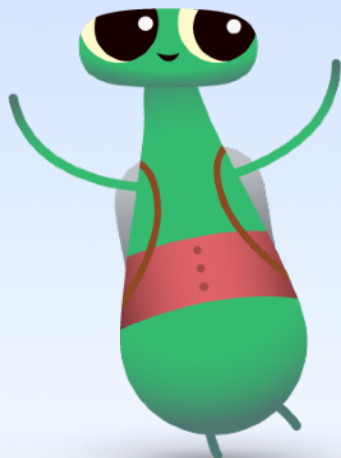




Swift Coding Club



# Swift Playgrounds Kit

# Welcome to the Swift Coding Club!

Learning to code teaches you how to solve problems and work together in creative ways. And it helps you build apps that bring your ideas to life.

Swift Coding Clubs are a fun 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 a teacher or a coding expert 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 kit is arranged in three sections:



## Get Started

Everything you need to launch a Swift Coding Club.



## Learn & Design

Tips and activities for designing club sessions.



## Celebrate

Helpful resources to plan and host an app showcase in your community.

## Swift Coding Clubs

### Block-Based Coding | Ages 8–11

Learn coding basics using visual apps on iPad.



### Swift Playgrounds | Ages 11+

Use Swift code to learn coding fundamentals with Swift Playgrounds on iPad.



### Xcode | Ages 14+

Learn to develop apps in Xcode on Mac.



# Get Started



## 1. Download club materials.

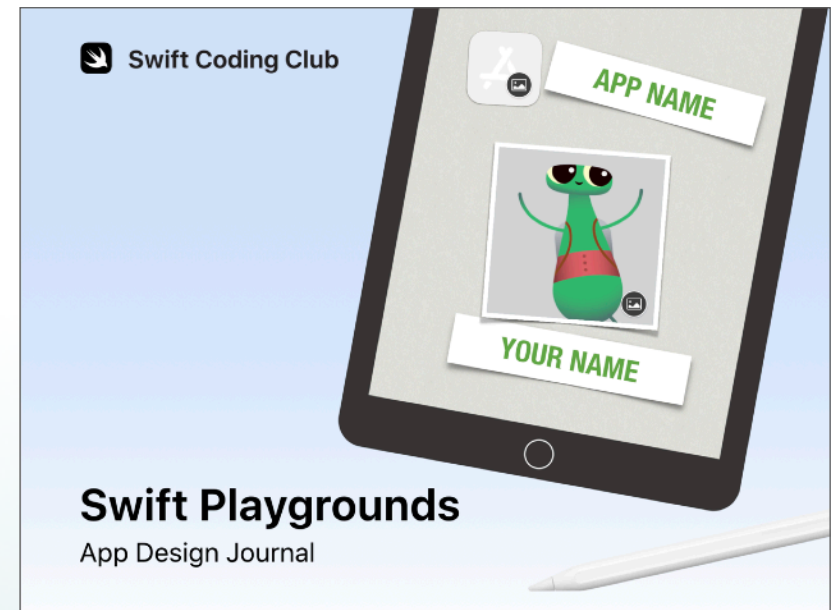
Use AirDrop to share these two guides with club members in your first club meeting. They're also included as part of this document.



### Coding Activities

Learn coding concepts with these fun, collaborative activities and solve puzzles with the Swift Playgrounds app on iPad.

[Download Swift Playgrounds Coding Activities >](#)



### App Design Journal

Explore the app design process with this Keynote journal. Brainstorm, plan, prototype and evaluate your club's app ideas.

[Download Swift Playgrounds App Design Journal >](#)



## 2. Check your tech.

Before your first meeting, make sure you have the following:

- **iPad.** iPad mini 2 or later, iPad Air or later, or iPad Pro running iOS 11 or later. It's best if each person has their own device, but they can also share and code together.
- **Swift Playgrounds app.** [Download Swift Playgrounds >](#)
- **Learn to Code 1 and 2 playgrounds.** Download these playgrounds from within the Swift Playgrounds app.
- **Keynote.** You'll use the Keynote app on iPad for your app prototypes.
- **Swift Coding Club materials.**



## 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?
- How often will your club meet? If you're planning a summer camp, 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:

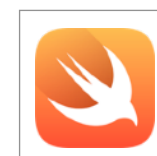
- **Announce your club.** Use email, social media, the web, flyers or word of mouth to let your community know about your club.
- **Host an informational meeting.** Ask potential club members about their interests and what types of app they'd want to create. Talk about ideas for holding an app design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

