03 장 원하는 이미지 화면에 출력하기 - 이미지 뷰

이제 그림이나 사진이 없는 앱은 힘들 정도로 이미지의 역할이 커졌다. 만약 앱이 텍스트와 버튼으로만 이루어져 있다면 답답해 보이고 단순해 보일 수 있는데, 앱에 이미지를 하나만 넣어도 분위기는 180 도 달라진다.

이번 장에서는 JPG 나 PNG 등의 이미지를 화면에 출력할 수 있는 '이미지 뷰'의 사용법을 배워 보겠다.

이미지 뷰는 확대나 축소도 자유롭게 할 수 있다.

03-1 이미지 뷰, 어디에 사용할까?

이미지 뷰(Image View)는 이름에서 알 수 있듯이 앱에서 사진을 보여 줘야 할 때 사용하는 객 체입니다. 간단한 갤러리 앱뿐만 아니라 사진을 함께 기록해야 하는 다이어리 앱, 아이폰으로 촬영한 사진을 불러와 편집까지 할 수 있는 사진 편집 앱까지 많은 앱에서 이미지 뷰를 사용 하고 있습니다.

아이폰의 기본 사진첩 앱

데이 원(Day One) 앱

인스타그램(Instagram) 앱

앱에 들어갈 이미지 크기

앱을 만들 때 권장하는 이미지 크기는 지원하고자 하는 기기의 최대 해상도에 맞추면 됩니다. 이미지가 너무 크면 앱의 크기가 너무 커지고 이미지를 불러오는 데 메모리를 많이 차지합니 다. 반면에 이미지가 너무 작으면 이미지를 확대했을 때 픽셀이 깨지는 현상이 발생합니다.

기기	아이폰 4s	아이폰 5, 5s	아이폰 6, 6s, 7, 8	아이폰 6+, 6s+, 7+, 8+	아이폰 X
권장 이미지 해상도	640×960px	640×1136px	750×1334px	1242×2208px	1125×2436px

기기	아이패드 2	아이패드 미니	아이패드 프로
권장 이미지 해상도	768×1024	1536×2048	2048×2732

출처: iOS 개발자 문서(https://goo.gl/p0IMvT)

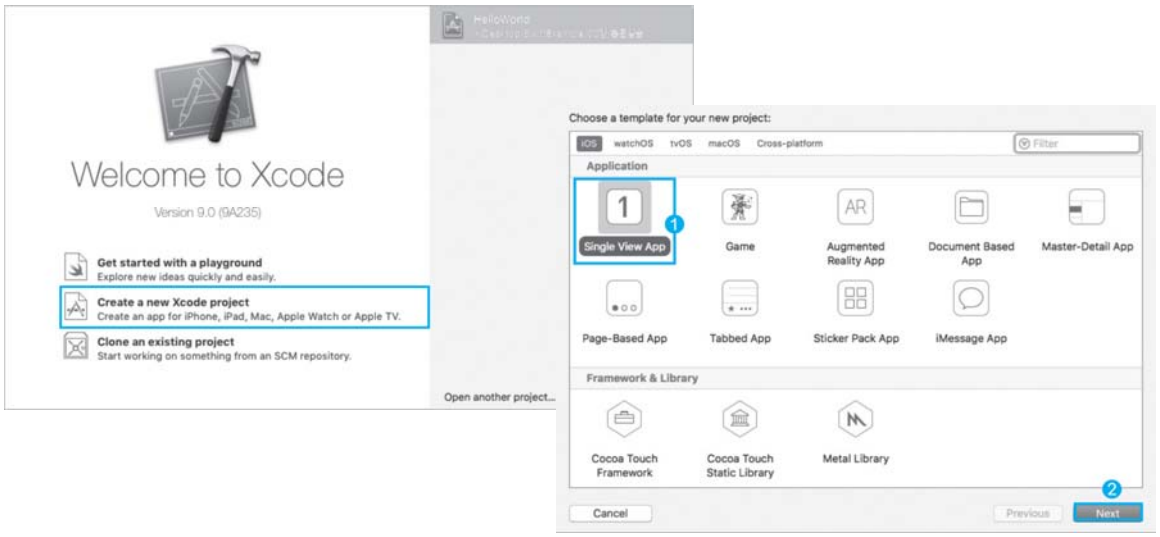
03-2 이미지뷰앱을위한기본환경구성하기

이절에서는 프로젝트에 전구 이미지를 추가하고 화면에 출력하는 방법과 이미지를 확대하고 축소하는 방법을 알아보자. 스토리보드 작업에 앞서 이미지 뷰앱에 필요한 파일을 불러오고 기본 환경을 구성하는 작업을 먼저 해보자.

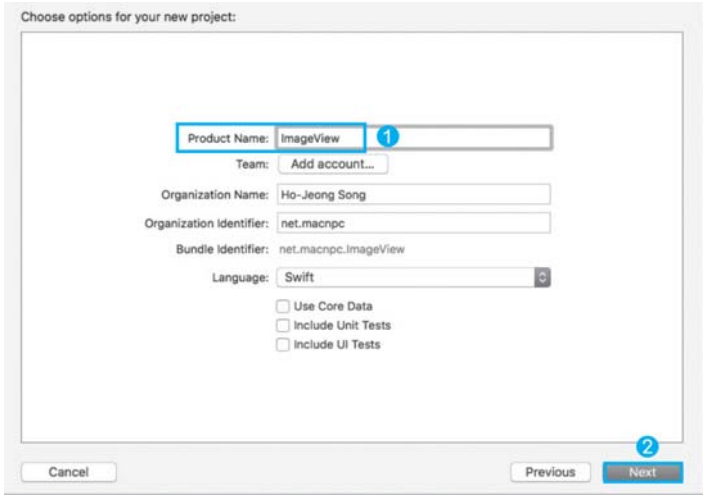
새프로젝트 만들기

1. Xcode 를 실행하면 다음과 같은 시작 화면이 나옵니다.
여기에서 [Create a New Xcode Project]를 클릭하여 새 프로젝트를 만든다.

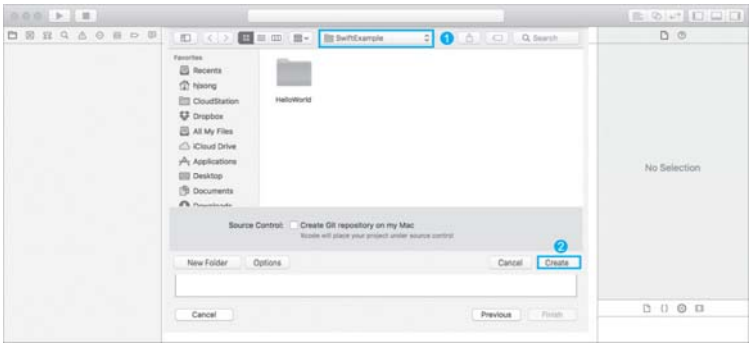
2. 템플릿을 선택하기
새프로젝트를 시작할 때 템플릿을 선택하는 창이 나타난다.
[Single View App]을 선택한 후 [Next]버튼을 클릭한다.



3. 프로젝트의 기본 정보를 입력하는 창이 나오면 프로젝트 이름, 사용하는 언어 지정 등 프로젝트의 기본 정보를 입력한 후 [Next]버튼을 클릭한다.
프로젝트 이름을 "ImageView"로 입력한다.



