Framework

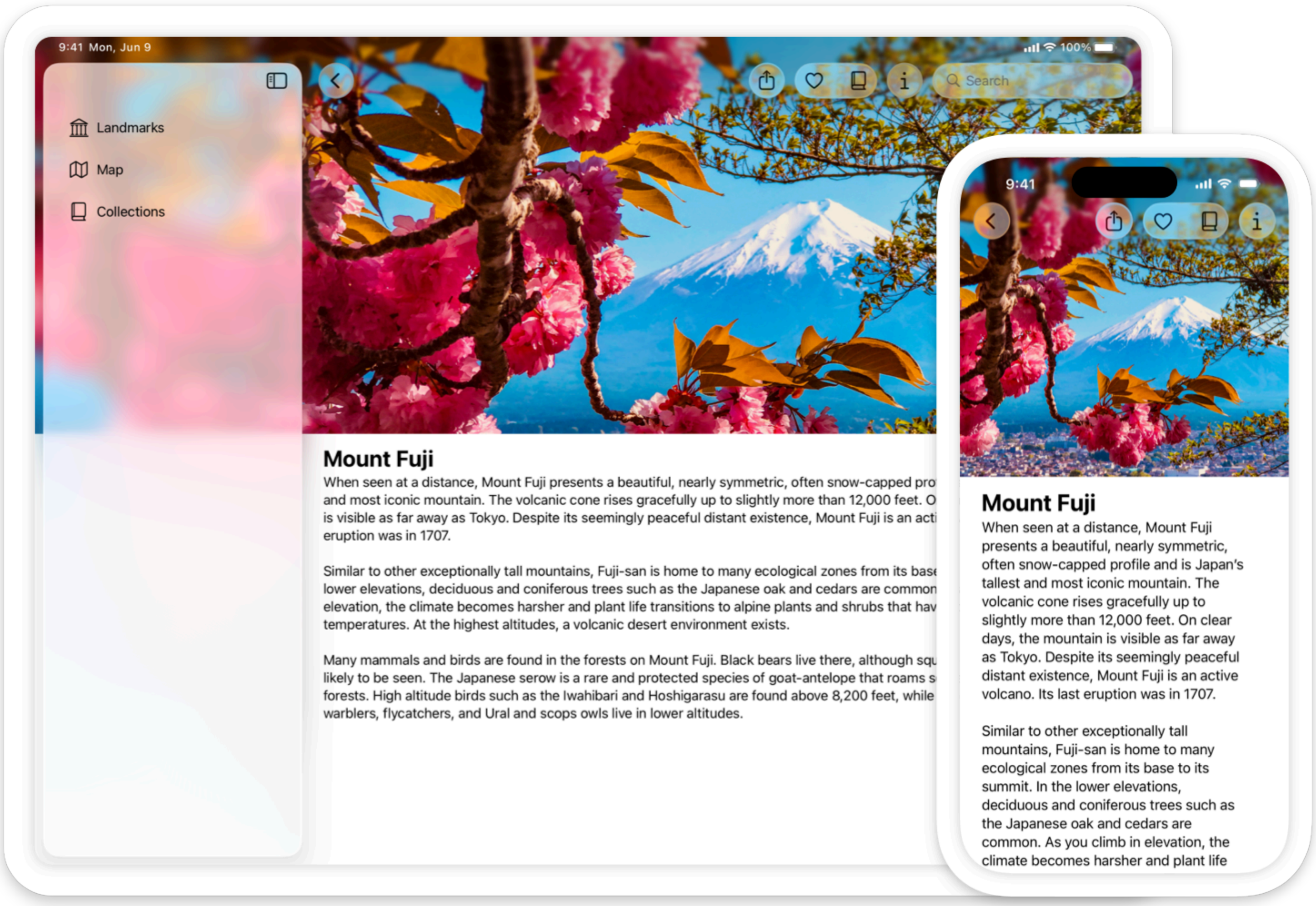
UIKit

Construct and manage a graphical, event-driven user interface for your iOS, iPadOS, or tvOS app.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.0+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

Overview

UIKit provides a variety of features for building apps, including components you can use to construct the core infrastructure of your iOS, iPadOS, or tvOS apps. The framework provides the window and view architecture for implementing your UI, the event-handling infrastructure for delivering Multi-Touch and other types of input to your app, and the main run loop for managing interactions between the user, the system, and your app.



Mount Fuji

When seen at a distance, Mount Fuji presents a beautiful, nearly symmetric, often snow-capped pro
and most iconic mountain. The volcanic cone rises gracefully up to slightly more than 12,000 feet. O
is visible as far away as Tokyo. Despite its seemingly peaceful distant existence, Mount Fuji is an acti
eruption was in 1707.

Similar to other exceptionally tall mountains, Fuji-san is home to many ecological zones from its base
lower elevations, deciduous and coniferous trees such as the Japanese oak and cedars are common
elevation, the climate becomes harsher and plant life transitions to alpine plants and shrubs that hav
temperatures. At the highest altitudes, a volcanic desert environment exists.

Many mammals and birds are found in the forests on Mount Fuji. Black bears live there, although squ
likely to be seen. The Japanese serow is a rare and protected species of goat-antelope that roams s
forests. High altitude birds such as the Iwahibari and Hoshigarasu are found above 8,200 feet, while
warblers, flycatchers, and Ural and scops owls live in lower altitudes.