These items can help you promote and personalise your Swift Coding Club:

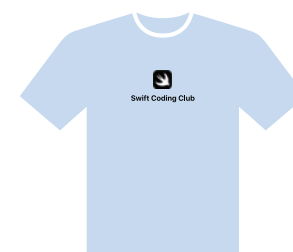
- **Posters.** [Download this free template](#), then personalise it to create your own poster. Print and display it, or make a digital poster to share online. Make sure you include details for when and where the club will meet and how to join.
- **Stickers and T-shirts.** Use these [Swift Coding Club stickers](#) to help promote your club. T-shirts are a great way to recognise 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

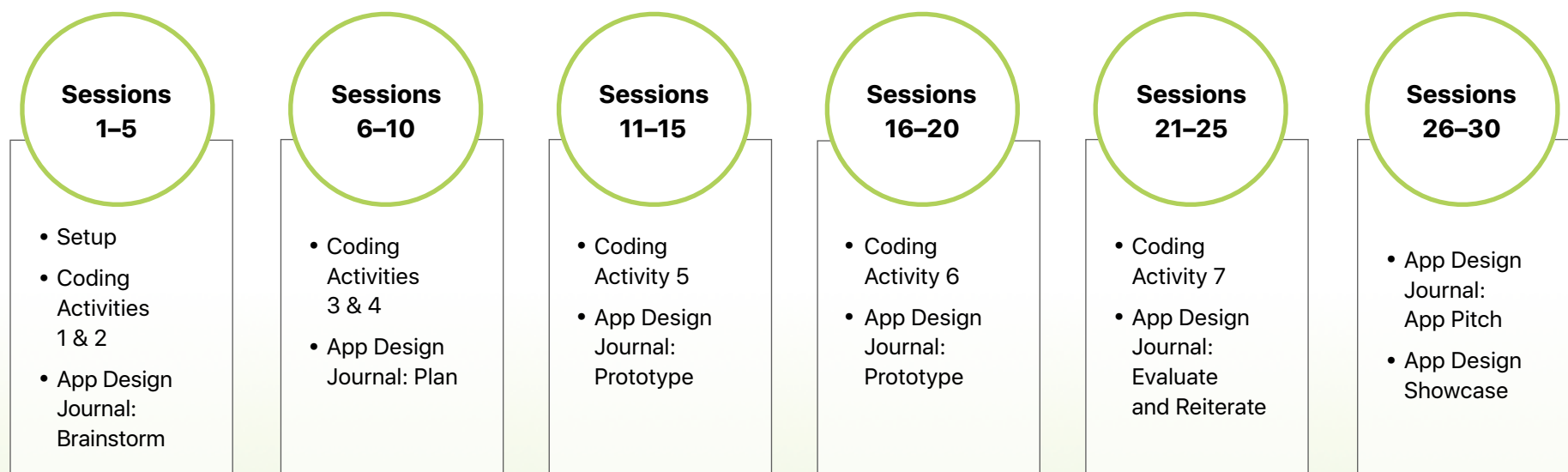


Swift Coding Club T-shirt



# Learn & Design

The club materials are designed for you to interweave coding and app design activities. You can also add sessions that support your members' interests. Below is a sample schedule for 30 one-hour club sessions.



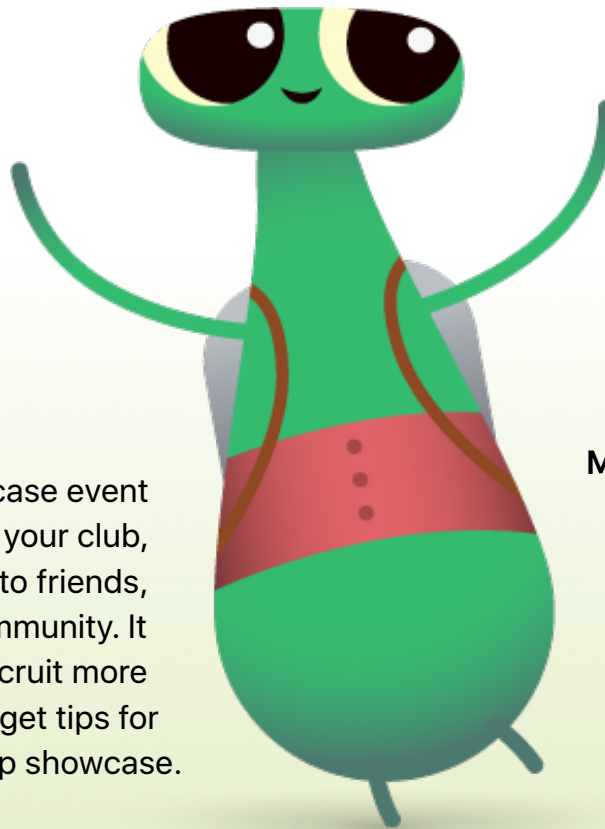
Consider adding sessions to expand on app design and coding activities, like building a drone obstacle course or creating a robot rescue mission challenge. To prompt app design brainstorming, you might even want to add guest speakers or field trips.