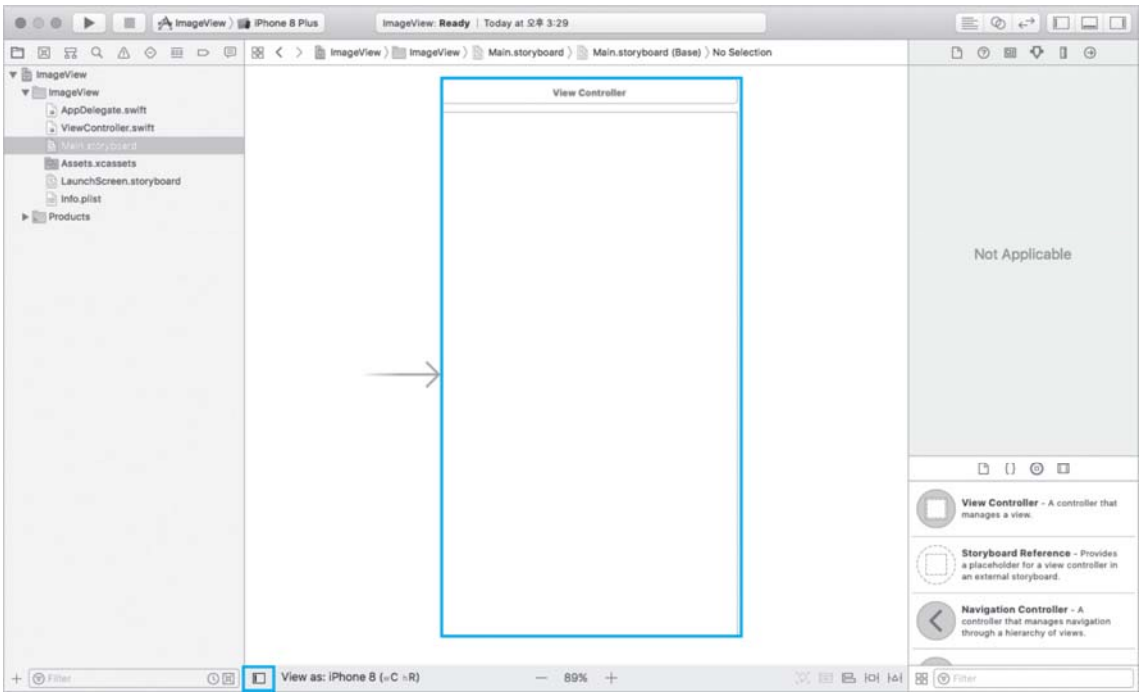
4. 프로젝트를 지정할 폴더를 선택한 후 [Create]버튼을 클릭한다,



화면 창 조절하기

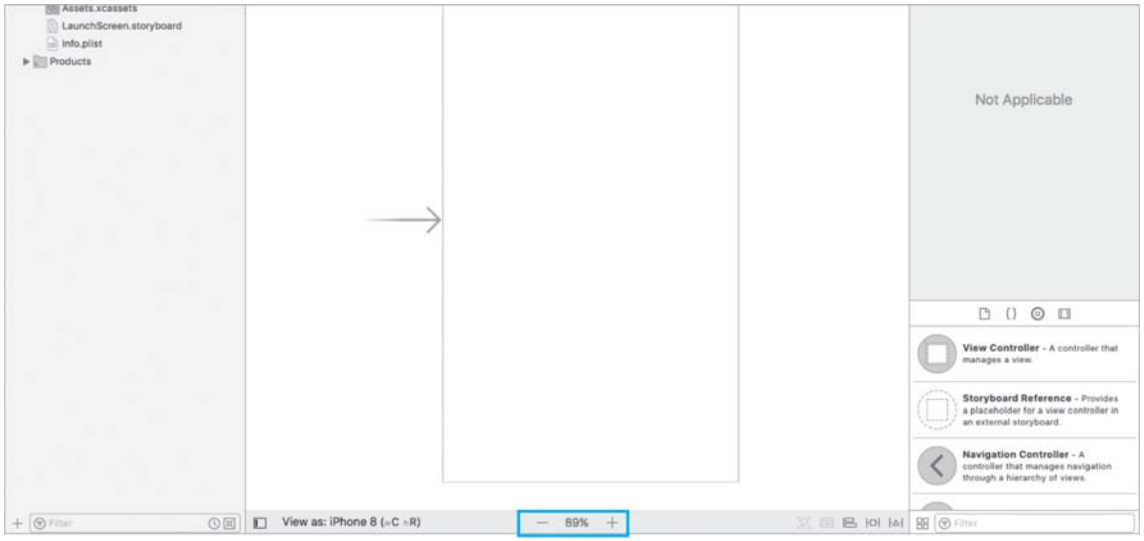
Xcode화면이 처음 열렸을 때 화면 왼쪽의 내비게이터 영역에서 [Mian.storyboard]를 선택하면 스토리보드가 화면에 나타난다.

만약 모니터가 작아 스토리보드의 일부만 보인다면 내비게이터 영역 또는 도큐먼트 아웃라인 영역을 닫아 스토리보드가 잘 보 이게 조절합니다.



뷰 컨트롤러 크기 조절하기

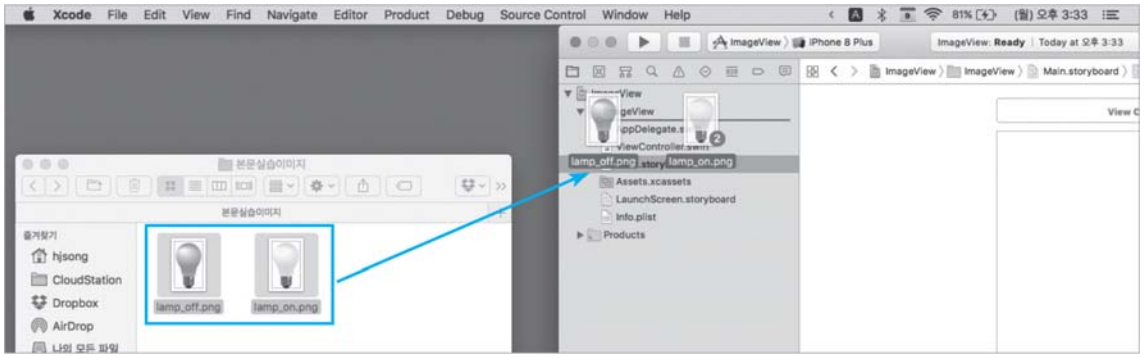
뷰의 크기를 조절하기 위해서는 아랫부분의 [Zoom In],[Zoom Out]버튼을 사용하여 조절하면 된다.



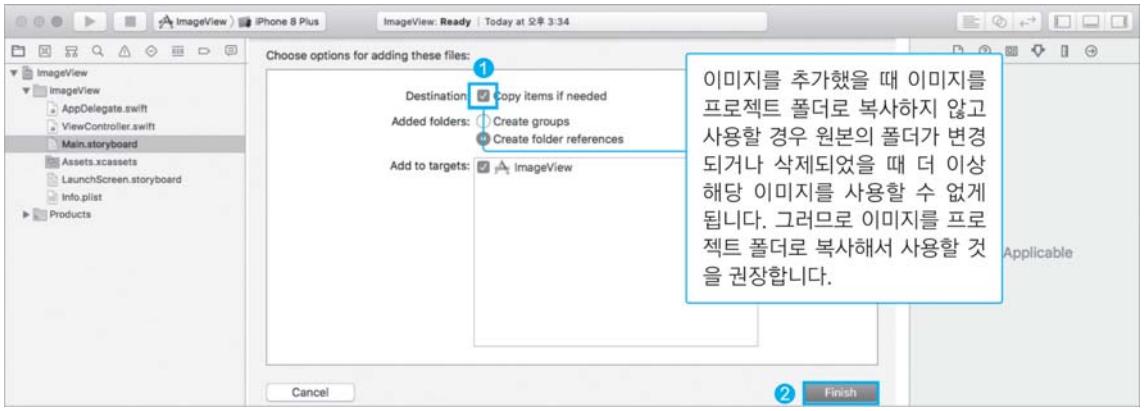
3. 이미지 뷰에 사용할 이미지 추가하기

앱에 사용할 이미지를 프로젝트에 추가하자.

