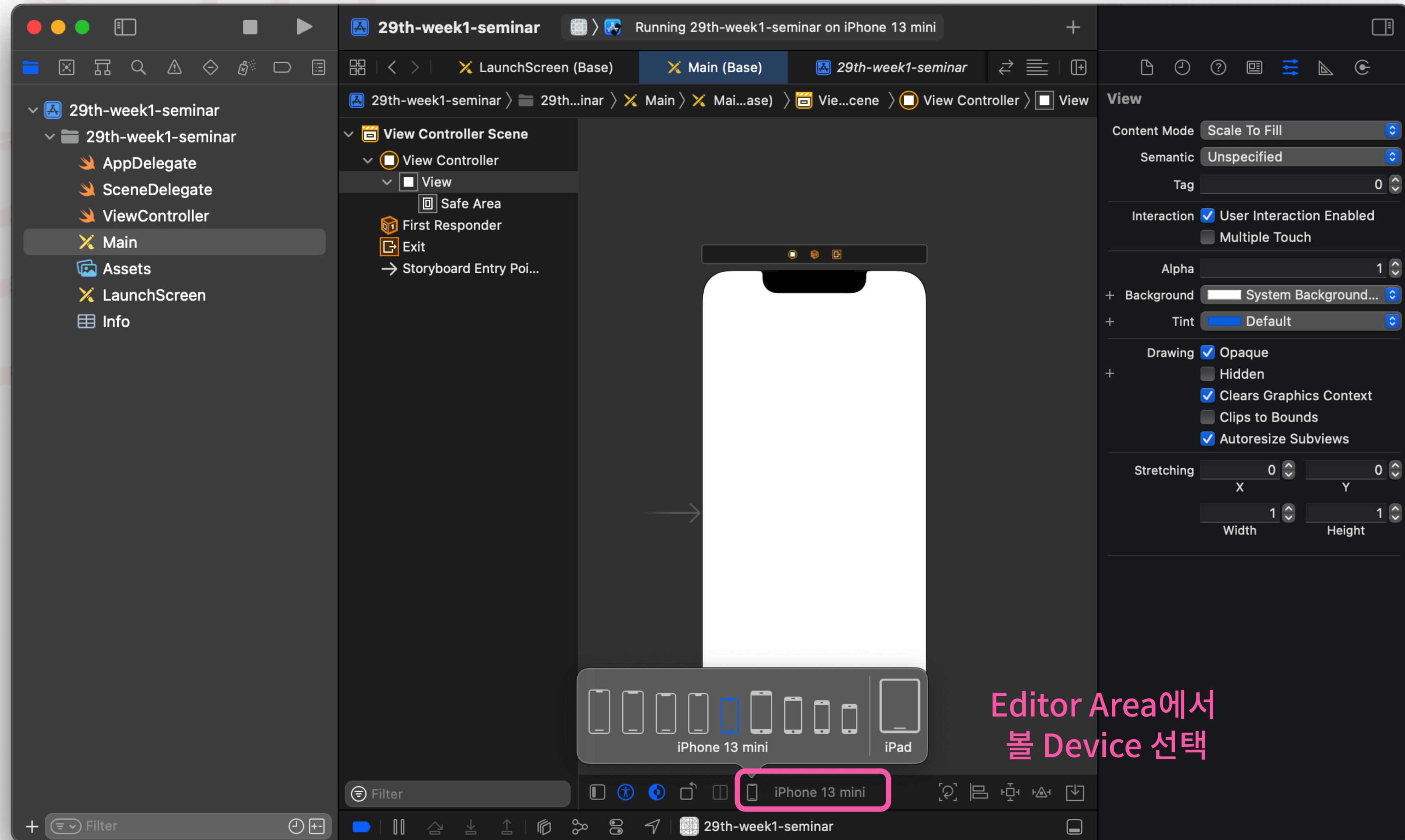


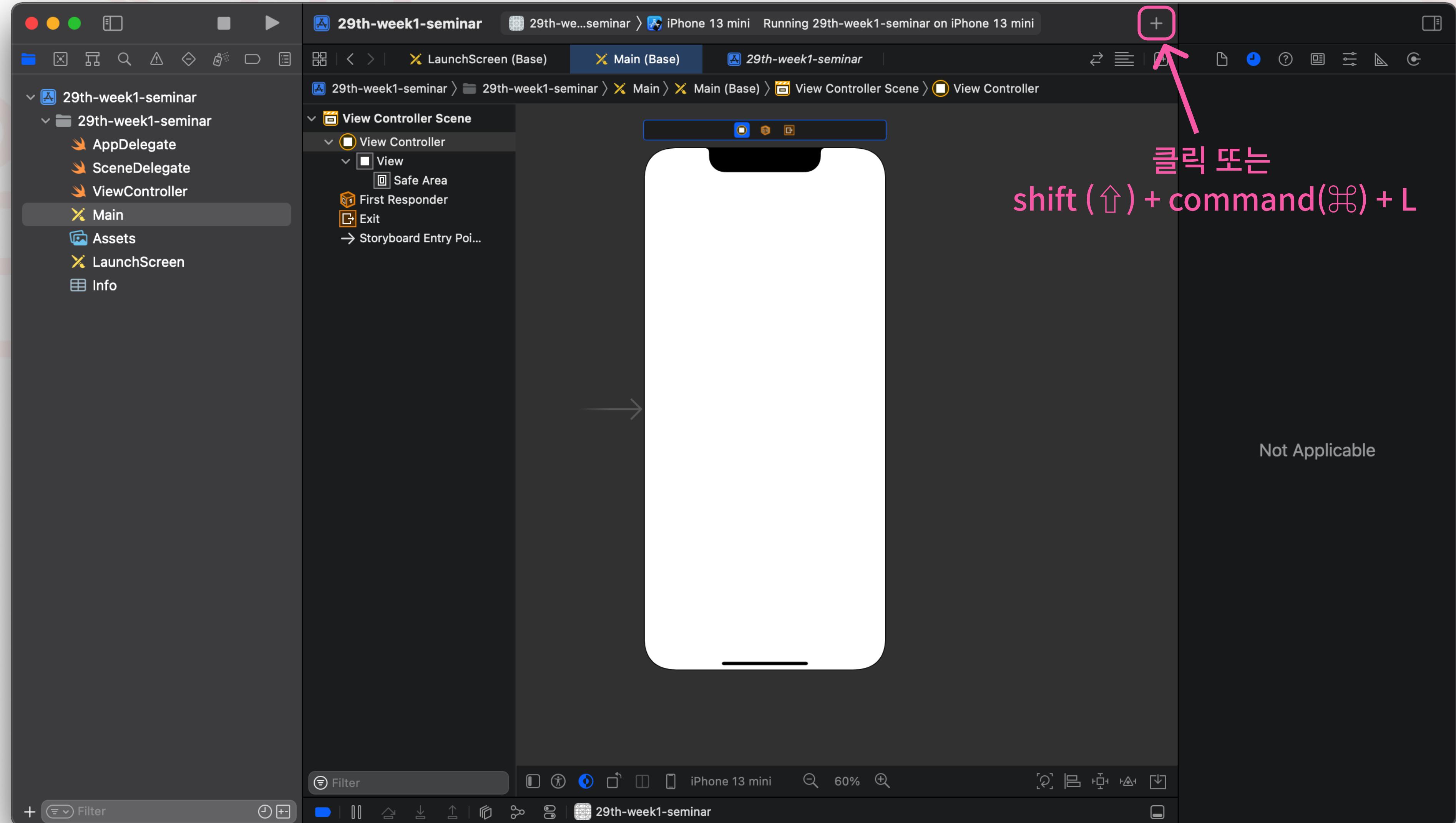
03

UI 구성하기

