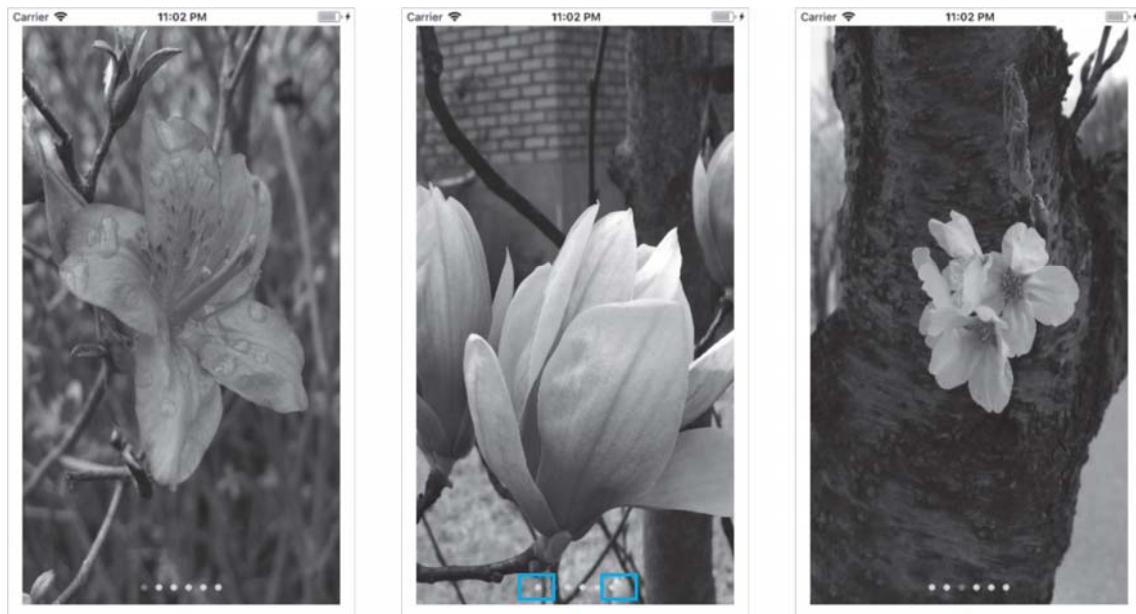


09 장. 페이지 이동하기 - 페이지 컨트롤

페이지 컨트롤(Page Control)은 여러 개의 내용을 페이지별로 보여주고자 할 때 사용하는 객체이다.

이 장에서는 페이지 컨트롤을 이용해 여러개의 이미지를 화면에 출력하는 간단한 캘러리 앱을 만들어 보자.



프로그램을 실행하면 첫 번째 이미지가 보이고 이미지 아래쪽에 페이지 컨트롤이 표시됩니다. 페이지 컨트롤의 오른쪽 부분을 터치하면 다음 페이지로 이동하고, 페이지 컨트롤의 왼쪽 부분을 터치하면 이전 페이지로 이동하여 해당 이미지를 보여 줍니다.

09-1 페이지 컨트롤이란?

페이지 컨트롤(Page Control)은 오른쪽 그림에서 작은 동그라미가 줄지어 있는 부분으로, 여러 페이지 중에서 현재 페이지를 알려주는 역할을 한다.

다시 말해 현재 뷰에서 보여주고자 하는 내용이 여러개일 때 페이지를 나눠서 보여 주는 기능을 말한다. 흔히 갤러리나 아이폰의 홈(Home)화면과 같은 앱에서 전체 페이지 중 현재 페이지가 어느 부분에 있는지 알려줄 때 사용한다.



