# Basic Exercises Part 3.6. UIPageViewController

## UIPageViewController

* The UIPageViewController class is a container view controller that manages navigation between pages of content, where each page is managed by a child view controller.
* 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.
* The screen navigation is controlled by the user gestures.
* Once UIPageViewController interface is defined, you can add the ViewController for page content.

### **1.1 Apple documentation**

For information about basic behaviors, visit:

https://developer.apple.com/documentation/uikit/uipageviewcontroller

### **1.2 Create a new project**

Create a basic Single View.

### **1.3 Setup the PageViewController**

Select the default UIViewController.swift file and inherit UIPageViewController instead of UIViewController.

**class** ViewController: UIPageViewController, UIPageViewControlleDataSource{

Create a new Swift file named PageContentViewController for PageViewController content by selecting file menu and inherit UIViewController class.

### **1.4 Assign class references**

Open Main.storyboard and select UIViewController and assign PageViewController class reference to it. Remember to set the storyboard ID.

### **1.5 Add UI elements to PageContentViewController**

Drag an ImageView from library to PageContentViewController and apply constraints to image view. Drag a Label to the same view controller.

### **1.6 Add PageViewController**

Drag new PageViewController from object library which is subclass of UIPageViewController and assign ViewController class reference to it.

### **1.6 PageViewController setting**

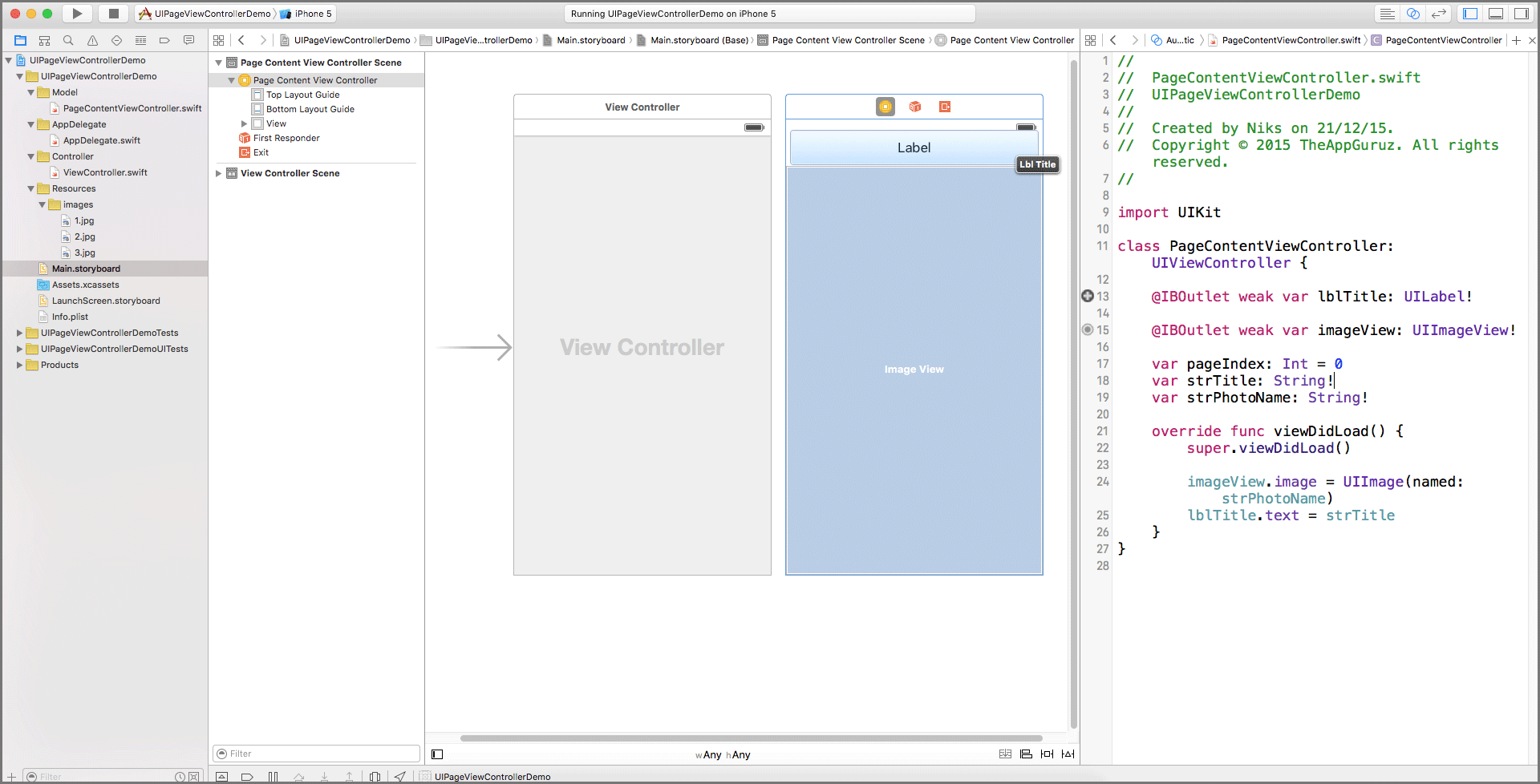
By default, the transition style of the page view controller is set as Page Curl. The Page Curl style is perfect for book kind of apps. For navigating between screens, using scrolling style. So change the transition style to Scroll under Attribute Inspector.