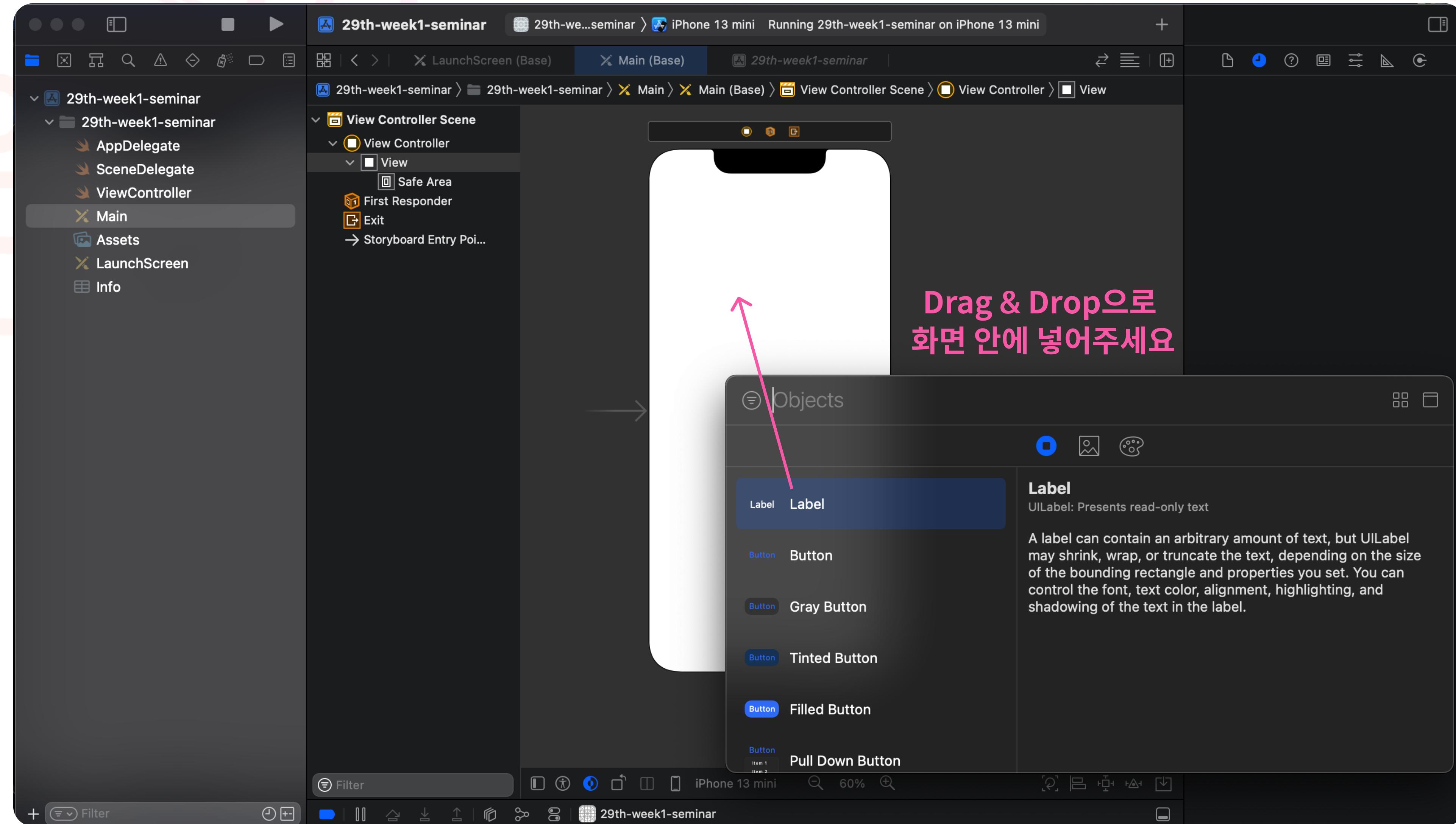
UI 구성하기 | 실습



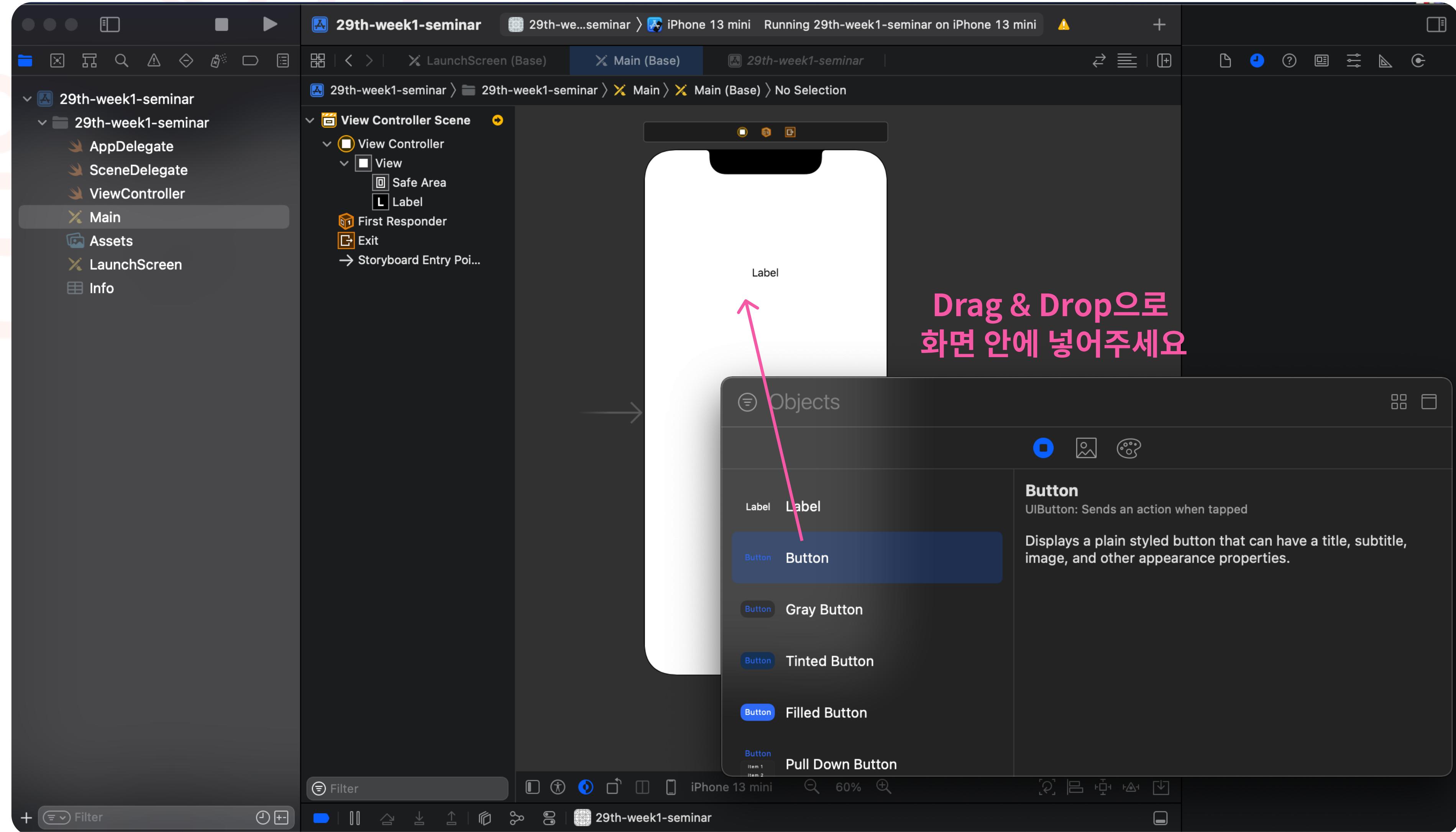
UI 구성하기 | 실습



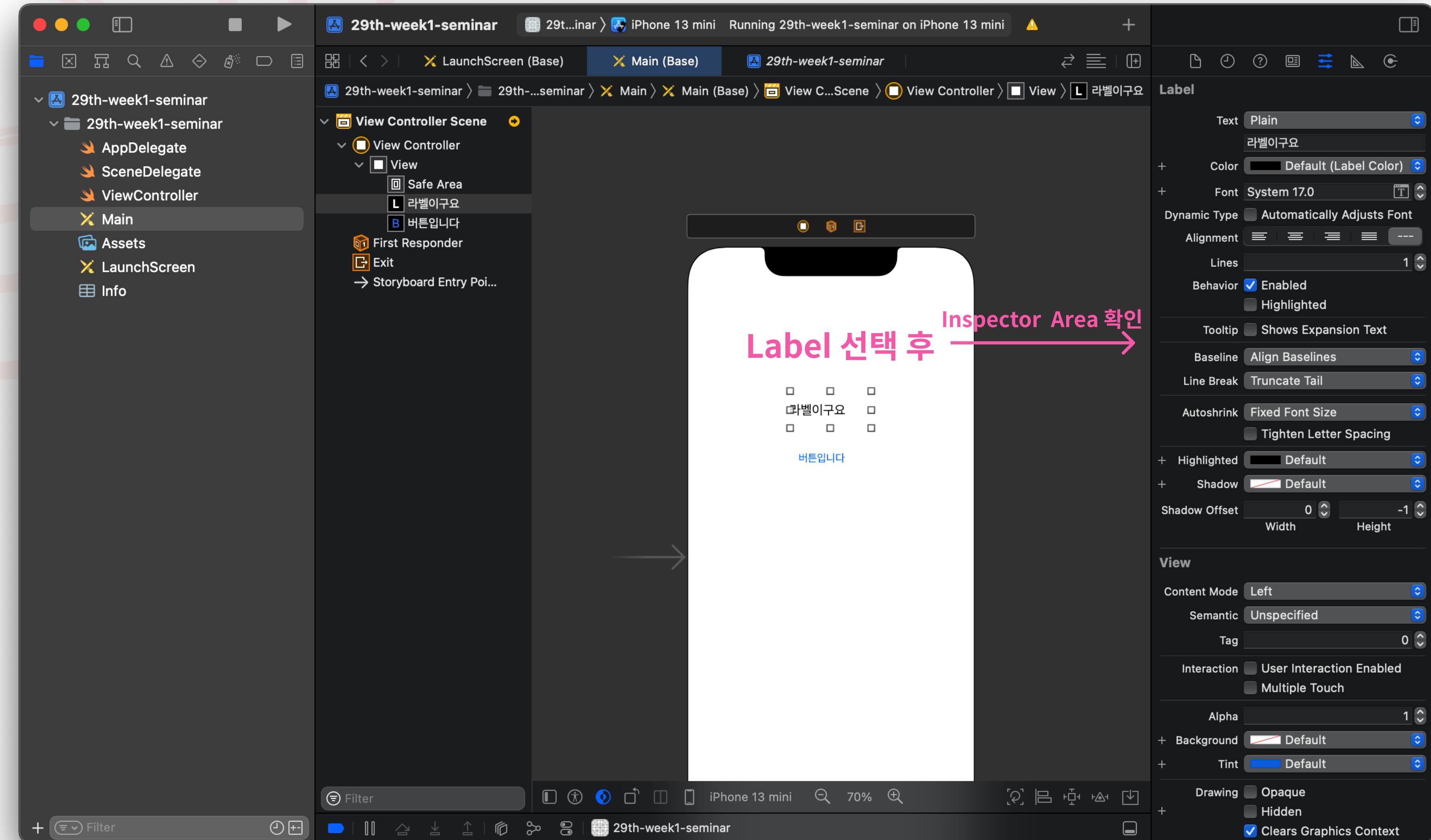
