, an app that generates all the colors of the rainbow—and more—by mixing red, green, and blue.

### Sessions 1–6

Learn about organizing data, experiment with defining methods and properties for a custom type, and work with arrays in Xcode playgrounds.

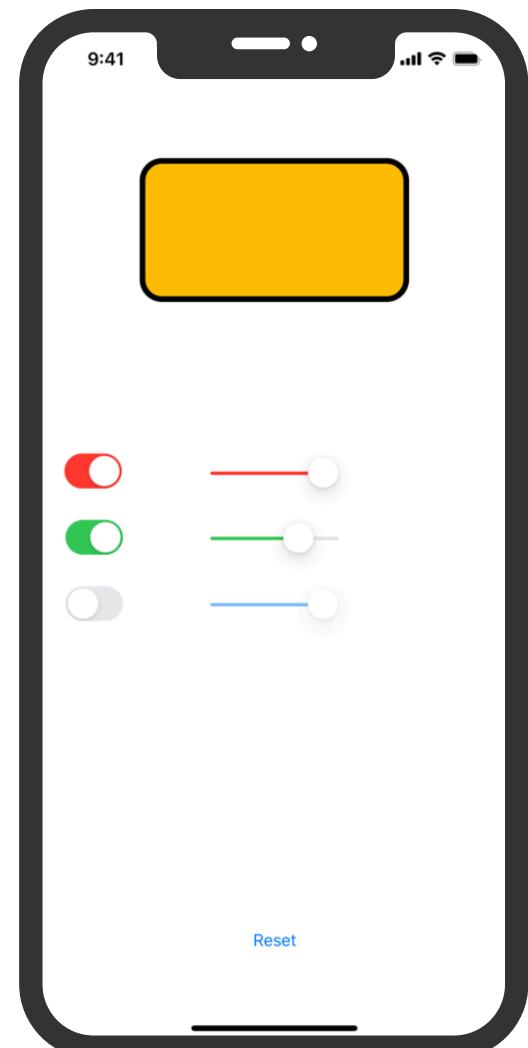
- Instances, Methods, and Properties
- Arrays and Loops
- Structs

### Sessions 7–8

Learn how graphics are created, then build graphics, emoji, and animated line drawings one pixel at a time.

### Sessions 9–12

Build a ColorMix app by adding switches and sliders to the UI.



# ColorMix App

1–2

## Instances, Methods, and Properties

Learn to create instances of types and how to use their methods and properties, and program a robot dance-off.

**Get Started:** Explore how types define methods and properties, and experiment with describing methods and properties for different kinds of animals.

Instances, Methods, and Properties  
(page 184)

**Play:** Build a program to set up a dance-off between two robots.

Instances, Methods, and Properties playground (pages 196–198)

3–4

## Arrays and Loops

Learn how to order data in arrays and how to process arrays using loops, and build a program to tally votes, track progress, and find key words.

**Get Started:** Use iteration in your algorithms, and use loops to work with elements in arrays. Create an algorithm to describe game play in a board game, and consider the ways you could work with a collection.

Lists and Arrays (page 185)  
Algorithms: Iteration (pages 186–187)  
Loops (page 188)  
Working with Arrays—Searching  
(pages 189–190)

**Play:** Build programs to tally votes, track your progress toward a daily goal, and filter messages for key words.

Arrays and Loops playground  
(pages 199–202)

5–6

## Structs

Learn how to create custom types with structs, and use a custom type to solve a programming challenge.

**Get Started:** Define your own types with structs, and create a custom type for an animal of your choosing.

Defining Your Own Types with Structs  
(pages 191–192)

**Play:** Build a program that uses a custom type to solve a problem.

Structures playground (pages 203–205)

# ColorMix App

## 7–8 Pixel Art

Learn how graphics are created, then build your own graphics, one pixel at a time.