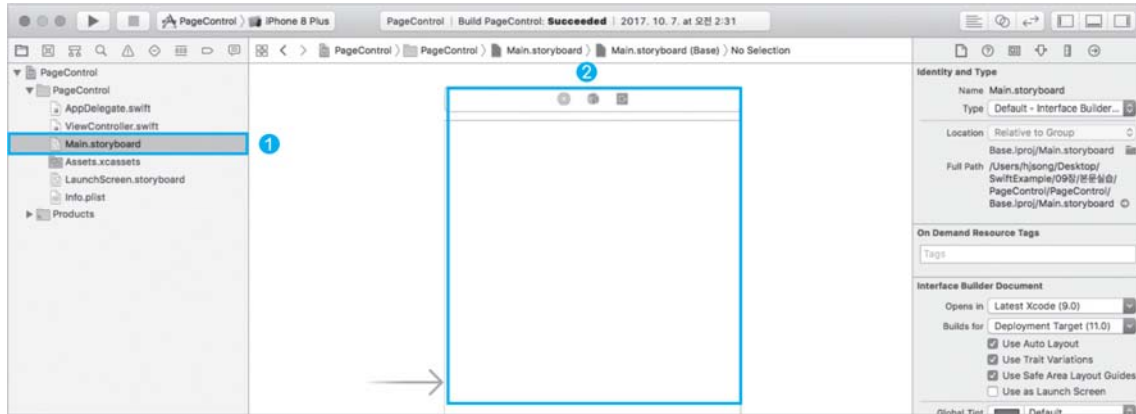
아이폰의 홈(Home) 화면

09-2 페이지 컨트롤 앱을 위한 기본 환경 구성하기

1. 새 프로젝트 만들기

Xcode를 실행한 후 'PageControl'이라는 이름으로 새 프로젝트를 만든다.

뷰 컨트롤러의 크기를 상황에 맞게 조절한다.

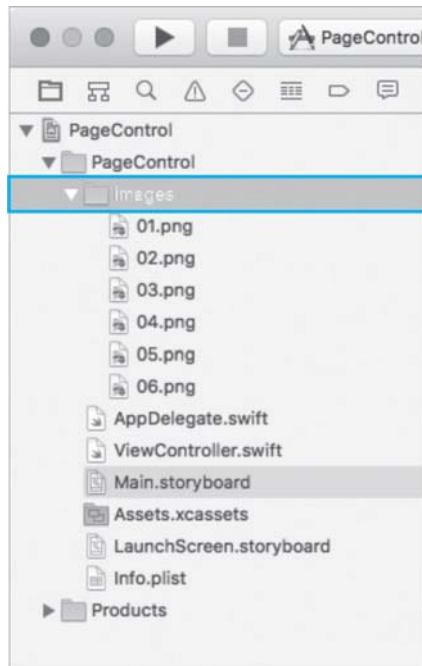


2. 프로젝트에 이미지 파일 추가하기

앱에서 사용할 이미지를 프로젝트에 추가해 보겠다.

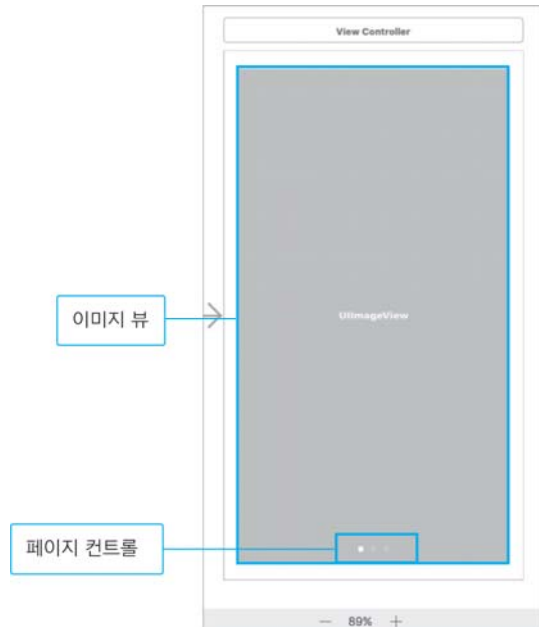
우선 내비게이터 영역의 [PageControl]폴더를 마우스 오른쪽 버튼으로 클릭하여 [새폴더 (New Group)]를 만든다, 폴더 이름은 [images]로 한다.

4. 앱에서 사용할 이미지 파일을 프로젝트에 추가한다. 추가한 이미지를 모두 선택한 후 오른쪽 인스펙트 영역에서 [Target Membership]의 [PageControl]에 체크한다



