Develop in Swift Tutorials | *Educator Guide*

Develop in Swift Tutorials are a great first step toward a career in app development using Xcode, Swift, and SwiftUI. They're designed for aspiring coders in high school, higher education, and beyond.

In each chapter, learners will complete:

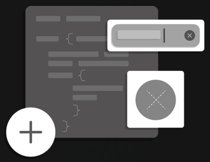
A tutorial

- Coding a project, ranging from an app prototype to a fully functioning app
- Building on prior knowledge, getting progressively more challenging

A wrap-up

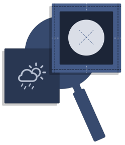
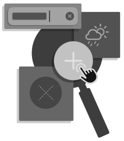
- Review of concepts
- Ideas for extending an app
- Suggestions for how to apply skills in a different context, often by creating a new project

Choose your approach: *You can present the content linearly, or you can incorporate other text, documentation, tutorials, videos, and projects to fit your needs. One option is to have learners complete the tutorial independently, then choose items from the "Continue practicing" section to complete together, allowing learners to work collaboratively and ask questions.*



SwiftUI foundations

Get familiar with the tools and technologies you'll use to create apps.

	Chapter	Description	Topics and skills covered			Estimated time*
	Explore Xcode	Get to know Xcode and SwiftUI by creating a prototype of a messaging app. Learn about syntax for Swift and how to use the source editor and preview.	• Background • **Color** • Creating a new project • Dot notation	• Modifiers • Padding • **String** • Swift syntax • **Text**	• Views • Xcode error messages • Xcode Library	1 hr
	Views, structures, and properties	Learn how to build a custom view to create a multiday weather forecast. In your view, you'll use properties to customize the display for each day.	• Arguments and parameters • **Bool** • Computed properties • Custom subviews • **Font**	• Foreground style • **Image** • Initializers • **Int** • **HStack** and **VStack** • Returning a value	• SF Symbols • Stored properties • String interpolation • Structures • Subviews • Type annotation	1.5 hrs
	**Layout and style**	Build two onboarding screens for an iOS app to learn useful tools for putting views where you want them onscreen and inspecting their size. Define new colors in the asset catalog and use them to create gradient backgrounds.	• Accent color • Arrays • Borders • Brightness • Color assets • Customizing a preview	• **Font** • Frames • **Gradient** • **Image** • Pinning a preview • **Shape** • **Spacer**	• **TabView** • Transparency • Type inference • **ZStack**	1.5 hrs
	Buttons and state	Explore adding buttons to your apps. Learn about Swift closures and their relationship to buttons. Use state properties to update the user interface automatically.	• Animation • Aspect ratio • Assignment operator • **Button** • Button styles • Closures • **Color**	• Disabling controls • Dynamic sizing • Equality operator • **ForEach** • Hierarchical SF Symbols • Randomization	• Range operator • Resizable images • **@State** • Trailing closure syntax • View tint	1.5 hrs
	**Lists and text fields**	Create a dynamic interface that stores a set of items in an array and displays them using lists. Use text fields and bindings to let people enter text.	• Arrays • Adding and removing from arrays • Bindings • Buttons with custom labels	• Disabling autocorrection • Clip shapes • **ForEach** • **List** • Not (**!**) operator	• Symbol rendering modes • Ternary conditional operator • **TextField** • **Toggle**	1.5 hrs

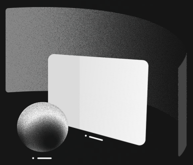
Estimated times are for the tutorial only.

Data modeling

Use SwiftData to store and load information for your app.

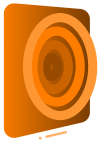
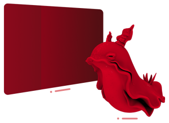
Chapter	*Description*	*Topics and skills covered*			*Estimated time**
Models and persistence	Build a list of your friends' birthdays, using SwiftData to save and retrieve that data across launches.	• **`Calendar`** • Classes • Data models • **`Date`** • **`DatePicker`**	• Date formatting • **`@Environment`** • Frameworks • **`NavigationStack`**	• Safe area • **`@Model`** macro • SwiftData context • **`@Query`** macro	1.5 hrs
Data editing and navigation	Create an app to track someone's favorite movies using SwiftData to manage the model objects. Use a query to display the movies in a list, and make a detail view to edit them.	• **`@Bindable`** • **`ContentUnavailableView`** • Creating sample data • Environment **`dismiss`** value • **`Form`**	• **`Group`** • Modal interfaces • Multiple previews • **`ModelConfiguration`** • **`ModelContainer`** • Navigation hierarchies	• **`NavigationLink`** • **`NavigationSplitView`** • Refactoring • **`Schema`** • Sheets • Toolbars	2 hrs
Relationships and queries	Learn how to create and display relationships between data. Then explore more advanced features of SwiftData queries.	• Custom view initializers • Or (**||**) operator • **`Picker`** • Predicate	• Property wrappers • Model relationships • Search • **`Section`**	• Sorting arrays • View tags	2.5 hrs

**Estimated times are for the tutorial only.*



Spatial computing

Design app experiences for spatial computing.

	Chapter	*Description*	*Topics and skills covered*			*Estimated time**
	**Windows in visionOS**	Create your first visionOS app with a window using SwiftUI.	• `Circle` • `ColorPicker` • `Double` • `Grid`	• `GridRow` • Padding for 3D views • Remainder (%) operator • `Slider`	• visionOS simulator • Window resizability • Windows	1 hr
	**Volumes in visionOS**	View 3D content from any angle in the Shared Space using Reality Composer Pro and SwiftUI.	• Arrays • `DragGesture` • Environment `openWindow` value	• `Model3D` • `NavigationSplitView` • Reality Composer Pro • Rotation in three dimensions	• Toolbars • Volumes • `WindowGroup`	1.5 hrs

Estimated times are for the tutorial only.