**Apply:** Write code to handcraft graphics, emoji, and animated line drawings.

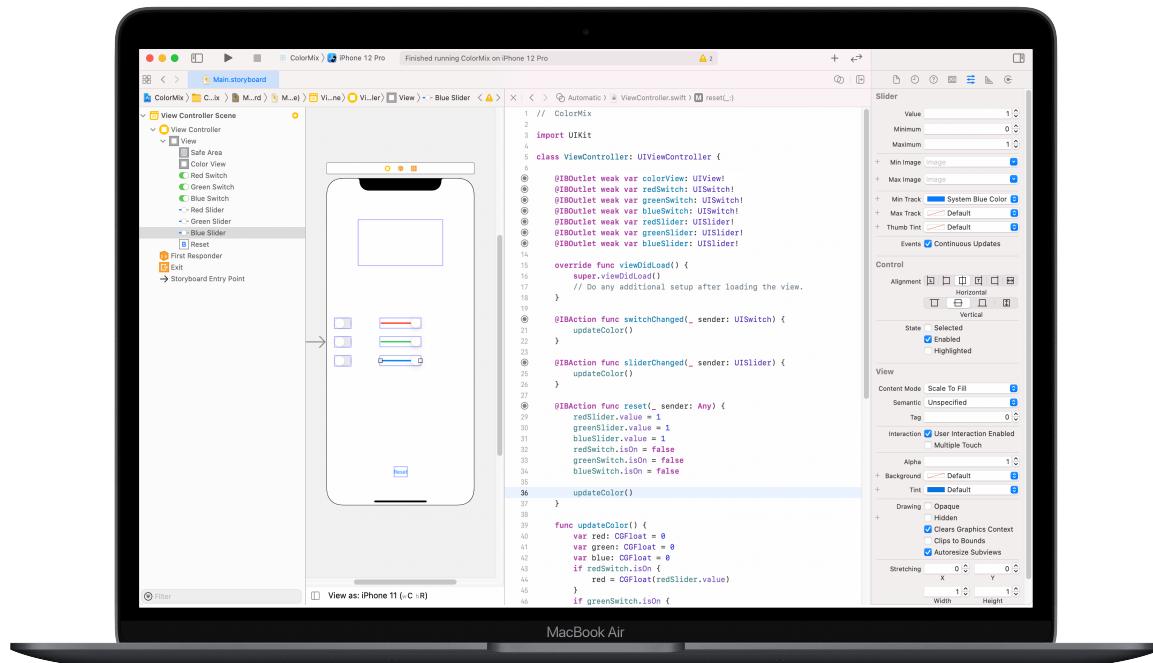
Pixel Art playground (pages 215–216)

## 9–12 Color Picker

Learn how to use actions and outlets to connect your Swift code to your app's UI.

**Apply:** Build an app with switches, sliders, and buttons that lets you mix your own colors.

ColorMix app project (pages 302–346)



# ElementQuiz App

## Module 4



# ElementQuiz App

## Module 4 Overview

Most people use apps to solve a particular problem—to help them get organized, calculate their finances, or get directions, for example. In ElementQuiz, you'll build an app that helps you memorize elements in the periodic table. You'll learn about enums, then apply your learning to tackle an independent app challenge of your choice. You could build a MemeMaker app, create a Rock, Paper, Scissors game, or take the ElementQuiz app to the next level.

### Sessions 1–4

Follow the steps to build the flash card interface of an ElementQuiz app.

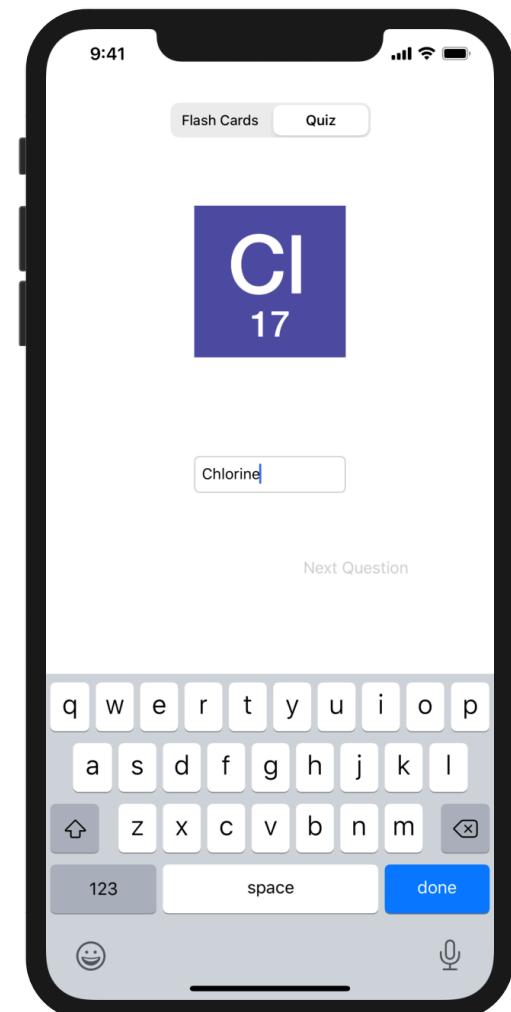
### Sessions 5–6

Learn about enums, and build a program to tally votes.