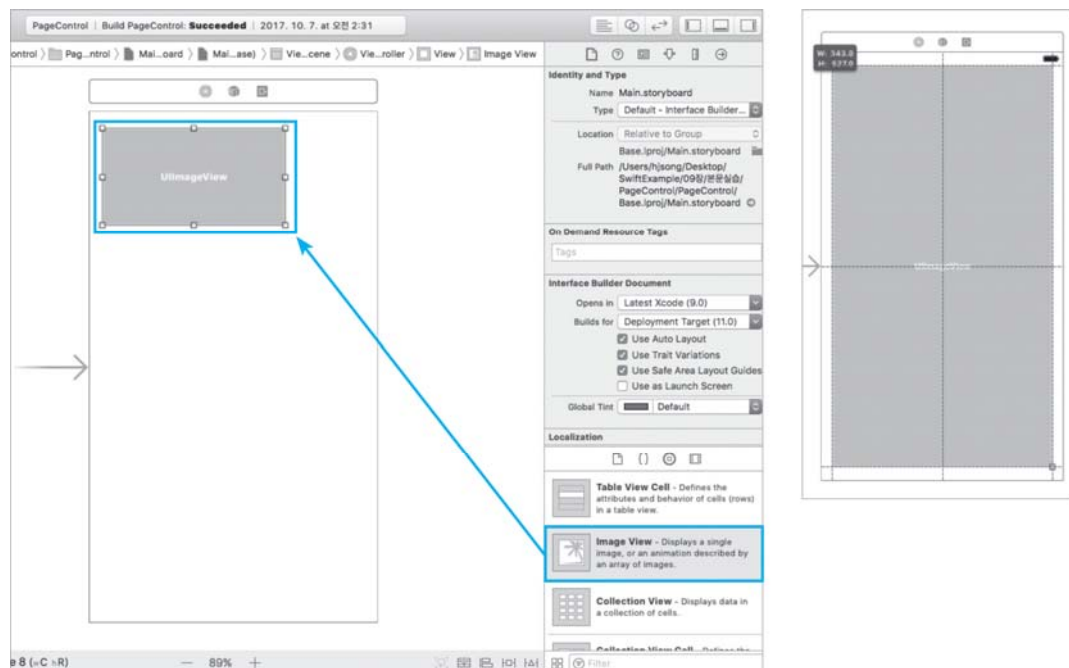
09-3 스토리보드로 페이지 컨트롤 앱 화면 꾸미기

이제 스토리보드를 사용하여 앱 화면을 꾸며 보겠습니다. 페이지 컨트롤 기능을 보여 주려면 페이지 컨트롤(Page Control) 객체와 각 페이지별 이미지를 보여 줄 이미지를 보여 줄 이미지 뷰(Image View)객체가 필요하다.



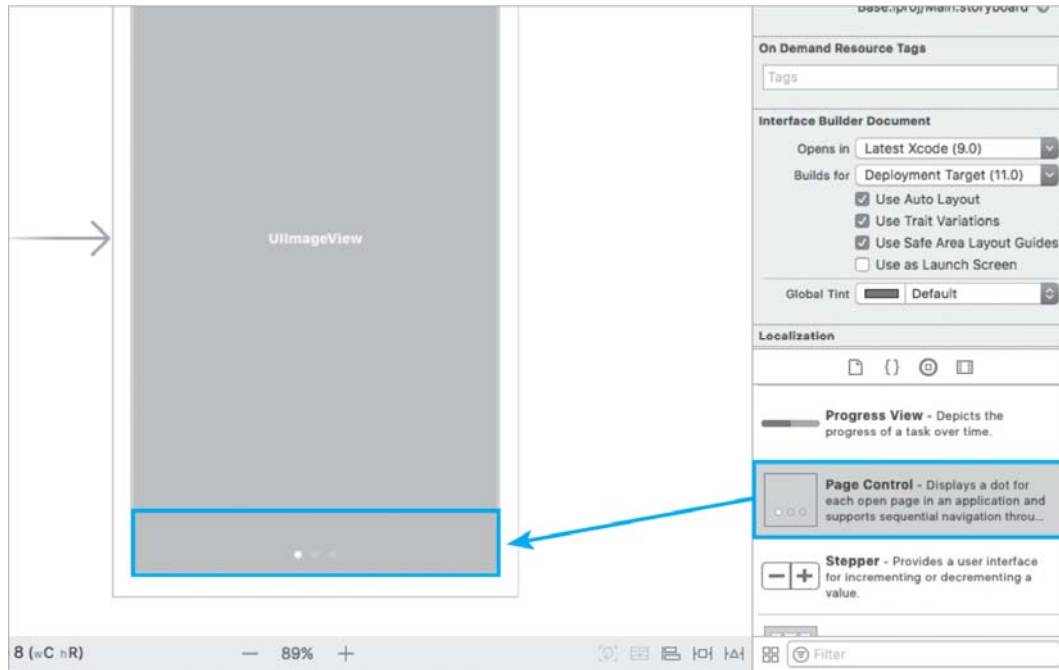
1. 이미지 뷰 추가하기

먼저 오른쪽 아랫부분의 오브젝트 라이브러리에서 [이미지 뷰(Image View)]를 찾아 스토리보드로 끌어와 화면의 위쪽에 배치한다, 그리고 이미지 뷰의 크기를 가득 차게 조절한다.