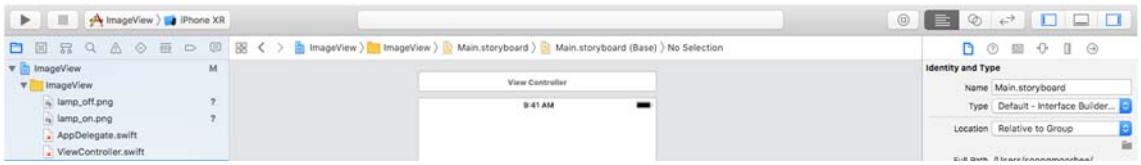
파인더에서 'lamp_on.png'와 'lamp_off.png'이미지를 선택하여 내비게이터 영역의 [ImageView]폴더 아래로 드래그앤 드롭한다.



4. 파일 추가에 대한 설정 창이 나타난다. 현재 추가하려고 하는 이미지를 프로젝트 폴더에 복사해주는 [Destination: Copy items if needed] 항목에 체크가 되어 있는지 확인한 후 [Finish]버튼을 클릭한다.



5. 이미지가 프로젝트에 추가되었다.



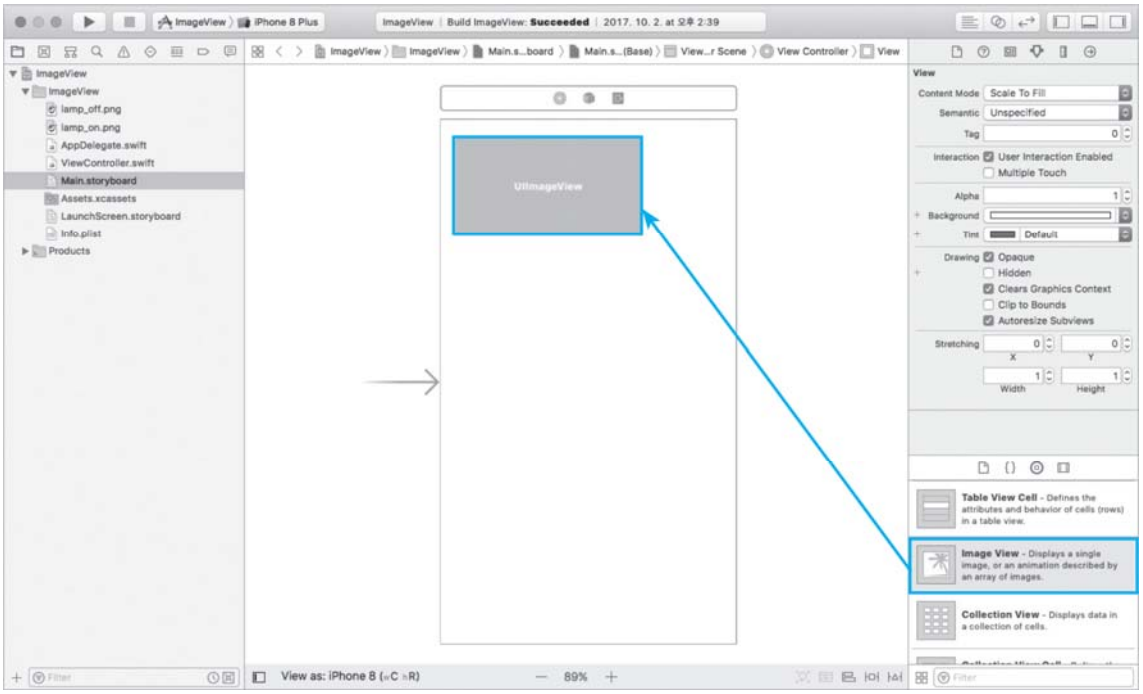
6. 방금 추가한 이미지를 모두 선택한 후 오른쪽 인스팩터 영역에서 [Target Membership]의 [ImageView]에 체크한다.

03-3 스토리보드로 이미지 뷰 앱 화면 꾸미기

스토리보드를 사용해 이미지 뷰 앱의 화면을 꾸며 보자. 이 앱에서는 전구 이미지를 보여 줄 이미지 뷰(Image View)객체, 이미지를 확대 및 축소하기 위한 버튼(Button)객체 그리고 전구를 켜고 끄기 위한 스위치(Switch)객체를 사용한다.

1. 이미지를 보여 줄 이미지 뷰 추가하기

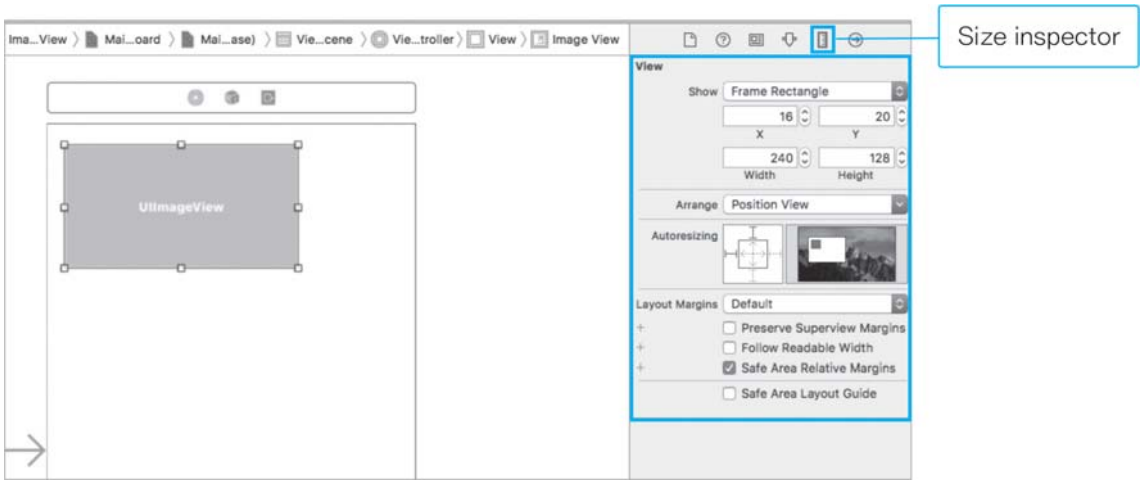
먼저 이미지를 보여 줄 이미지 뷰 객체를 추가한다. Xcode 화면 오른쪽 아랫부분의 오브젝트 라이브러리를 스크롤하여 [이미지 뷰(Image View)]를 찾아 스토리보드로 끌어와 화면의 왼쪽 윗부분에 배치한다.



2. 이제 이미지 뷰의 크기를 조절하자.

화면 아래쪽에는 버튼이 들어갈 공간이 필요하기 때문에 그 공간을 제외하고 확대되기 전의 크기에 맞춰 이미지를 조절한다.