Mount Fuji

When seen at a distance, Mount Fuji presents a beautiful, nearly symmetric, often snow-capped profile and is Japan's tallest and most iconic mountain. The volcanic cone rises gracefully up to slightly more than 12,000 feet. On clear days, the mountain is visible as far away as Tokyo. Despite its seemingly peaceful distant existence, Mount Fuji is an active volcano. Its last eruption was in 1707.

Similar to other exceptionally tall mountains, Fuji-san is home to many ecological zones from its base to its summit. In the lower elevations, deciduous and coniferous trees such as the Japanese oak and cedars are common. As you climb in elevation, the climate becomes harsher and plant life

UIKit also includes support for animations, documents, drawing and printing, text management and display, search, app extensions, resource management, and getting information about the current device. You can also customize accessibility support, and localize your app's interface for different languages, countries, or cultural regions.

UIKit works seamlessly with the SwiftUI framework, so you can implement parts of your UIKit app in SwiftUI or mix interface elements between the two frameworks. For example, you can place UIKit views and view controllers inside SwiftUI views, and vice versa.

To build a macOS app, you can use SwiftUI to create an app that works across all of Apple's platforms, or use AppKit to create an app for Mac only. Alternatively, you can bring your UIKit iPad app to the Mac with Mac Catalyst.

Important

Use UIKit classes only from your app's main thread or main dispatch queue, unless otherwise indicated in the documentation for those classes. This restriction particularly applies to classes that derive from `UIResponder` or that involve manipulating your app's user interface in any way.

Topics

Essentials

Adopting Liquid Glass

Find out how to bring the new material to your app.

UIKit updates

Learn about important changes to UIKit.

About App Development with UIKit

Learn about the basic support that UIKit and Xcode provide for your iOS and tvOS apps.

Protecting the User’s Privacy

Secure personal data, and respect user preferences for how data is used.

App structure

UIKit manages your app’s interactions with the system and provides classes for you to manage your app’s data and resources.

App and environment

Manage life-cycle events and your app’s UI scenes, and get information about traits and the environment in which your app runs.

Documents, data, and pasteboard

Organize your app’s data and share that data on the pasteboard.

Resource management

Manage the images, strings, storyboards, and nib files that you use to implement your app’s interface.

App extensions

Extend your app’s basic functionality to other parts of the system.

Interprocess communication

Display activity-based services to people.

Mac Catalyst

Create a version of your iPad app that users can run on a Mac device.

User interface

Views help you display content onscreen and facilitate user interactions; view controllers help you manage views and the structure of your interface.

- Views and controls

  Present your content onscreen and define the interactions allowed with that content.

- View controllers

  Manage your interface using view controllers and facilitate navigation around your app’s content.

- View layout

  Use stack views to lay out the views of your interface automatically. Use Auto Layout when you require precise placement of your views.

- Appearance customization

  Apply Liquid Glass to views, support Dark Mode in your app, customize the appearance of bars, and use appearance proxies to modify your UI.

- Animation and haptics

  Provide feedback to users using view-based animations and haptics.

- Windows and screens

  Provide a container for your view hierarchies and other content.

User interactions

Responders and gesture recognizers help you handle touches and other events. Drag and drop, focus, peek and pop, and accessibility handle other user interactions.

- Touches, presses, and gestures

  Encapsulate your app’s event-handling logic in gesture recognizers so that you can reuse that code throughout your app.

- Menus and shortcuts

  Simplify interactions with your app using menu systems, contextual menus, Home Screen quick actions, and keyboard shortcuts.

- Drag and drop

  Bring drag and drop to your app by using interaction APIs with your views.

Pointer interactions

Support pointer interactions in your custom controls and views.

Apple Pencil interactions

Handle user interactions like double tap and squeeze on Apple Pencil.

Focus-based navigation

Navigate the interface of your UIKit app using a remote, game controller, or keyboard.

Accessibility for UIKit

Make your UIKit apps accessible to everyone who uses iOS and tvOS.

Graphics, drawing, and printing

UIKit provides classes and protocols that help you configure your drawing environment and render your content.

Images and PDF

Create and manage images, including those that use bitmap and PDF formats.

Drawing

Configure your app's drawing environment using colors, renderers, draw paths, strings, and shadows.

Printing

Display the system print panels and manage the printing process.

Text

In addition to text views that simplify displaying text in your app, UIKit provides custom text management and rendering that supports the system keyboards.

Text display and fonts

Display text, manage fonts, and check spelling.

TextKit

Manage text storage and perform custom layout of text-based content in your app's views.

Keyboards and input

Configure the system keyboard, create your own keyboards to handle input, or detect key presses on a physical keyboard.

- Writing Tools

  Add support for Writing Tools to your app’s text views.

- Handwriting recognition

  Configure text fields and custom views that accept text to handle input from Apple Pencil.

Deprecated

Avoid using deprecated classes and protocols in your apps.

- Deprecated symbols

  Review unsupported symbols and their replacements.

Reference

- UIKit Enumerations

- UIKit Constants

  This document describes constants that are used throughout the UIKit framework.

- UIKit Data Types

  The UIKit framework defines data types that are used in multiple places throughout the framework.

- UIKit Functions

  The UIKit framework defines a number of functions, many of them used in graphics and drawing operations.

Classes

`class UIColorEffect`

A visual effect that applies a solid color background.