2. 페이지 컨트롤 추가하기

[페이지 컨트롤(Page Control)]도 마찬가지로 오른쪽 아랫부분의 오브젝트 라이브러리에서 찾아 스토리보드로 끌어와 화면의 아랫쪽에 배치한다. 그리고 좌우로 늘려 크기를 조절한다.



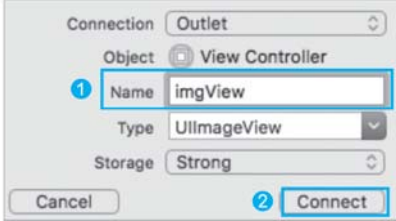
09-4 아웃렛 변수와 액션 함수 추가하기

1. 코드 작성을 위한 액션 함수를 추가하기 위해 오른쪽 윗부분의 [Show the Assistant editor] 버튼을 클릭하여 보조 편집기 영역을 연다.

스토리보드의 이미지 뷰 객체를 마우스 오른쪽 버튼으로 클릭한 후 오른쪽의 보조 편집기 영역으로 드래그하면 다음 그림과 같이 연결선이 나타난다. 드래그한 연결선을 뷰 컨트롤러의 클래스 선언문 바로 아래에 배치한다,

연결 설정 창이 나타나면 아웃렛 변수의 이름(Name)에 'imageView'를 입력하고 [Connect] 버튼을 클릭하여 이미지 뷰와 아웃렛 변수를 연결한다.

위치	뷰 컨트롤러의 클래스 선언문 바로 아래
연결(Connection)	Outlet
이름(Name)	imageView
유형(Type)	UIImageView



```

4 //
5 // Created by Ho-Jeong Song on 2017. 10. 2..
6 // Copyright © 2017년 Ho-Jeong Song. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12     @IBOutlet var imageView: UIImageView!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16     }
17 }
  
```

@IBOutlet var imageView: UIImageView!

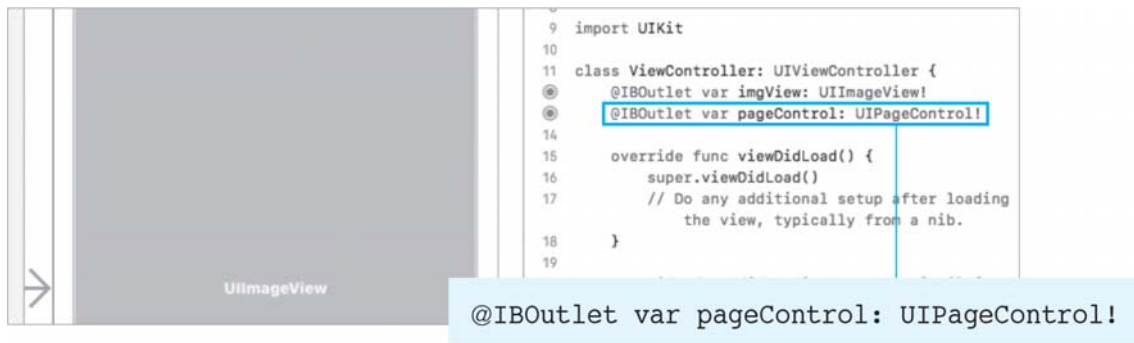
2. 페이지 컨트롤에 대한 아웃렛 변수 추가하기

스토리보드 상의 페이지 컨트롤을 마우스 오른쪽 버튼으로 클릭한 후 드래그해서 imageView 아웃렛 변수 아래쪽에 갖다 놓는다

연결설정창이 나타나면 아웃렛 변수의 이름(Name)에 'pageControl'을 입력하고 [Connect] 버튼을 클릭하여 페이지 컨트롤과 아웃렛 변수를 연결한다,

위치	imageView 아웃렛 변수 아래
연결(Connection)	Outlet
이름(Name)	pageControl
유형(Type)	UIPageControl





3. 페이지 컨트롤에 대한 액션 함수를 추가한다.