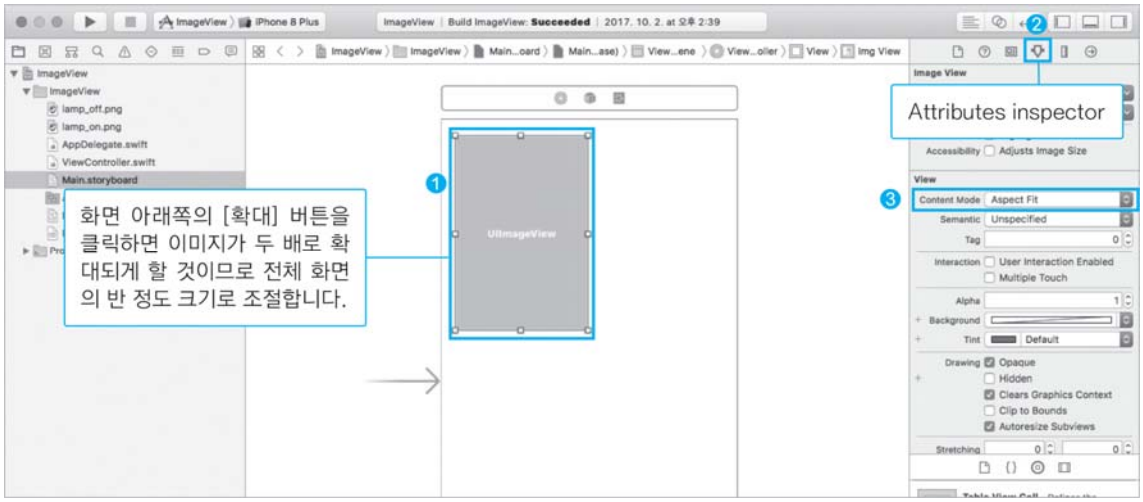
오른쪽 인스팩터 영역의 [Size inspector]버튼을 클릭하면 객체의 위치 및 크기를 더욱 세밀하게 수정할 수 있다.



3. 뷰 모드를 수정한다.

이미지 뷰의 크기에 상관없이 이미지를 가로, 세로 비율을 유지하기 위하여 뷰 모드를 변경한다,

[Attributes inspector]버튼을 클릭한 후 View 항목의 Mode 를 [Aspect Fit]로 변경한다.



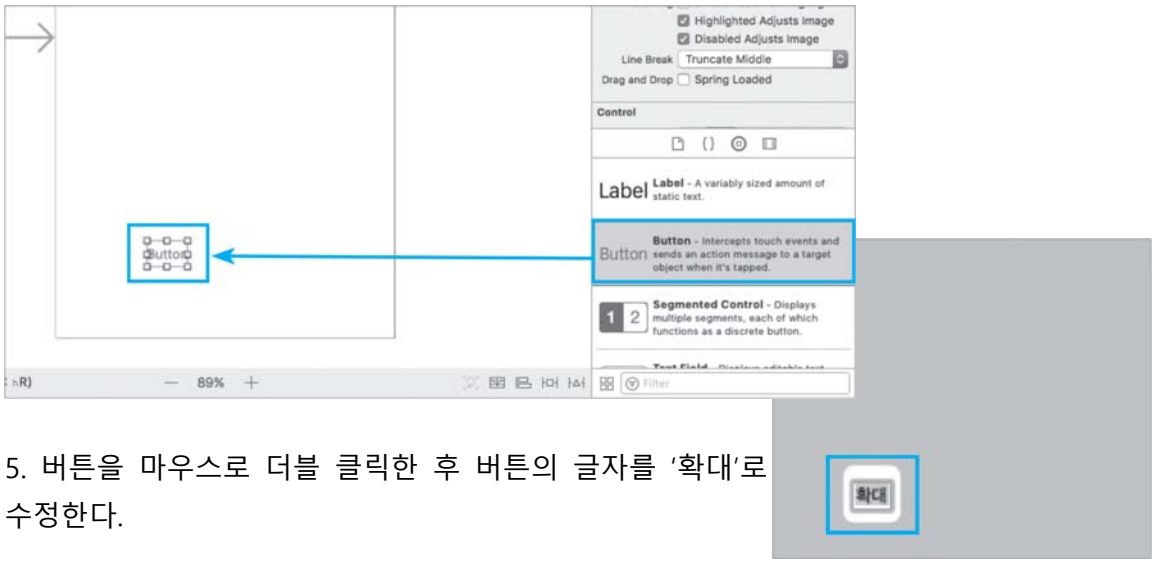
이미지 뷰에서 이미지를 나타내는 뷰 모드에 따라 이미지가 다르게 표시된다,

- Scale to Fill : 기본 설정값으로, 이미지 뷰의 크기에 맞게 이미지의 가로, 세로 비율이 변경된다. 비율이 맞지 않으면 뭉개져 보인다.
- Aspect Fit : 이미지의 가로, 세로 비율은 유지하면서 뷰의 크기에 맞게 이미지 크기를 바꾼다.
- Aspect Fill : 이미지의 비율을 유지하면서 이미지 뷰를 채웁니다. 뷰와 이미지 비율이 맞지 않으면 이미지가 넘쳐서 잘릴 수 있다.
- Center : 이미지의 원본 크기를 유지한 채 이미지의 중앙을 이미지뷰에 출력
- Top : 이미지의 원본 크기를 유지한 채 이미지의 윗부분을 이미지 뷰에 출력

4. 이미지를 확대하고 축소할 버튼을 추가하기

이미지를 확대하거나 축소할 수 있는 버튼을 만들어 보겠다.

오브젝트 라이브러리에서 [버튼(Button)]을 찾아 스토리보드로 끌어와 화면 아래쪽에 배치한다.

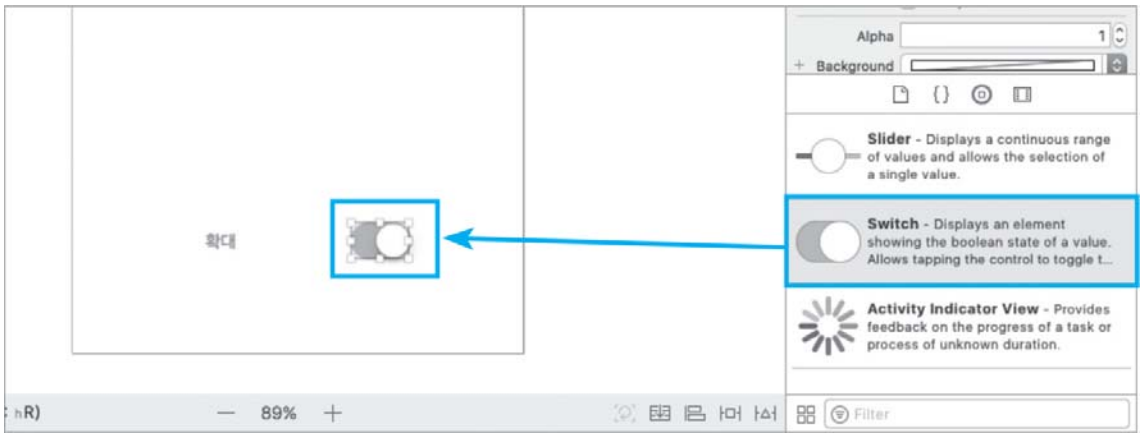


5. 버튼을 마우스로 더블 클릭한 후 버튼의 글자를 '확대'로 수정한다.

6. 이미지를 변경할 [스위치]추가하기

이미지를 변경할 수 있는 스위치를 추가한다. 라이브러리를 스크롤하여 [스위치(Switch)]를 찾아 스토리보드로 끌어와 오른쪽 아랫부분에 배치한다.

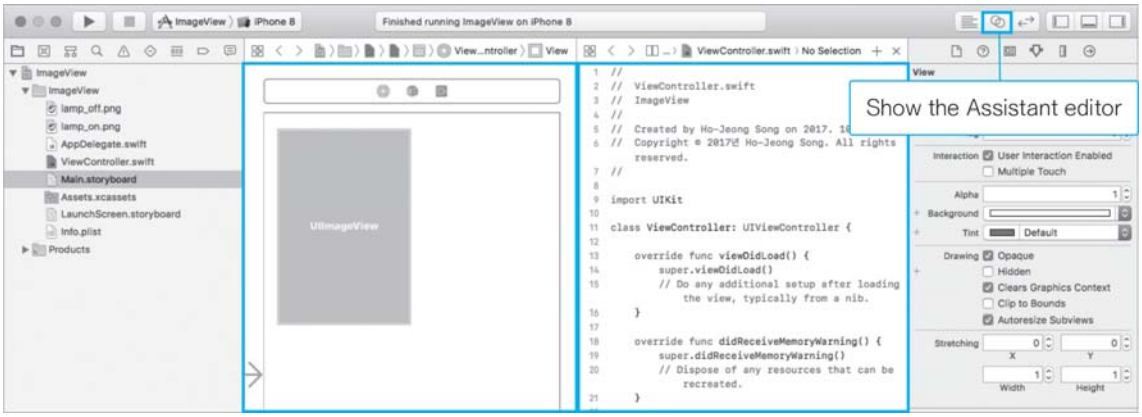
버튼은 일반적으로 한 가지 일을 하도록 지시하는 역할을 하고, 스위치는 [On/Off]상태를 바꾸는 역할을 하는 객체이다.