- Enums and Switch

### Sessions 7–12

Choose one of three app projects to build.



# ElementQuiz App

---

1–4

## ElementQuiz App: Part A

Learn how to build a flash card app that helps you to memorize elements in the periodic table.

**Apply:** Build a Quiz app with a flash card interface that helps users learn the chemical symbols of elements.

ElementQuiz app project, parts 1–3  
(pages 402–417)

---

5–6

## Enums and Switch

Learn about enums and build a program to tally votes.

**Play:** Build a program that tallies the results of polls.

Enums and Switch playground  
(pages 206–208)

# ElementQuiz App

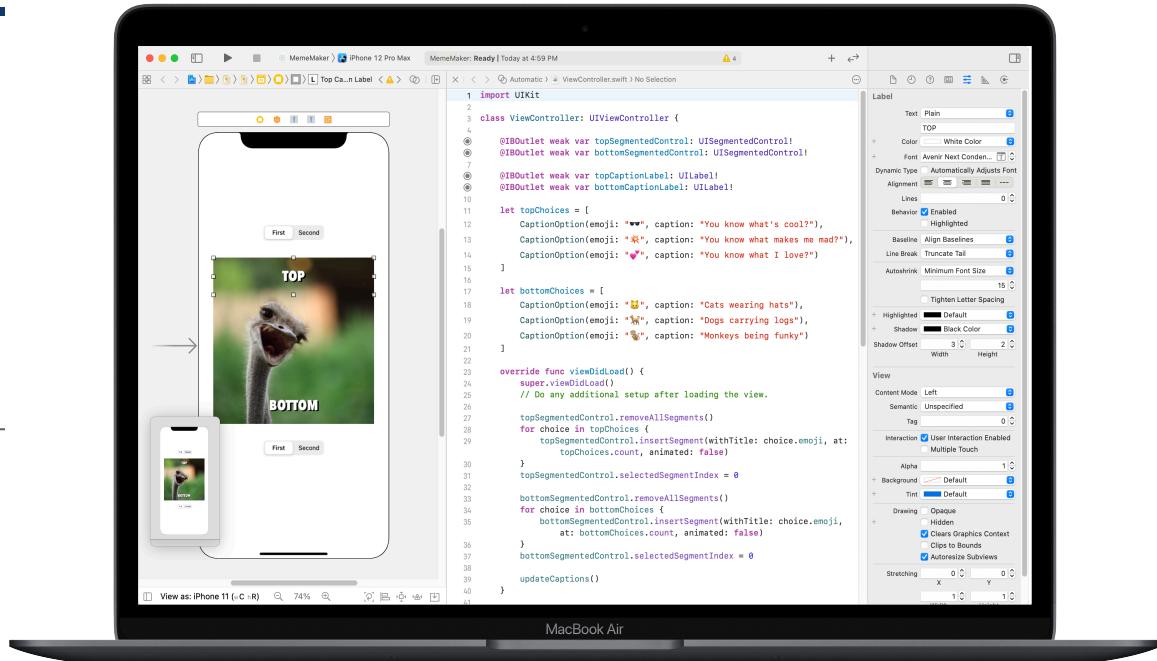
For sessions 7–12, select from one of the following three app projects. Stars indicate level of difficulty.

## 7–12 MemeMaker App

Learn how to use segmented controls to display different captions above and below an image. Since the controls are independent, you can mix and match the text to create customized combinations. Learn how to use gesture recognizers to let the user drag captions around the screen.