**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, teachers and the community. It might even help you recruit more members. See [page 13](#) 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!

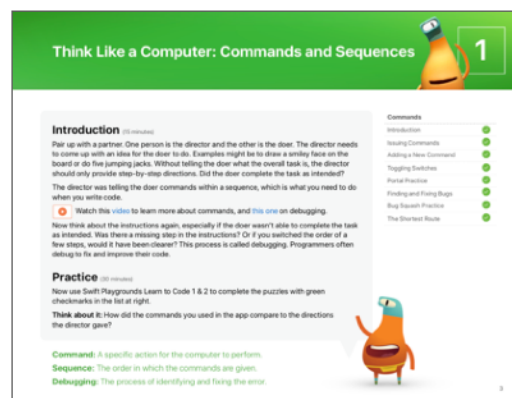




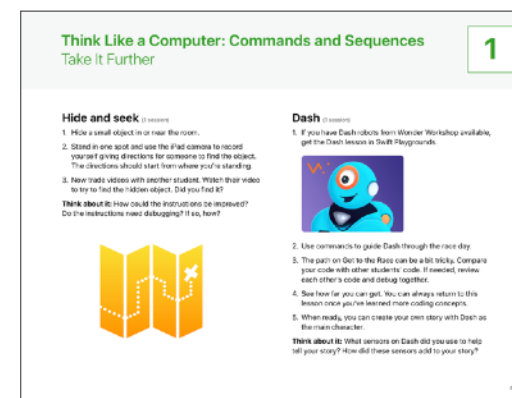
# Swift Playgrounds Coding Activities



**Coding activities:** Built around Swift Playgrounds, these collaborative activities introduce fundamental coding concepts and skills.

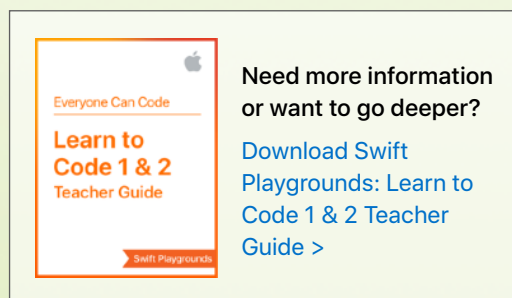


**Coding concepts:** In each activity, club members will learn about a fundamental coding concept and explore it in an everyday context. They'll then apply the coding concept to solve puzzles in Swift Playgrounds.



**Take It Further:** Each coding concept has two Take It Further activities. The first activity deepens understanding of the coding concept and fosters communication and teamwork. Members use iPad to apply their understanding in a creative project.

The second optional activity challenges members to apply the concept in a playground from the Challenges, Starting Points and Subscriptions sections of Swift Playgrounds. Some activities require specific connected devices.





## Tips for Learning with Swift Playgrounds



**Explore the puzzles first.** Encourage club members to zoom and rotate Byte's world in the live view so they can take a good look at what they need to accomplish. They can also view it full screen by touching and holding the partition between the two windows, then dragging to the left.

**Break down the puzzles.** The puzzles get tricky. Club members can divide a puzzle into parts to help them think through all the steps to solve it. They can use Pages or Notes to plan and write out their steps before entering the code.

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



**Solve in multiple ways.** Each puzzle has many solutions. If members finish early, encourage them to think of different ways to solve the puzzles. Thinking flexibly and comparing different solutions can help them improve their critical thinking skills.


**Pair coding.** Have club members try working together on one iPad. They can brainstorm on how to solve the puzzles and take turns writing the code.