03-4 아웃렛 변수와 액션 함수 추가하기

소스 작업을 위한 보조 편집기 영역 열기

아웃렛 변수와 액션 함수를 추가하기 위해서는 보조 편집기 영역(Assistant editor)을 열어야 한다. 화면 오른쪽 윗부분의 [Show the Assistant editor]버튼을 클릭하면 화면 가운데의 스토리보드 부분이 둘로 나눠어지면서 왼쪽에는 스토리보드, 오른쪽에는 소스를 편집하는 영역이 나타난다.

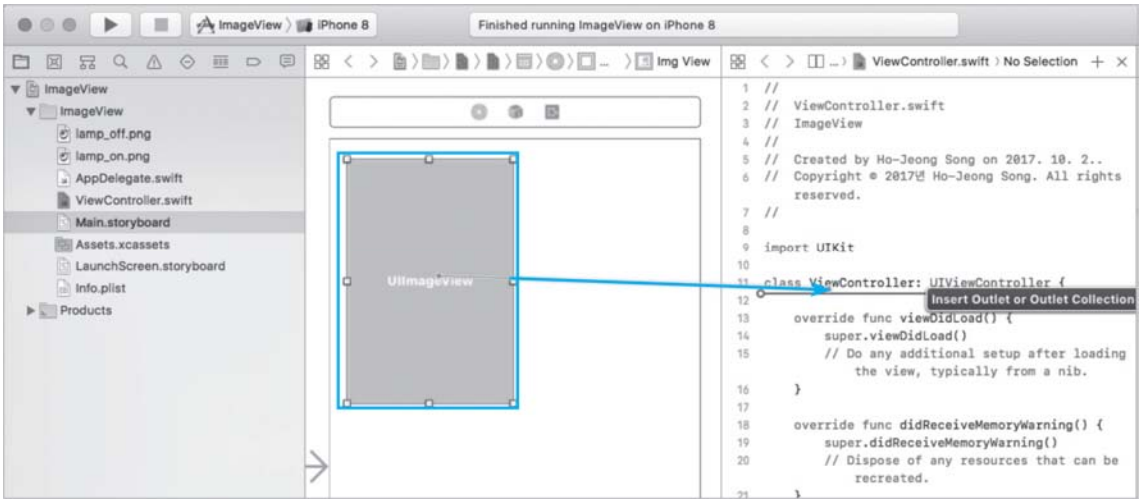


이미지 뷰에 아웃렛 변수 추가

1. 이미지 뷰를 마우스 오른쪽 버튼으로 선택한 후 오른쪽 보조 편집기 영역으로 드래그한다.

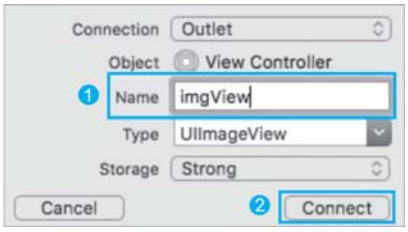
연결선이 나타나면 연결선을 뷰 컨트롤러(View Controller)의 클래스 선언 바로 아래에 놓는다.

아웃렛 변수는 일반적으로 클래스(class)선언부 바로 아래에 추가 한다.

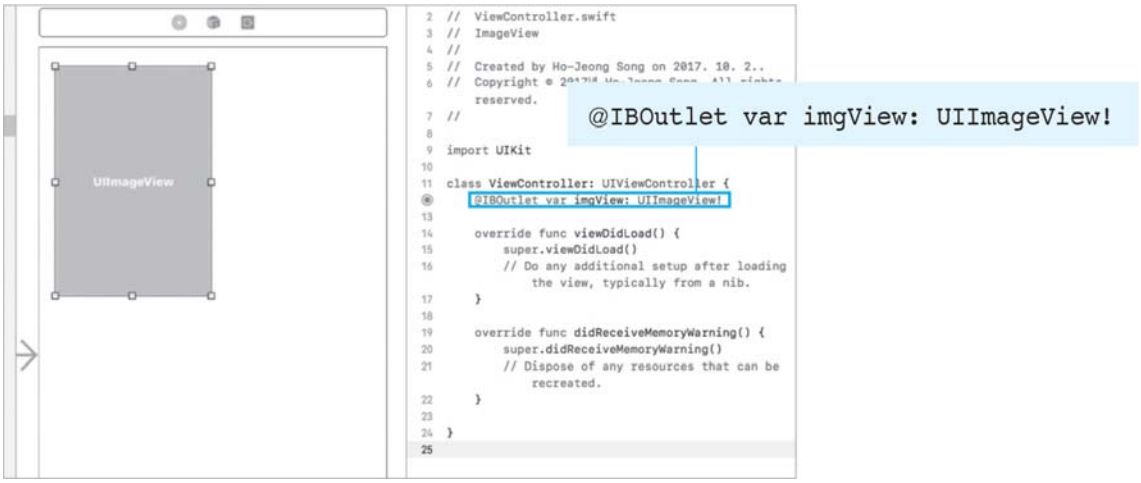


2. 아웃렛 변수의 이름(Name)입력란에 'imgView'라고 입력하고 [Connect]버튼을 클릭하면 이미지 뷰와 아웃렛 변수가 연결된다,

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

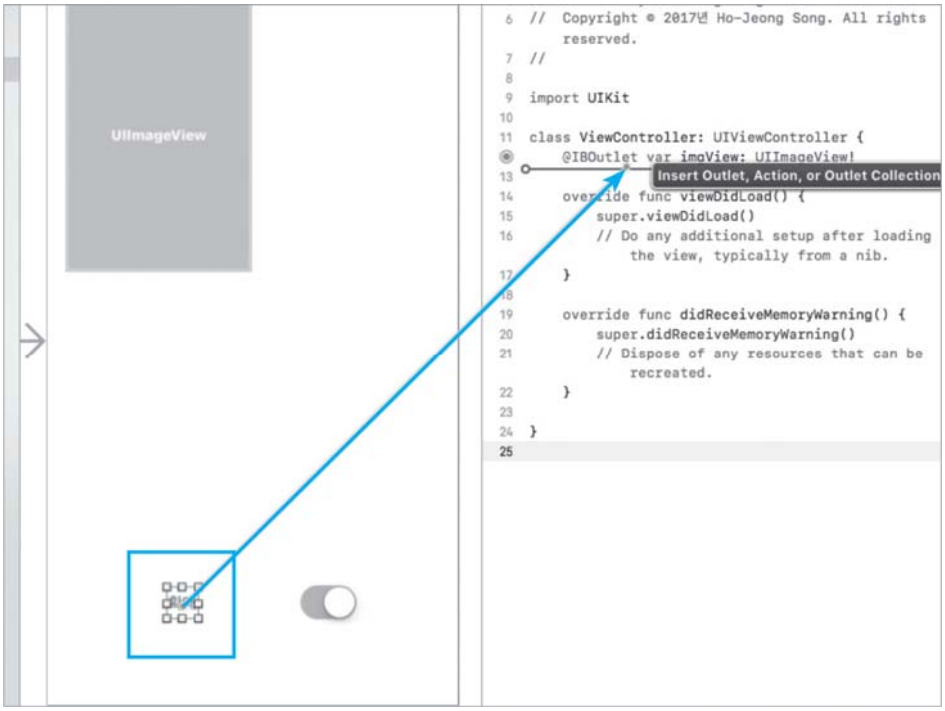


3. 이미지 뷰에 대한 아웃렛 변수가 추가 되었다.



