UIPageViewController

A container view controller that manages navigation between pages of content, where each page is managed by a child view controller.

Declaration

```
class UIPageViewController : UIViewController
```

Overview

Page view controller–navigation can be controlled programmatically by your app or directly by the user using gestures. When navigating from page to page, the page view controller uses the transition that you specify to animate the change.

> **Important**
>
> In tvOS, the `UIPageViewController` class provides only a way to swipe between full-screen content pages. Unlike in iOS, a user cannot interact with or move focus between items on each page.

When defining a page view controller interface, you can provide the content view controllers one at a time (or two at a time, depending upon the spine position and double-sided state) or as-needed using a data source. When providing content view controllers one at a time, you use the `setViewControllers(_:direction:animated:completion:)` method to set the current content view controllers. To support gesture-based navigation, you must provide your view controllers using a data source object.

The data source for a page view controller is responsible for providing the content view controllers on demand and must conform to the `UIPageViewControllerDataSource`protocol. The delegate object—an object that conforms to the `UIPageViewControllerDelegate` protocol—provides some appearance-related information and receives notifications about gesture-initiated transitions.

This class is generally used as-is, but can also be subclassed.

Topics

Initializing a Page View Controller

```
init(transitionStyle: UIPageViewController.TransitionStyle, navigationOrientation: UIPageViewController.NavigationOrientation, options: [UIPageViewController.OptionsKey : Any]? = nil)
```

Initializes a newly created page view controller.

```
init?(coder: NSCoder)
```

```
struct UIPageViewController.OptionsKey
```

Keys for creating the page view controller.

Providing the Page Content

```
var dataSource: UIPageViewControllerDataSource?
```

The object that provides view controllers.

```
protocol UIPageViewControllerDataSource
```

The `UIPageViewControllerDataSource` protocol is adopted by an object that provides view controllers to the page view controller on an as-needed basis, in response to navigation gestures.

Customizing the Page View Behavior

```
var delegate: UIPageViewControllerDelegate?
```

The delegate object.

```
protocol UIPageViewControllerDelegate
```

The delegate of a page view controller must adopt the `UIPageViewControllerDelegate` protocol. These methods allow the delegate to receive a notification when the device orientation changes and when the user navigates to a new page. For page-curl style transitions, the delegate can provide a different spine location in response to a change in the interface orientation.

Providing Content

```
func setViewControllers([UIViewController]?, direction: UIPageViewController.NavigationDirection, animated: Bool, completion: ((Bool) -> Void)? = nil)
```

Sets the view controllers to be displayed.

```
enum UIPageViewController.NavigationDirection
```

Directions for page-turn transitions.

```
var viewControllers: [UIViewController]?
```

The view controllers displayed by the page view controller.

```
var gestureRecognizers: [UIGestureRecognizer]
```

An array of `UIGestureRecognizer` objects that are configured to handle user interaction.

Display Options

```
var navigationOrientation: UIPageViewController.NavigationOrientation
```

The direction along which navigation occurs.

```
enum UIPageViewController.NavigationOrientation
```

Orientations for page-turn transitions.

```
var spineLocation: UIPageViewController.SpineLocation
```

The location of the spine.

```
enum UIPageViewController.SpineLocation
```

Locations for the spine.

```
var transitionStyle: UIPageViewController.TransitionStyle
```

The style used to transition between view controllers.

```
enum UIPageViewController.TransitionStyle
```

Styles for the page-turn transition.

```
var isDoubleSided: Bool
```

A Boolean value that indicates whether content appears on the back of pages.