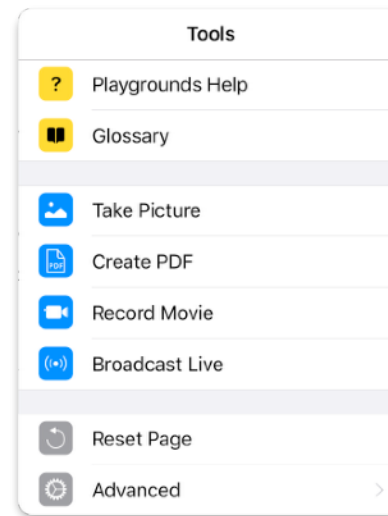
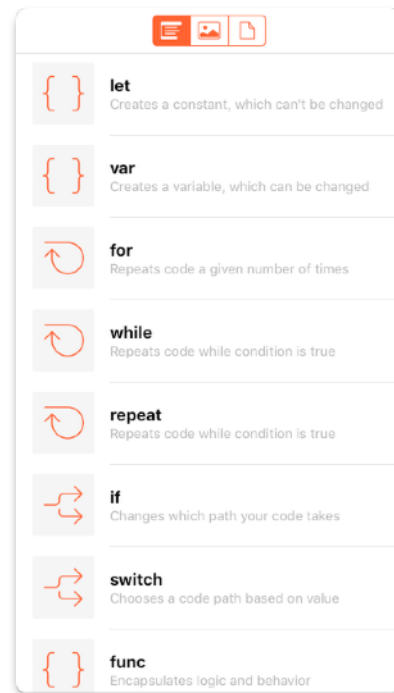
**Use accessibility features.** Swift Playgrounds works well with the built-in assistive features in iOS so that everyone can learn to code. For example, coders can invert the colours, enable greyscale, and zoom to adjust visibility.