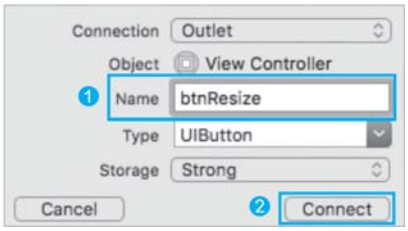
4. [확대/축소]버튼의 아웃렛 변수 추가하기

화면 왼쪽 아랫부분에 있는 [확대]버튼을 마우스 오른쪽 버튼으로 클릭한 후 드래그하여 오른쪽 보조 편집기 영역의 조금전에 추가한 'imgView'변수 아래에 놓는다.

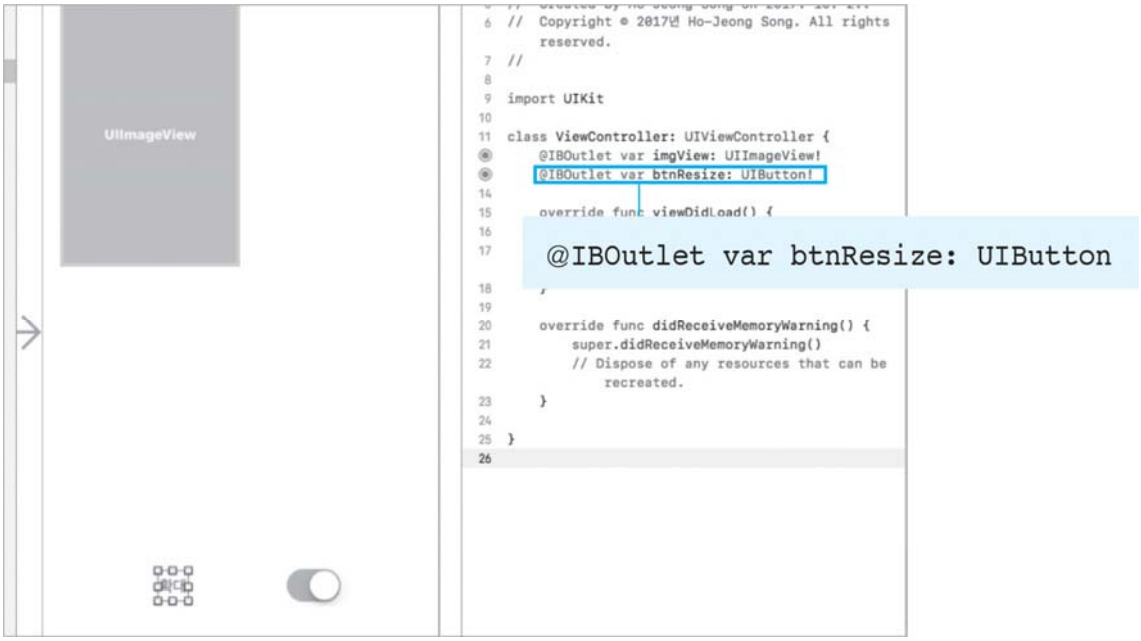


5. [확대]버튼을 마우스 오른쪽 버튼으로 클릭한 후 드래그하여 편집기 영역에 드롭하면 다음과 같이 연결 설정 창이 나타난다. 여기서 아래 표와 같이 적어 준다.

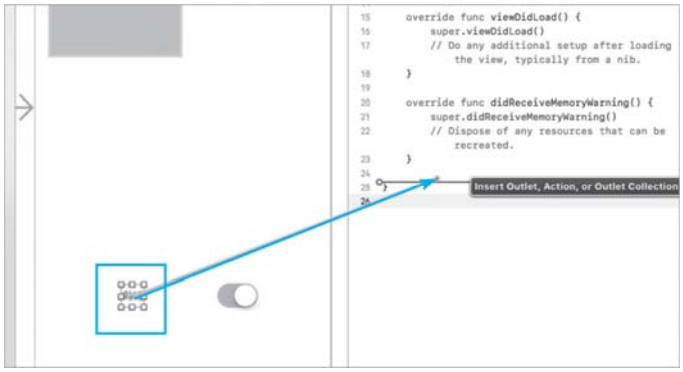
위치	imgView 아웃렛 변수 아래
연결(Connection)	Outlet
이름(Name)	btnResize
유형(Type)	UIButton



6. 버튼에 대한 아웃렛 변수가 추가되었다.
btnResize 변수를 사용해 버튼의 타이틀을 수정할 수 있다.



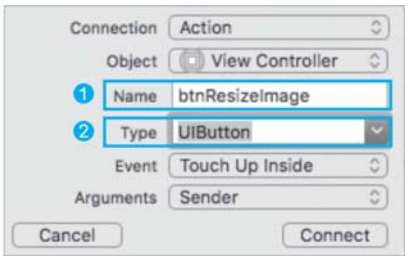
7. 버튼에 대한 액션 추가하기
[확대]/[축소] 버튼에 대한 액션 함수를 추가해 보자. 이 버튼을 클릭하면 이미지가 확대되거나 축소되는 기능을 하는 액션 함수이다.
이 액션 함수를 만들기 위해 마우스 오른쪽 버튼으로 [확대]버튼을 클릭한 후 드래그해서 오른쪽 보조 편집기 영역의 맨 아랫부분(클래스 닫음 괄호')' 바로 윗부분)에 놓는다.



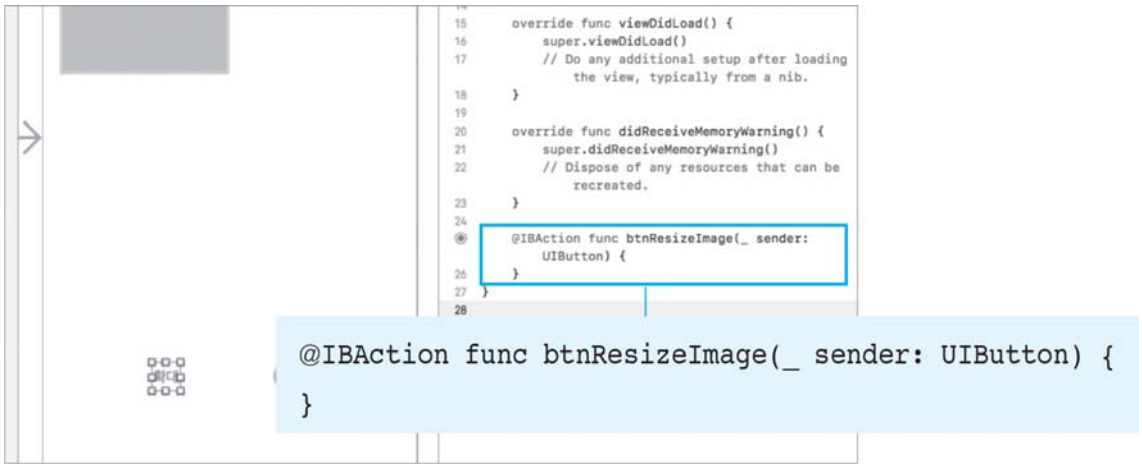
[Outlet]으로 선택되어 있는 연결을 [Action]으로 변경해야 한다.