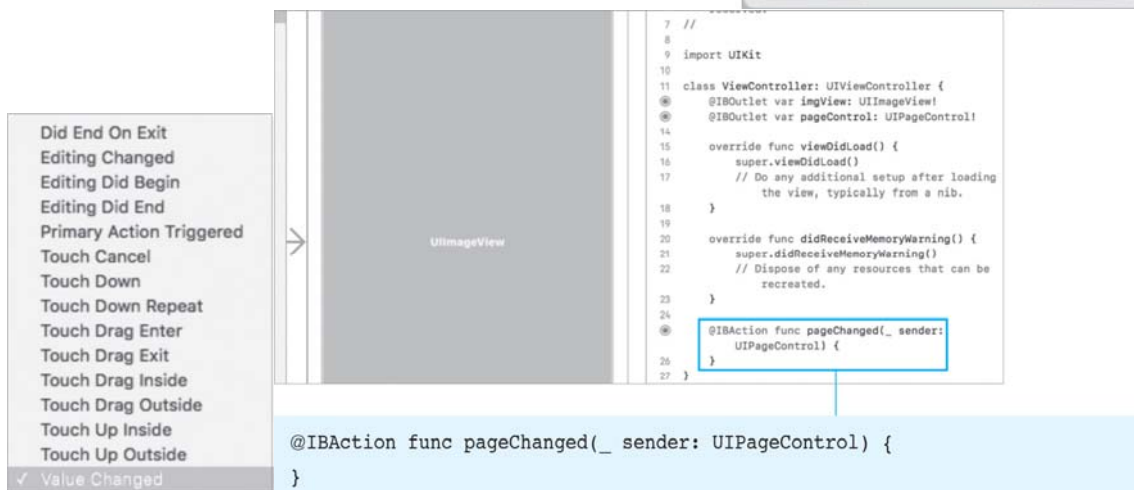
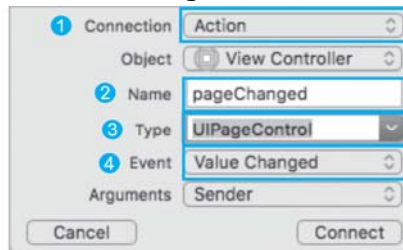
마우스 오른쪽 버튼으로 스토리보드의 페이지 컨트롤 객체를 클릭한 후 드래그해서 오른쪽 소스의 가장 아래쪽에 있는 뷰 컨트롤러 클래스 닫음 괄호 '}'의 바로 위에 갖다 놓는다.

연결 설정 창이 나타나면 이 설정 창에서 다음과 같이 설정한 후 Event 가 [Value Changed]인지 확인하고 [Connect] 버튼을 클릭하여 액션을 추가한다.

'Event' 설정은 액션 함수를 언제 실행할지를 결정하는 중요한 항목이다. 즉 해당 객체에서 어떠한 이벤트가 발생하였을 때 액션 함수를 실행할지 선택하는 것이다.

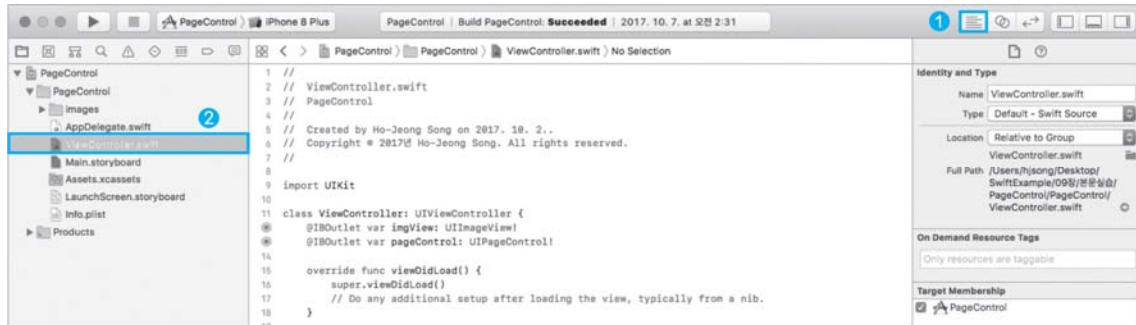
페이지 컨트롤러 값이 변할 때 액션 함수를 실행하기 위해 [value Changed]를 선택한다.