UI 구성하기 | 실습



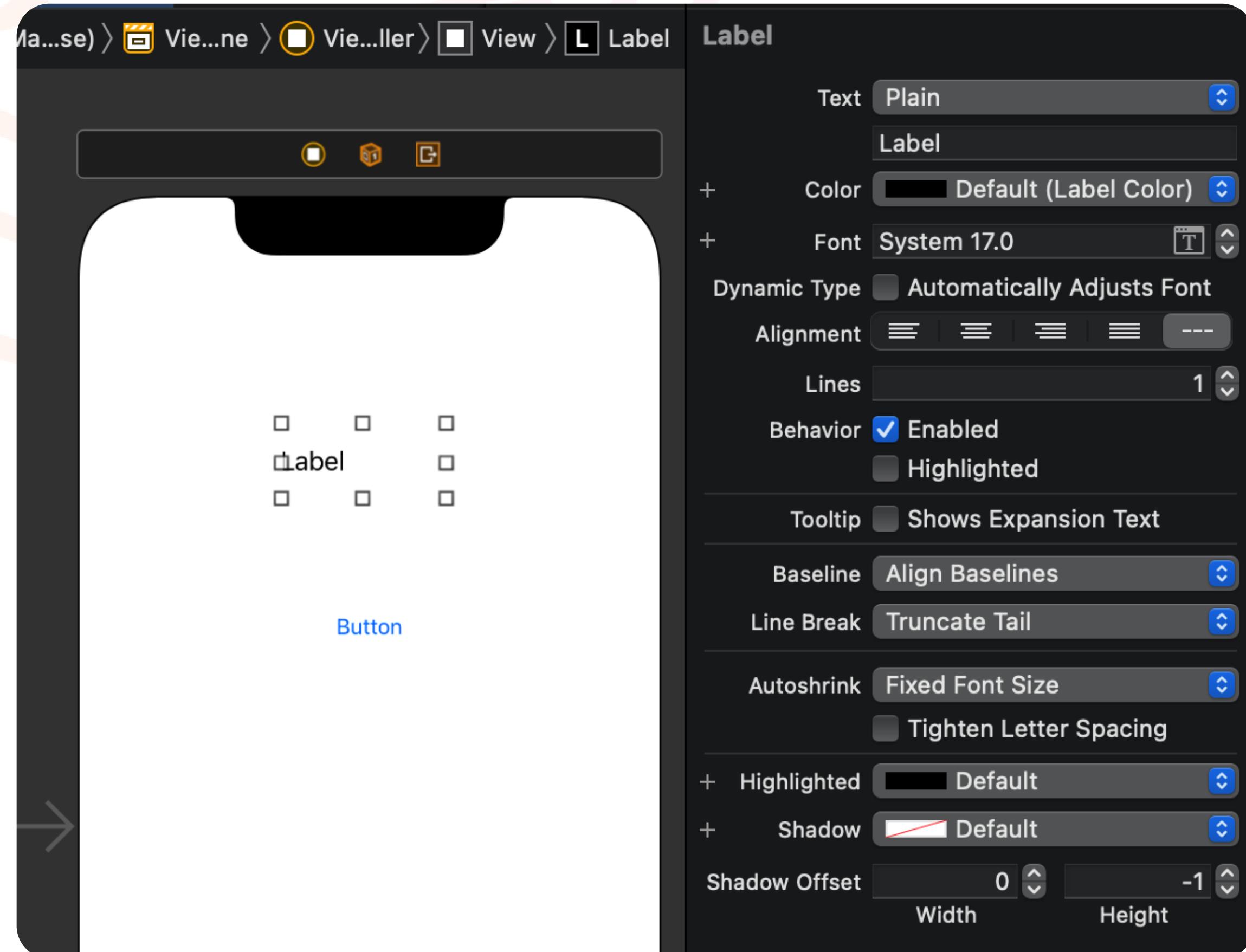
UI 구성하기 | 실습



UI 구성하기 | 실습