8. 이름을 아래 표와 같이 입력을 하고 버튼에 액션을 추가하는 것이므로 (Type)은 [UIButton]으로 설정한다.
변경하지 않으면 [AnyObject]가 자동으로 선택된다. 동작에는 문제는 없지만 명확하게 지정해주는 것이 좋다.

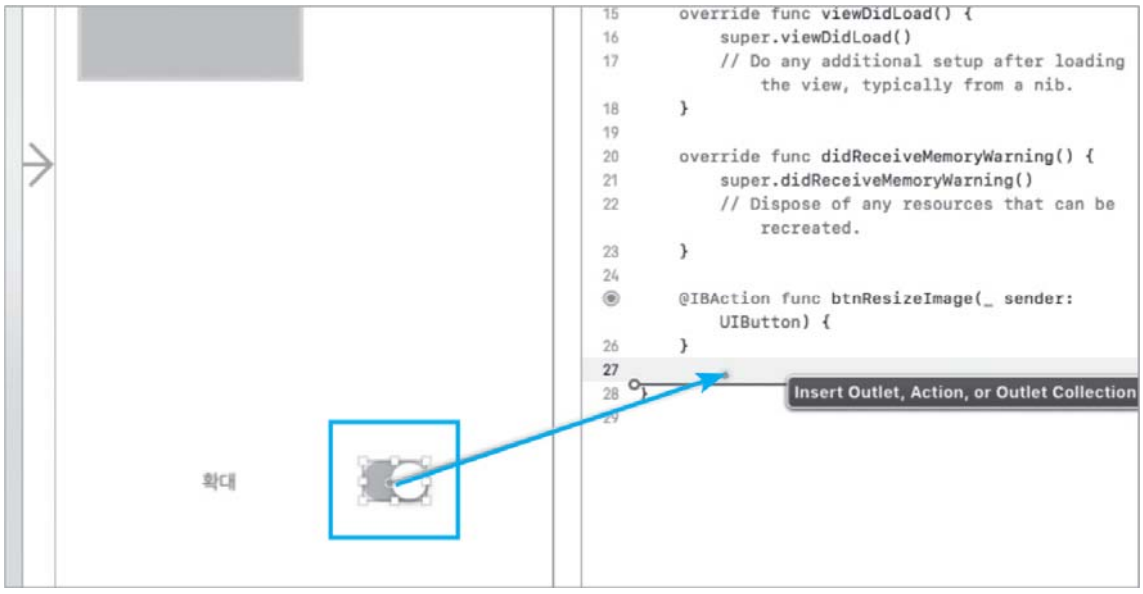
위치	클래스 닫음 괄호 '}' 바로 위
연결(Connection)	Action
이름(Name)	btnResizeImage
유형(Type)	UIButton



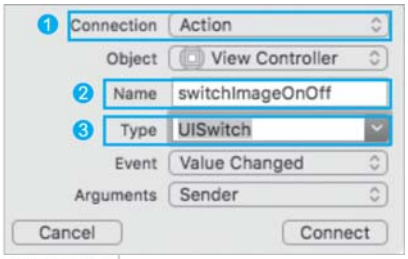
9. [Connect]버튼을 클릭하면 [확대]버튼에 대한 액션 함수가 추가 된 것을 볼 수 있다.

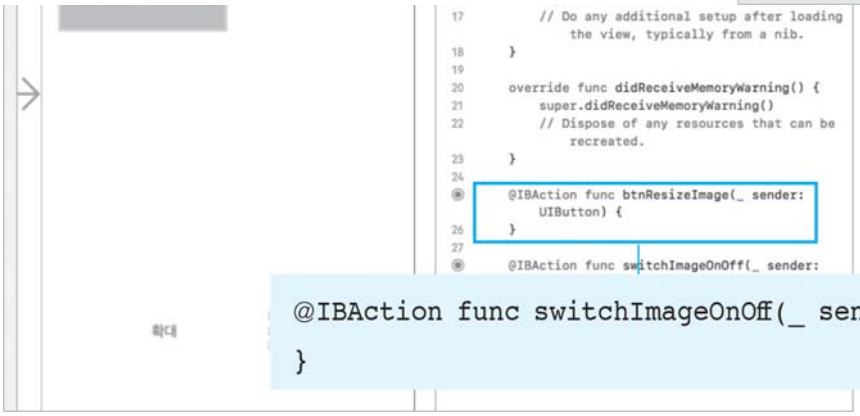


11. 스위치에 대한 액션 함수 추가하기
마지막으로 스위치에 대한 액션 함수를 추가하면 된다. 아래표와 같이 입력을 해준다.



위치	btnResizeImage 액션 함수 아래
연결(Connection)	Action
이름(Name)	switchImageOnOff
유형(Type)	UISwitch






```
@IBAction func switchImageOnOff(_ sender: UISwitch) {
}
```

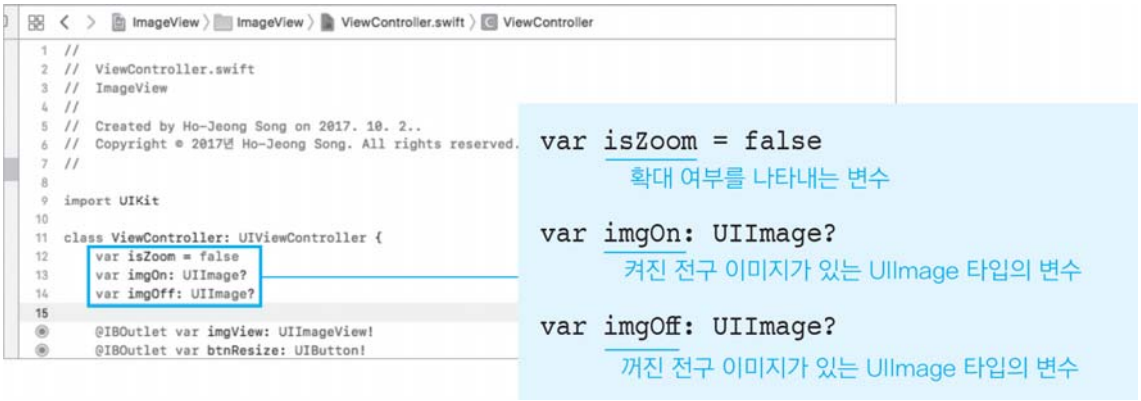
03-5 이미지뷰 앱의 기능 구현하기기

사용자가 [확대]버튼을 클릭할 때 동작할 액션 함수와 스위치를 [On]/[Off]하였을 때 동작할 액션 함수를 코딩해 보자.

1. 오른쪽 윗 부분에서 [Show the Standard editor]버튼을 클릭한다. 그리고 나서 왼쪽의 내비케이터 영역에서 [ViewController.swift]를 선택한다.

2. 변수 추가하기

코딩에 필요한 변수들을 뷰 컨트롤러(ViewController)의 클래스 선언문 바로 아래에 추가한다.



**** 옵셔널 변수 ****

오른쪽 소스처럼 스위프트 코드를 입력하다 보면 변수 선언 뒤에 물음표 '?'가 붙은 것을 볼 수 있다.

```
var imgOn: UIImage?
```

이 물음표 '?'를 삭제하면 바로 에러가 발생한다.

그러면 이 물음표는 왜 붙었을까요? 이것은 C나 오브젝티브-C에는 존재하지 않고 오직 스위프트에만 있는 '옵셔널(Optionals)'이라는 개념이다.