**Apply:** Build an app that lets you add fun customized captions to a photo based on your mood.

MemeMaker app project (pages 385–400)



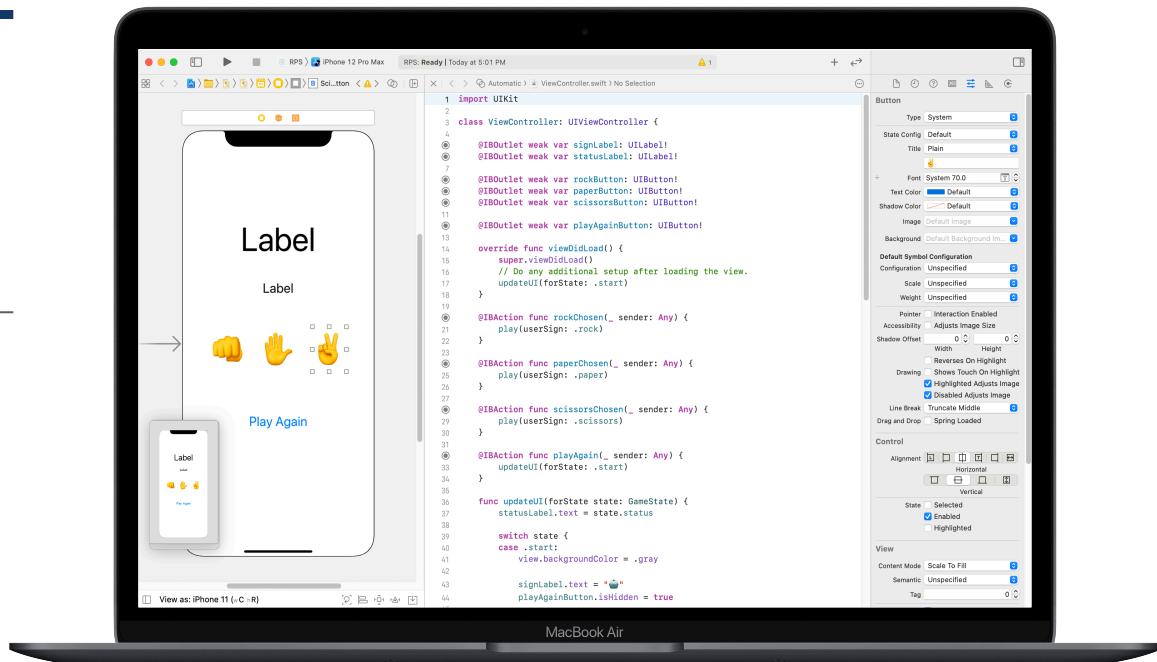
# ElementQuiz App

## 7-12 Rock, Paper, Scissors App

Learn how to customize structs and enums to create the model and logic for a game of Rock, Paper, Scissors; use random numbers so that users can play against the computer endlessly.

**Apply:** Build a game app using emojis and buttons.

Rock, Paper, Scissors app project  
(pages 369–384)