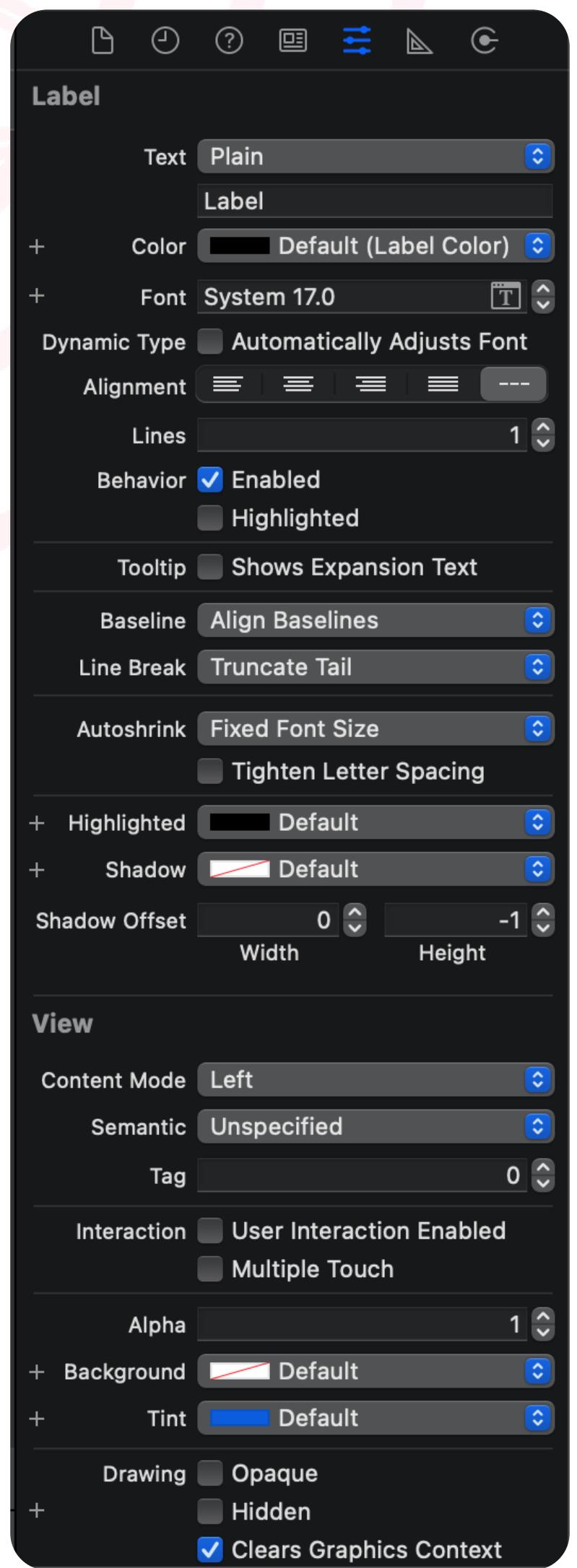


UI 구성하기 | 실습

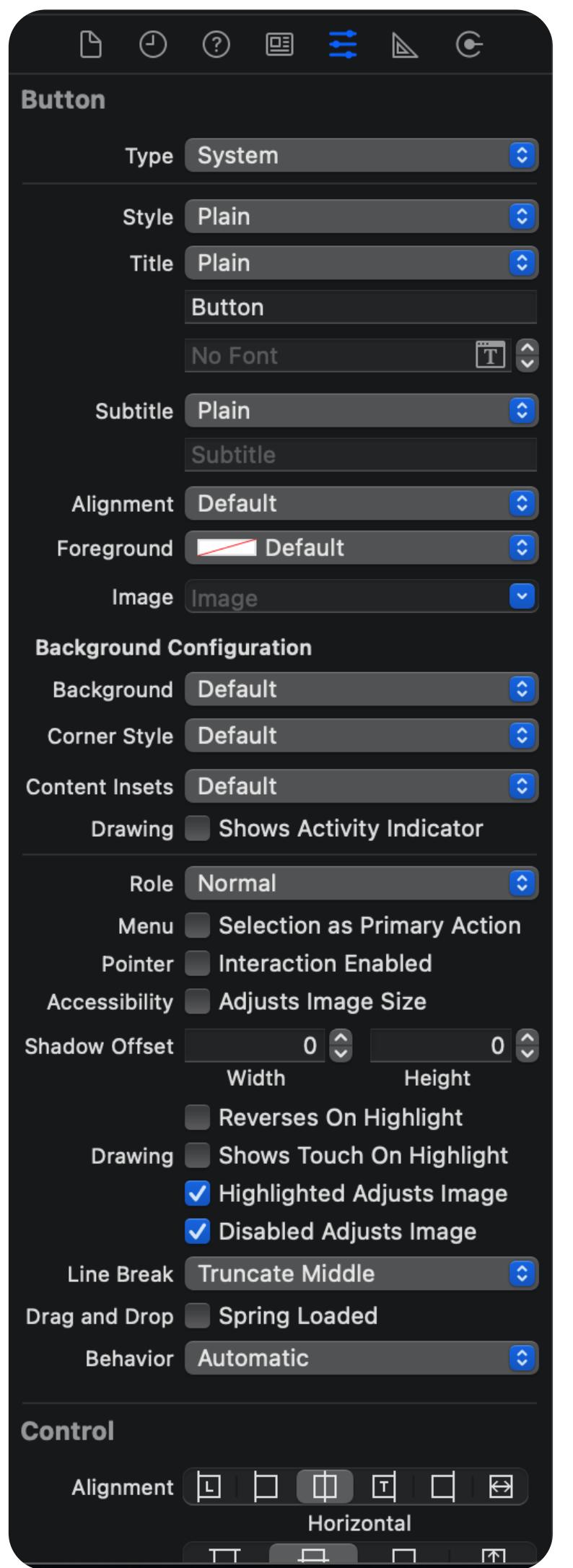


- Text Style : 텍스트 스타일을 정의. Attributed 를 선택하면 세밀한 스타일 조정이 가능
- Text : 원하는 텍스트 내용을 넣을 수 있음.
- Color : 텍스트 컬러 변경
- Font : 폰트 변경
- Dynamic Type : 사용자가 글씨 크기를 조정했을 때, 동적으로 크기 조절 유무
- Alignment : 글자 정렬 (LtR,RtL,중앙정렬등) 방향
- Lines : 최대로 들어갈 줄 수 (제한을 두고 싶지 않다면, 0)