# Explore Swift Playgrounds

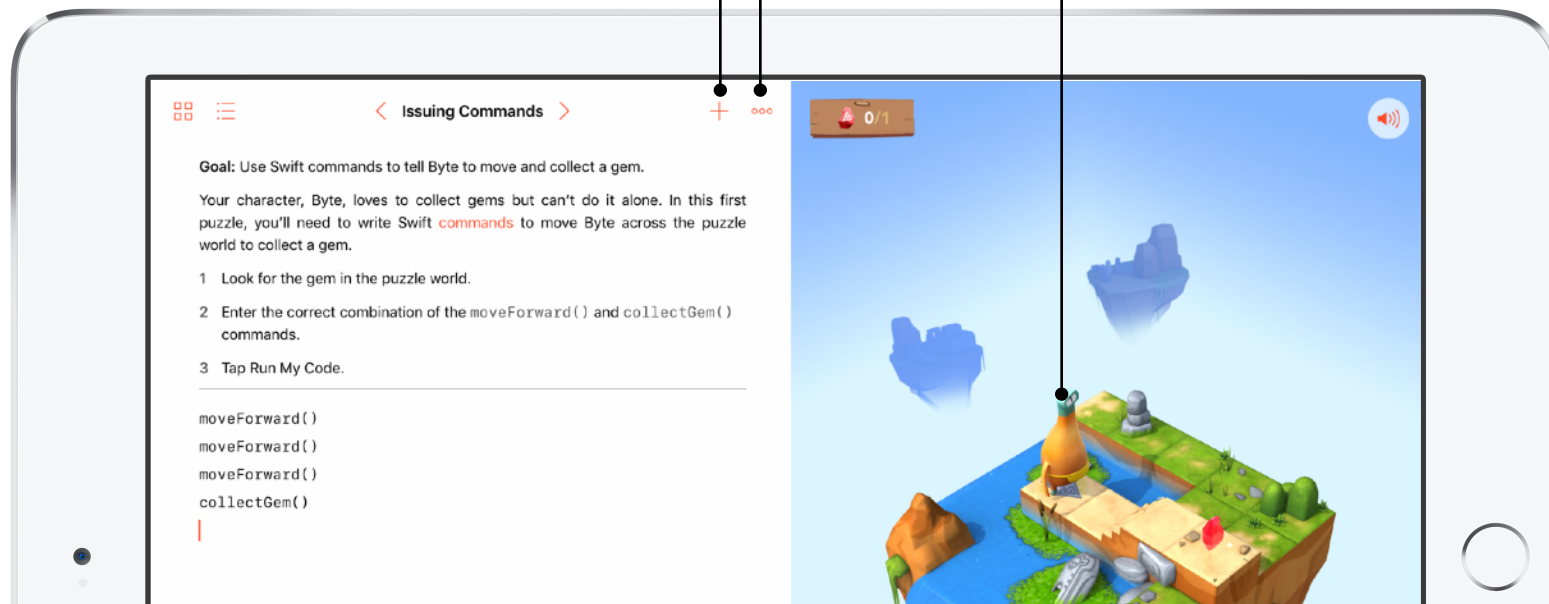


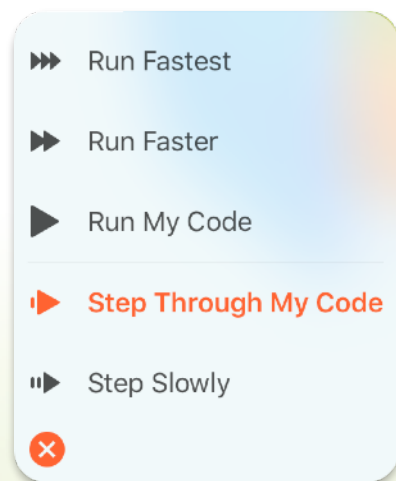
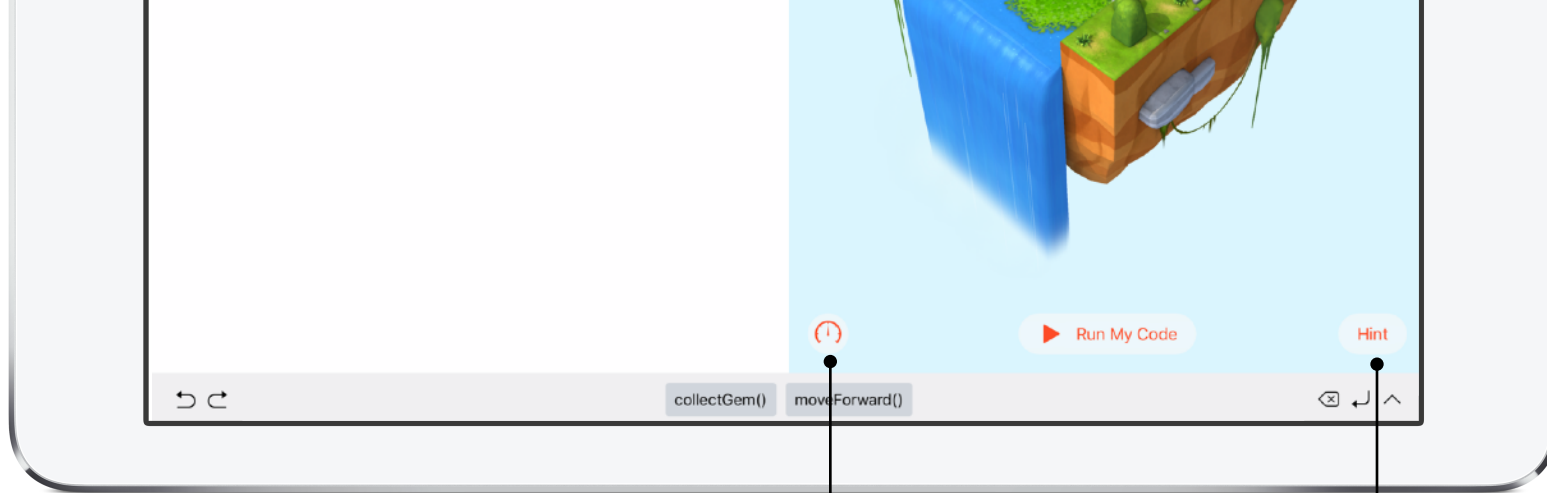
## Snippets Library.

To minimise typing, tap  the toolbar to access the Snippets Library and quickly drag commonly used pieces of code.