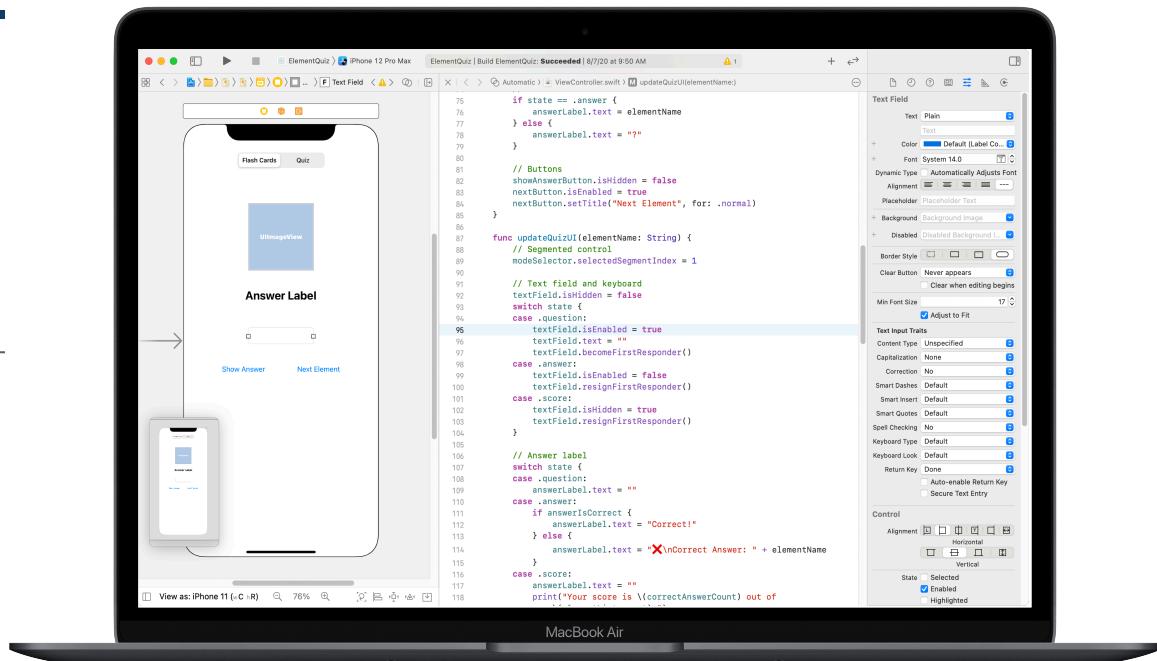
# ElementQuiz App

## 7-12 ElementQuiz App: Part B

Learn how to handle text input to create a scored quiz mode in the ElementQuiz app. Learn how to structure user interface logic and how to refactor your code as it becomes more complex.

**Apply:** Extend the Quiz app to include a scored quiz mode.

ElementQuiz app project, Parts 4–10  
(pages 417–468)



# App Design Challenge

## Module 5

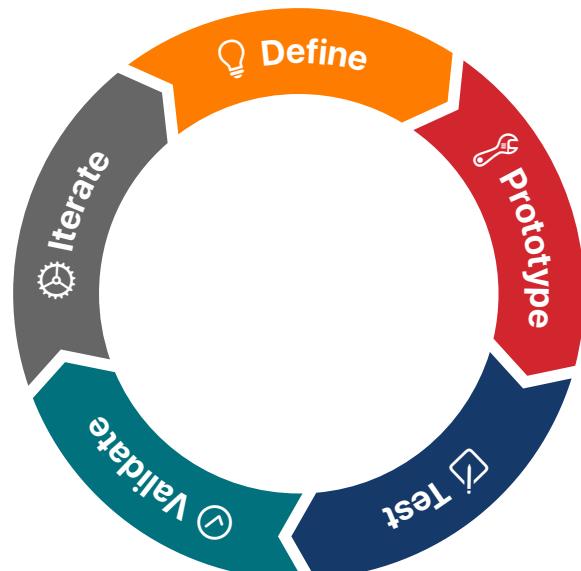


# App Design Challenge

## Module 5 Overview

In this module, club members use the Develop in Swift App Design Workbook to design an app to help solve a challenge they care about. Club participants will use a design thinking framework to learn app design—a fundamental skill of iOS app development. They will explore the relationship between app design and coding in Swift through each stage of the app design cycle to bring their app ideas to life.

You can then use the App Showcase Guide to help members plan an app pitch video that documents their process and shows off their app design. Then host an app showcase to celebrate your club members' ingenuity.



**App Design Cycle**

## Resources

 **Develop in Swift**  
App Design Workbook



[Develop in Swift App Design Workbook >](#)

 **App Showcase Guide**



[App Showcase Guide >](#)

# App Design Challenge

Club participants form small teams, identify a challenge they care about, then design an app to help solve it. Using the Develop in Swift App Design Workbook, they follow a design process of defining app ideas, building a working prototype in Keynote, and testing and evaluating the app with users before taking their discoveries back to refine their app.

 <b>Define</b> <hr/>	 <b>Prototype</b> <hr/>	 <b>Test</b> <hr/>	 <b>Validate</b> <hr/>	 <b>Iterate</b> <hr/>
Discover a challenge and the people it affects. Then analyze it and plan a detailed solution.	Map and build a wireframe for the features of your app. Then create a prototype that adheres to iOS design guidelines.	Create a plan to test your prototype. Then script and prepare to conduct your tests.	Gather and organize the data you collected in your tests, then form key insights and conclusions.	Make a plan to address the issues you discovered by revisiting specific parts of the design process.



© 2021 Apple Inc. All rights reserved. Apple, the Apple logo, Apple TV, Apple Watch, iPad, iPhone, Keynote, Mac, MacBook Pro, macOS, Siri, Swift, Swift Playgrounds, the Swift logo, watchOS, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. tvOS is a trademark of Apple Inc. App Store is a service mark of Apple Inc., registered in the U.S. and other countries. IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license. Other product and company names mentioned herein may be trademarks of their respective companies. April 2021