Label

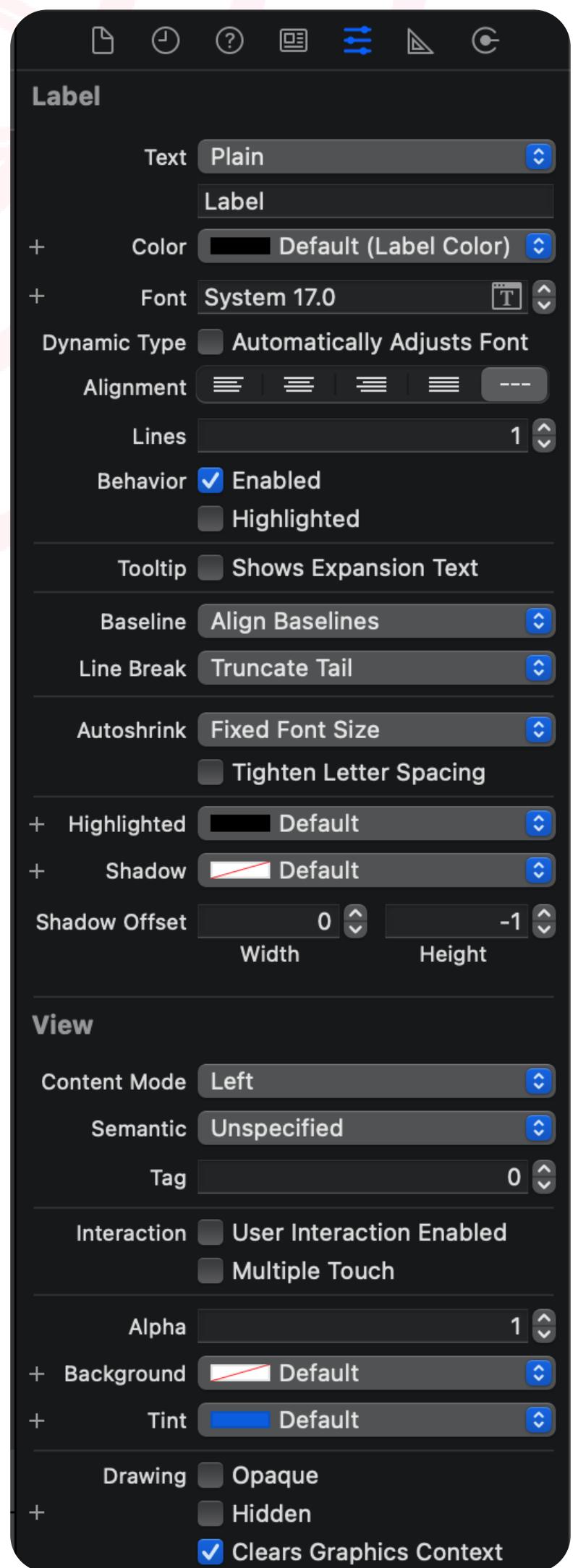


Button

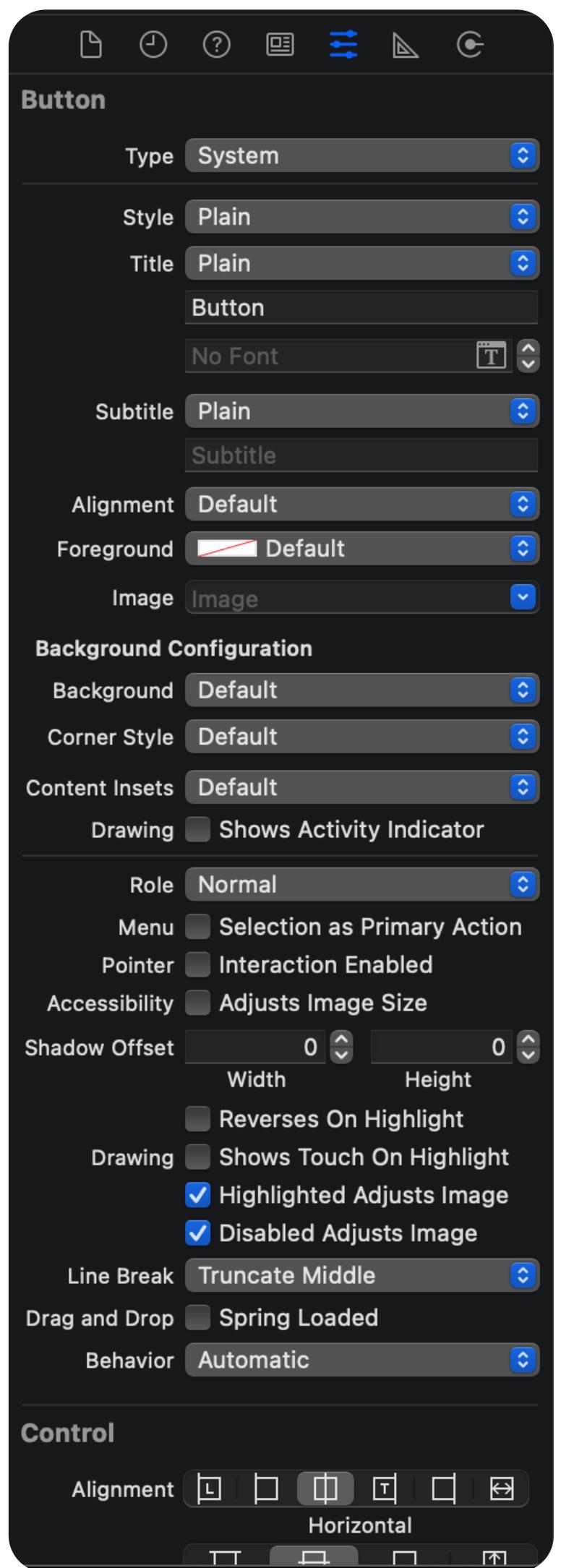


자유롭게 Label과 Button 텍스트, 색상, 정렬, 폰트 등 바꾸어봅시다!