**Choose a character.** Personalise your experience by tapping the character to choose a different one.





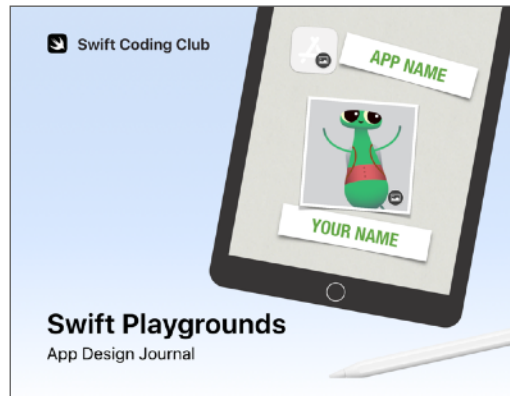
**Hint.** This feature provides suggestions to help learners. It also reveals a puzzle's solution eventually, though coders can't simply cut and paste the solution. To move on, they still have to complete the steps and write the code themselves.

**Control the speed.** Speed up or slow down the code.

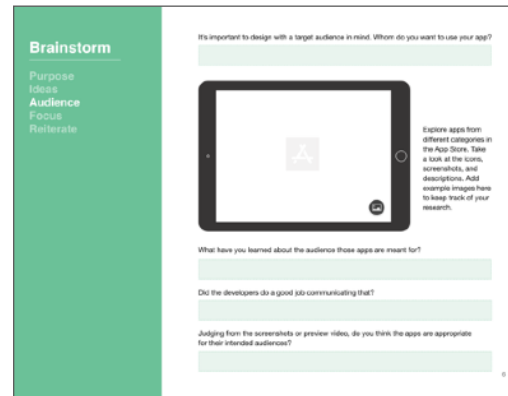
**Highlight code as it runs.** Use Step Through My Code to highlight each line of code as it runs to better understand what the code is doing.



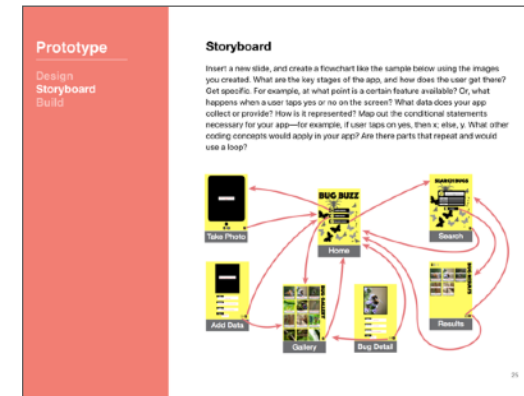
# App Design Journal



Coders use this Keynote journal to learn about app features and design an app to solve a community problem.

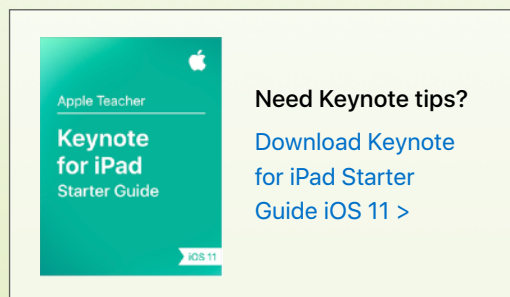


Club members work in small teams to brainstorm and plan the app solution, then create a working prototype of the app in Keynote.



The journal walks coders through the process of evaluating their designs and iterating their prototypes — just like professional app designers.

Members create a three-minute app pitch presentation or video and celebrate their work in an app design showcase.



# Celebrate



## App design showcase

The app design process and the showcase are powerful opportunities to involve the wider community and explore the potential of apps for solving contemporary problems. The showcase is also the perfect way to show off the talents of your club members!

**1. Plan the big event.** Set a date for the showcase and invite students, teachers, parents and community members to attend.

Allow time for each team to present their app pitch and to hold a short Q&A session. If you have a large group, you can split the club into two rounds where members can watch each other's pitches.

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 recognise specific strengths in app design. Consider awards for:

- Best Engineering
- Best Innovation
- Best Design
- Best Pitch

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



You can download and modify this [certificate](#) for different awards.



**3. Recruit judges and mentors.** Judges and mentors can be teachers or staff, students with expertise in coding, experts from the developer or design industry, school governors, local community leaders, or individuals who would benefit from the app idea.

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

**4. Pick a winner.** Judges can use the rubric on the next page to help them evaluate the app pitches and provide feedback. You could also share the rubric with coders before the showcase as part of the evaluation phase of the app design process.