위치	뷰 컨트롤러 클래스 닫음 괄호 '}' 바로 위
연결(Connection)	Action
이름(Name)	pageChanged
유형(Type)	UIPageControl



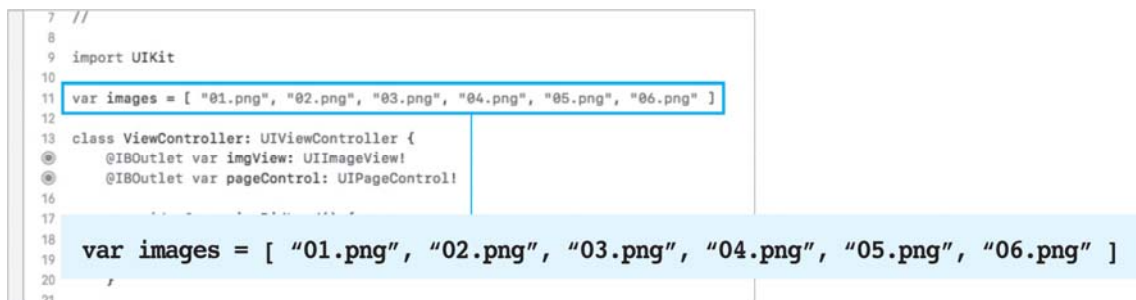
09-5 페이지 컨트롤 동작 구현하기

1. 코딩을 위해 화면 모드를 수정한다, 오른쪽 윗부분에서 [Show the Standard editor] 버튼을 클릭한 후 왼쪽의 내비게이터 영역에서 [ViewController.swift]를 선택한다.



2. 변수 추가하기

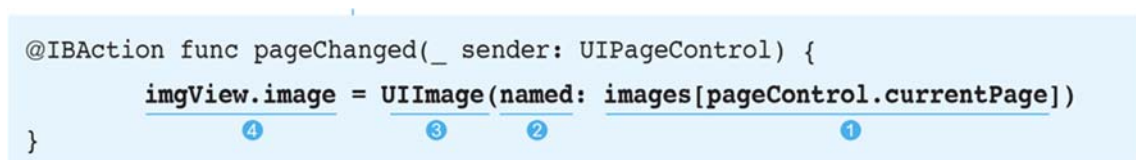
이제 페이지 컨트롤을 작동하는 데 필요한 변수를 뷰 컨트롤러 클래스 선언문의 바로 위에 추가한다.



여기서 변수 images 는 화면에 보여 줄 이미지 파일의 이름을 저장한 배열이다.

3. 페이지가 변경될 때 동작할 함수 코딩하기

앱에서 페이지(Page)가 변경될 때 동작할 pageChanged 함수를 코딩한다. 다음 코드를 pageChanged 함수 안에 입력하려면 페이지가 변경되었을 때 현재 페이지에 해당하는 이미지 파일의 이름을 image 배열에서 가지고 와서 imageView에 할당하여 화면에 이미지를 출력한다.



- 1) 페이지 컨트롤의 현재 페이지를 가지고 온다.
- 2) 현재 페이지에 해당하는 이미지 파일의 이름을 image 배열에서 가지고 온다.
- 3) UIImage 타입의 이미지를 만든다.
- 4) 만든 이미지를 이미지 뷰에 할당한다.

4. 앱을 시작할 때 수행할 내용 코딩하기

처음 앱이 실행되어 뷰가 보일 때 화면에 페이지롤의 위치와 이미지를 나타내기 위한 코드를 작성해 보자, 페이지 컨트롤의 현재 페이지를 출력할 것이다. 페이지 컨트롤의 뷰가 보일 때 자동으로 호출되는 함수는 viewDidLoad 함수이다. 이 함수 안에 뷰를 보여줄 때 실행하고 싶은 코드를 추가하면 된다.

```
17     override func viewDidLoad() {  
18         super.viewDidLoad()  
19         // Do any additional setup after loading the view, typically from a nib.  
20  
21         pageCount.numberOfPages = images.count  
22         pageCount.currentPage = 0  
23  
24         pageCount.pageIndicatorTintColor = UIColor.green  
25         pageCount.currentPageIndicatorTintColor = UIColor.red  
26  
27         imageView.image = UIImage(named: images[0])  
28     }
```

21 행은 numberOfPages : 페이지 컨트롤의 전체 페이지수를 의미한다.

22 행은 currentPage : 현재 페이지를 의미한다,

24 행은 pageIndicatorTintColor : 페이지 컨트롤의 페이지를 표시하는 부분의 색상을 의미한다. UIColor.green 을 입력하면 페이지를 표시하는 페이지 컨트롤의 전체 동그라미 부분이 초록으로 표시된다.

25 행은 currentPageIndicatorTintColor : 페이지 컨트롤의 현재 페이지를 표시하는 색상을 의미한다. UIColor.red 를 입력하면 현재 페이지가 빨간색으로 표시된다.