Now, assign Storyboard ID for PageContentViewController which we will use later in our code.

* Select the PageContentViewController
* Go to identity inspector.
* Enter the text PageContentViewController.

### **1.7 SetUp PageContentViewController**

Now create an Outlet from the ImageView and Label in the PageContentViewController.



Create variables for content.

var pageIndex: Int = 0

var strTitle: String!

Var strPhotoName: String!

Set data to object in viewDidLoad() method.

override func viewDidLoad() {

super.viewDidLoad()

imageView.image = UIImage(named: strPhotoName)

lblTitle.text = strTitle

}

Create two NSArray to store Image title and Image name

var arrPageTitle: NSArray = NSArray()

var arrPagePhoro: NSArray = NSArray()

### **1.8 Initializations and implementations**

Back to ViewController. Set the datasource for PageViewController.

Initialize the arrays.

Set first page of PageViewController from viewDidLoad() method.

override func viewDidLoad() {

super.viewDidLoad()

arrPageTitle = ["This is The App", "This is a View", "This is a second view"];

arrPagePhoto = ["1.jpg", "2.jpg", "3.jpg"];

self.dataSource = self

self.setViewControllers([getViewControllerAtIndex(0)] as [UIViewController], direction: UIPageViewControllerNavigationDirection.Forward, animated: false, completion: nil)

}

Use this method viewControllerBefore ViewController: to return to previous view.

func pageViewController(\_ pageViewController: UIPageViewController, viewControllerBefore viewController: UIViewController) -> UIViewController?

{

let pageContent: PageContentViewController = viewController as! PageContentViewController

var index = pageContent.pageIndex

if ((index == 0) || (index == NSNotFound))

{

return nil

}

index-=1;

return getViewControllerAtIndex(index)

}

Use this method viewControllerAfter viewController: to return to return next view

func pageViewController(pageViewController: UIPageViewController, viewControllerAfter viewController: UIViewController) -> UIViewController?

{

let pageContent: PageContentViewController = viewController as! PageContentViewController

var index = pageContent.pageIndex

if (index == NSNotFound)

{

return nil;

}

index+=1;

if (index == arrPageTitle.count)

{

return nil;

}

return getViewControllerAtIndex(index)

}

### **1.9 User interactions**

Open the Library

When user swipe using gesture then we have to instantiate ViewController from storyboard and assign all content data of ViewController and return it.

func getViewControllerAtIndex(index: NSInteger) -> PageContentViewController

{

// Create a new view controller and pass suitable data.

let pageContentViewController = self.storyboard?.instantiateViewController(withIdentifier: "PageContentViewController") as! PageContentViewController

pageContentViewController.strTitle = "\(arrPageTitle[index])"

pageContentViewController.strPhotoName = "\(arrPagePhoto[index])"

pageContentViewController.pageIndex = index

return pageContentViewController

}

Open and review the code attached.

### **1.8 SetUp same project**

Now you can change the pages on the Swift app. As requirement, create the same project using Objective C.