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



# Evaluation Rubric

[Download >](#)

Category	Novice (1 point)	Intermediate (2 points)	Proficient (3 points)	Mastered (4 points)	Points
<b>Pitch Content</b>	Shares basic information, such as purpose and target audience	Gives clear explanation of the app's purpose, design and how it addresses user needs	Presents clear and compelling explanation of the problem they're trying to solve, market demand, audience and how the app was designed to meet user needs	Makes a persuasive pitch backed by evidence that shows how the app meets, exceeds or redefines user needs	
<b>Pitch Delivery</b>	Informational; one team member presents	Confident, enthusiastic; more than one team member presents	Engaging, good use of visuals to support story; team highlights contributions of each member	Creative, memorable storytelling; engaging visual support; smooth transitions between team members	
<b>User Interface</b>	Consistent screens that support app's purpose	Clear, functional design with familiar elements; prototype supports basic user tasks	Elegant, concise, pleasing design with thoughtful use of colour, layout and readability; prototype gives user a sense of place within navigation	Design empowers the user to interact with content; prototype uses animation, colour and layout to create a seamless, engaging experience	
<b>User Experience</b>	Clear intent; users can accomplish one or more goals	Consistent and standard navigation; intuitive path through app content	Adaptable to user needs; addresses accessibility, privacy and security.	Innovative, surprising and delightful; gives users a new kind of experience that sets it apart from competitors	
<b>Coding Concepts</b>	Some connection between app functionality and underlying code	Explanation of how general coding concepts like data types, conditional logic or touch events relate to the app	Description of specific coding tasks necessary to build their app; demonstration of how that code powers the app's functionality	Explanation of the app's architecture, data structure, algorithms and features; discussion of decision-making in developing this approach	
<b>Technical Review (Optional)</b> <i>For functional app prototypes in Xcode. Judges should be familiar with Swift and iOS development best practices.</i>	Swift code runs in specific examples; code is basic with no abstraction	Code runs without error in all cases; code is basic with some evidence of abstraction	Code is organised with clear Swift naming conventions; high evidence of abstraction; follows iOS guidelines	Code is well documented with comments; effective use of Swift features; employs organisation, such as Model-View-Controller	
<b>Comments:</b>					<b>0</b> <b>Total score</b>





# Swift Coding Club

Swift Playgrounds

## Certificate of Achievement

Awarded to

For



---

Signature

---

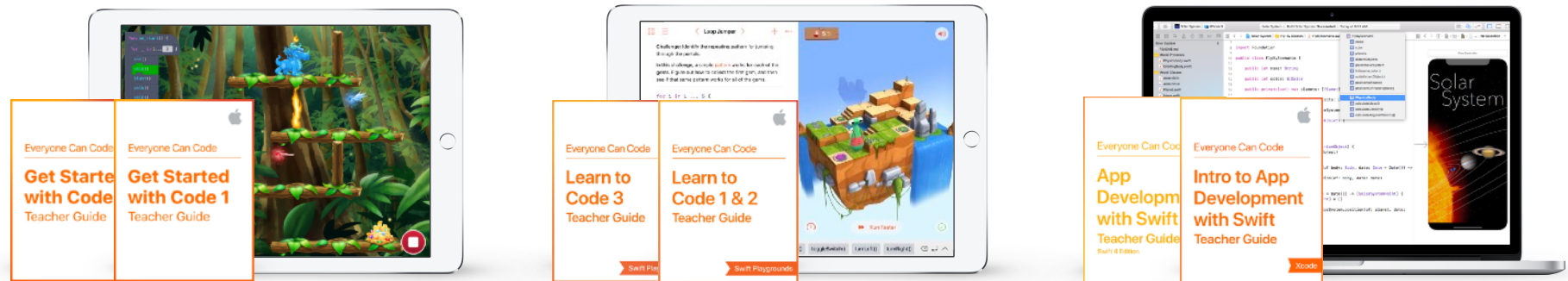
Date

# Take It Further

Swift Coding Club is just the beginning of your coding journey. The Everyone Can Code curriculum provides fun, supportive resources to take coders from learning the basics on iPad to building real apps on Mac.

And you don't have to stop at club activities. Comprehensive Teacher Guides also enable teachers to bring coding into the classroom, with step-by-step, curriculum-aligned lessons for students from primary school to university.

[See all the Everyone Can Code resources >](#)



[Learn more about the  
Get Started with Code  
curriculum >](#)

[Learn more about the  
Swift Playgrounds  
curriculum >](#)

[Learn more about the  
App Development with Swift  
curriculum >](#)