옵셔널은 어떤 값이 존재하지 않는다는 것을 나타낼 때 사용 한다. 즉, 변수가 nil(=null)이거나 값의 존재 여부를 알 수 없다는 것을 의미한다. 스위프트에서는 변수를 선언할 때는 변수에 반드시 nil 이 아닌 값을 할당해야 하지만 옵셔널 타입을 사용해서 변수에 값이

```
var index: Int?

index = 3

if index != nil {
   print(index!)
}
```

없다는 것을 알 릴 수 있다. 오른쪽 위 소스에서는 변수 imgOn 을 선언 했는데 초깃값을 주지 않았기 때문에 '값이 없을 수 있다'는 의미로 [?]를 붙여 줘야 하는 것이다.

```
var index: Int!

index = 3

if index != nil {
   print(index)
}
```

옵셔널로 선언된 변수에 값이 할당되면 그 값은 '옵셔널에 래핑(wrapped)되었다'고 하며, 이 값은 '!'를 사용하여 강제 언래핑(force unwrapping)하여 값에 접근할 수 있다.
또한 옵셔널은 암묵적인 언래핑(implicity unwrapping) 이 되도록 선언할 수 있는데, 이 때는 강제 언래핑을 사용하지 않아도 값에 접근할 수 있다.

3. 이미지 지정 후 보여주기
viewDidLoad 함수 내 UIImage 타입의 변수에 이미지를 지정하기 위한 코드를 추가한다.

```
@IBOutlet var btnResize: UIButton!

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    imgOn = UIImage(named: "lamp_on.png")
    imgOff = UIImage(named: "lamp_off.png")

    imgView.image = imgOn
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
```

```
override func viewDidLoad() {
    super.viewDidLoad()
   // Do any additional setup after loading the view, typically from a nib.

    imgOn = UIImage(named: "lamp_on.png")
    imgOff = UIImage(named: "lamp_off.png")

    imgView.image = imgOn
}
```

1) UIImage 타입의 변수 imgOn 과 imgOff 에 이미지 파일명을 입력한다.
이때 이미지 파일명은 앞에서 프로젝트에 추가한 이미지의 파일명을 입력하면 된다.
2) 스토리보드에 추가한 이미지 뷰의 아웃렛 변수 'imgView'에 방금 선언한 'imgOn' 이

미지를 지정한다.

viewDidLoad 함수란?
viewDidLoad 함수는 내가 만든 뷰를 불러왔을 때 호출되는 함수로, 부모 클래스인 UIViewController 클래스에 선언되어 있다. 뷰가 불러진 후 실행하고자 하는 기능이 필요할 때 이 viewDidLoad 함수내에 코드를 입력하면 된다.

4. 버튼 클릭 시 동작하는 함수 코딩
[확대] 버튼을 클릭했을 때 동작하는 btnResizeImage 함수를 코딩해 보자.
아래 코드에서 'CGFloat'은 'Xcode 에서 Float 을 재정의해 놓은 자료형'으로, Float 과 같다고 생각하면 된다.

```
19      override func viewDidLoad() {
20          super.viewDidLoad()
21          // Do any additional setup after loading the view, typically from a nib.
22          imgOn = UIImage(named: "lamp_on.png")
23          imgOff = UIImage(named: "lamp_off.png")
24          imgView.image = imgOn
25      }
26
        @IBAction func btnResizeImage(_ sender: UIButton) {
28          let scale:CGFloat = 2.0        Immutable value 'scale' was never used; consider replacing with '_' or removing it
29          var newWidth:CGFloat, newHeight:CGFloat  2  Variable 'newHeight' was never used; consider replacing with '_' o...
30      }
31
```

28 행은 [확대]버튼을 클릭할 경우 이미지를 두배로 확대할 것이므로 확대할 스케일 값을 보관할 scale 상수를 CGFloat(실수형)타입으로 선언하고 값을 "2.0"으로 설정한다.
29 행은 확대할 크기를 계산해서 보관할 변수 newWidth 와 newHeight 를 CGFloat 다입으로 선언한다.

[스위프트 문법] 기본 데이터 자료형
자료형이란 자료의 형태를 의미합니다. 변수나 상수를 선언할 때 사전에 어떠한 값을 저장할지는 자료형을 사용해서 정의해야 합니다. 예를 들어 정수를 저장하기 위해서는 Int(Integer) 자료형을 사용해야 하고, 실수를 저장하기 위해서는 Float 자료형을 사용해야 합니다. 또한 문자열을 저장하기 위해서는 String 자 료형을 사용해야 합니다.

타 입	특 징	예 제
Bool	참 또는 거짓 중 하나를 표현하는 이진법	True, False
Int, Int32, Int64	큰 수(분수 제외)를 표현하는 데 32 또는 64비트 음수나 양수의 정수 값 사용	4, 543, -674837, 5745
Int8, Int16	작은 수(분수 제외)를 표현하는 데 8 또는 16비트 음수나 양수의 정수 값 사용	-23, 58, 145

UInt, UInt32, UInt64	큰 수(분수 제외)를 표현하는 데 32 또는 64비트 양수 값 사용	5, 132, 70, 10023
UInt8, UInt16	작은 수(분수 제외)를 표현하는 데 8 또는 16비트 양수 값 사용	35, 86, 254
Float, Double	음수 또는 양수의 부동 소수점 또는 분수를 포함할 수도 있다	11.542, -3002.5899, 17.0
Character	단일 글자나 숫자 또는 다른 부호를 큰따옴표로 묶어서 표현	"T", "K", "*", "3"
String	일련의 문자를 큰따옴표로 묶어서 표현	"Fish", "Pigs", "New York"

6. 이제 isZoom 변수의 상태에 따라 이미지 프레임의 크기를 확대 또는 축소하고 버튼의 텍스트를 변경하는 코드를 작성해 보자.
우선 버튼을 눌렀을 때 isZoom 변수의 조건에 따라 일을 수행하기 위하여 if 문을 사용하는 구조를 만든다. 그리고 맨 마지막에 isZoom 변수의 상태를 '!'연산자를 사용하여 반전을 시킨다.

```
35  @IBAction func btnResizeImage(_ sender: UIButton) {
36      let scale: CGFloat = 2.0
37      var newWidth: CGFloat, newHeight: CGFloat
38
39      if (isZoom) {   // true
40          newWidth = imgView.frame.width/scale
41          newHeight = imgView.frame.height/scale
42          imgView.frame.size = CGSize(width: newWidth, height: newHeight)
43
44          btnResize.setTitle("확대", for: .normal)
45      }
46      else {  // false
47          newWidth = imgView.frame.width*scale
48          newHeight = imgView.frame.height*scale
49          imgView.frame.size = CGSize(width: newWidth, height: newHeight)
50
51          btnResize.setTitle("축소", for: .normal)
52      }
53
54      isZoom = !isZoom
  }
```

38 행은 현재 상태가 "확대"일 때 즉, isZoom 변수 값이 True 일 때
39 행과 40 행은 이미지 프레임의 가로, 세로 크기에 scale 값을 나누어 newWidth 와 new Height 에 할당한다.
41 행의 CGSize 메서드를 사용하여 imgView 의 프레임크기를 변경한다,
43 행 버튼의 텍스트를 "확대"로 변경한다.

45 행은 현재가 '축소'일 때 (즉, isZoom 변수 값이 false 일 때)
46 행~47 행은 이미지 프레임의 가로, 세로 크기에 scale 값을 곱하여 newWidth 와 newHeight 에 할당한다.
48 행은 CGSize 메소드를 사용하여 이미지 뷰의 프레임 크기를 변경한다.
50 행은 버튼의 텍스트를 "축소"로 변경한다.

7. 스위치 클릭 시 동작하는 함수 코딩
[On]/[Off]스위치를 클릭할 때 동작하는 switchImageOnOff 함수를 코딩한다, 스위치 상태에 따라 이미지 뷰에 나타낼 이미지를 선택한다.

```
45      @IBAction func switchImageOnOff(_ sender: UISwitch) {
46          if sender.isOn {
47              imgView.image = imgOn
48          }else{
49              imgView.image = imgOff
50          }
51      }
```