Label

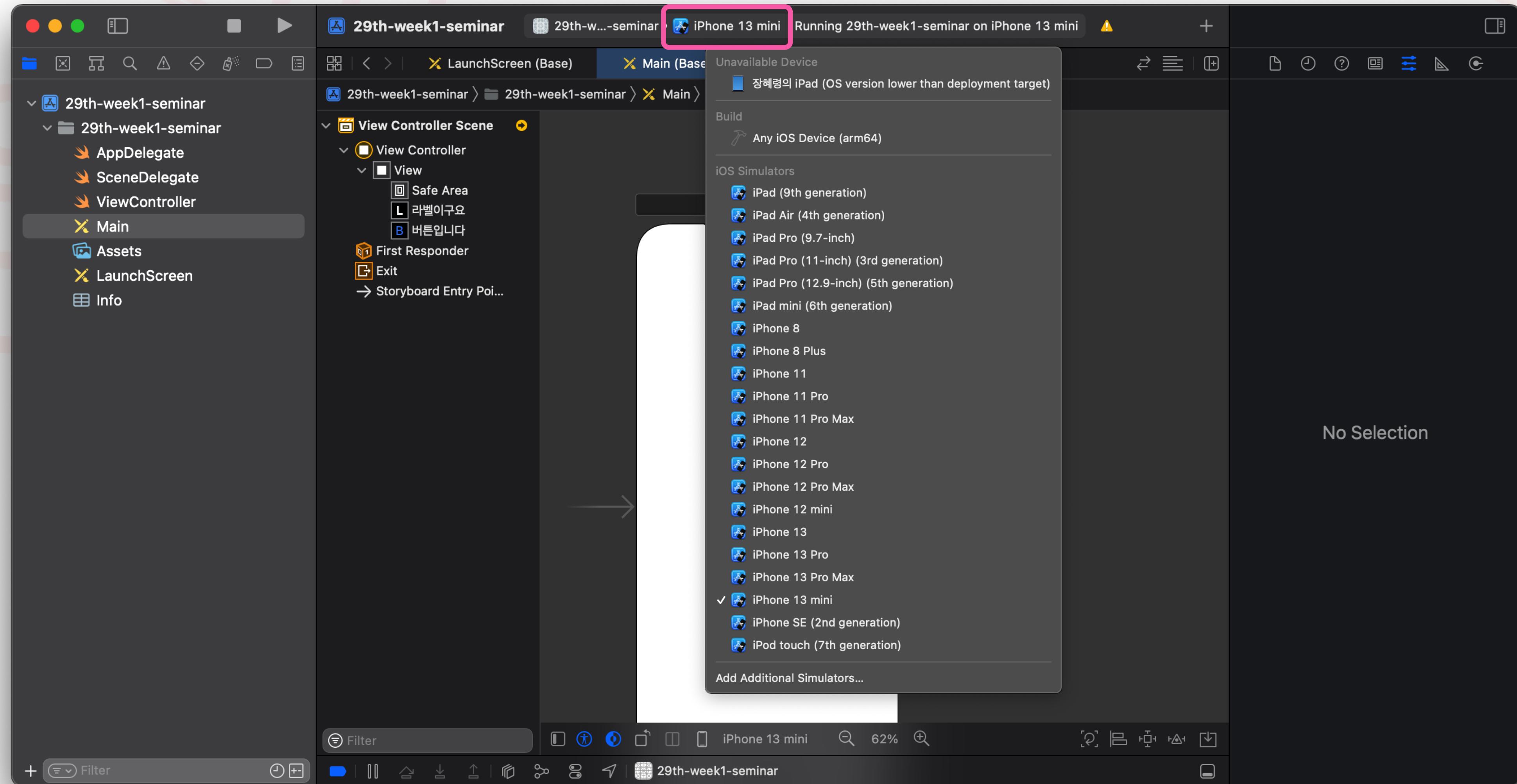


Button



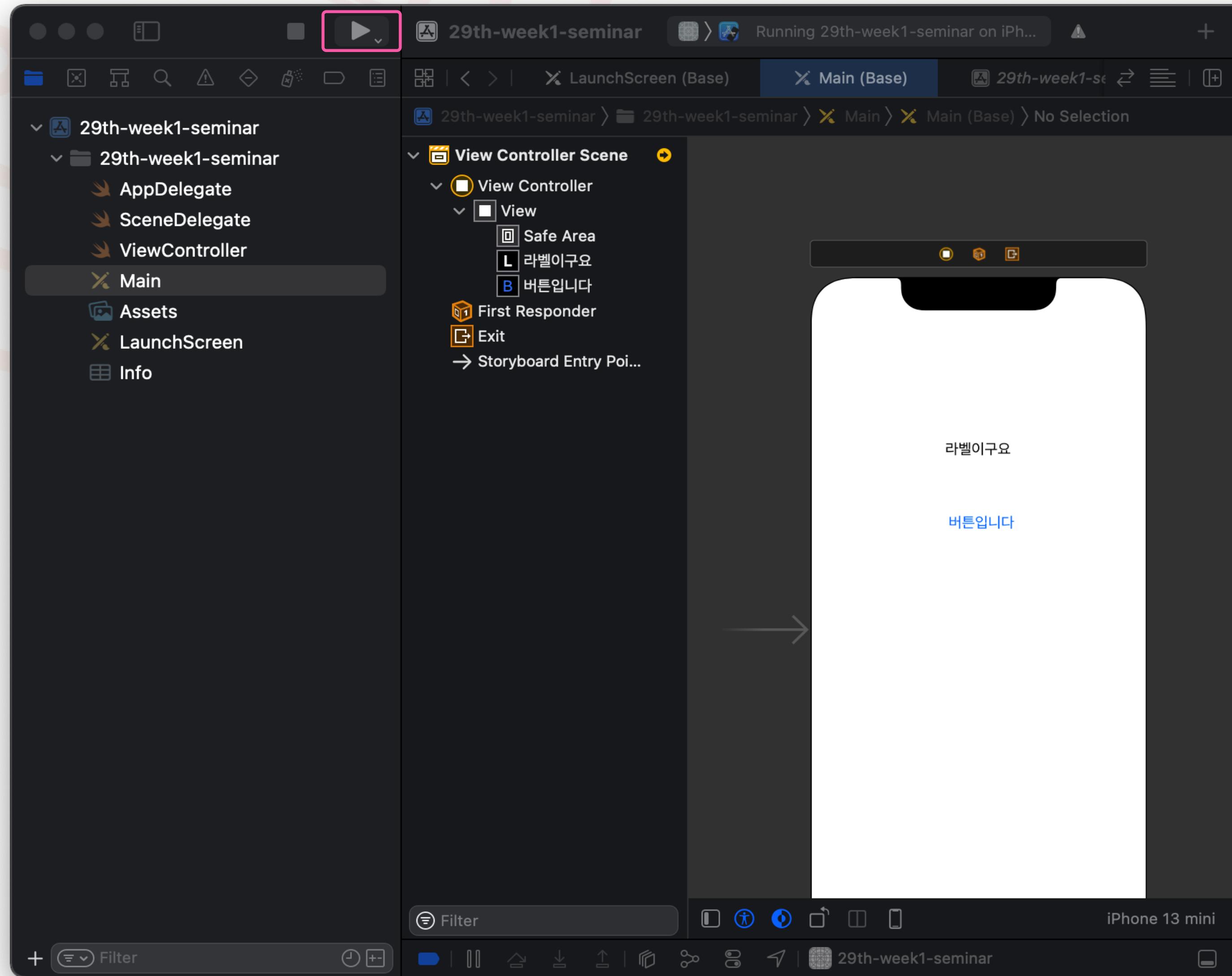
자유롭게 Label과 Button 텍스트, 색상, 정렬, 폰트 등 바꾸어봅시다!

눌러서 원하는 시뮬레이터를 선택해주세요!

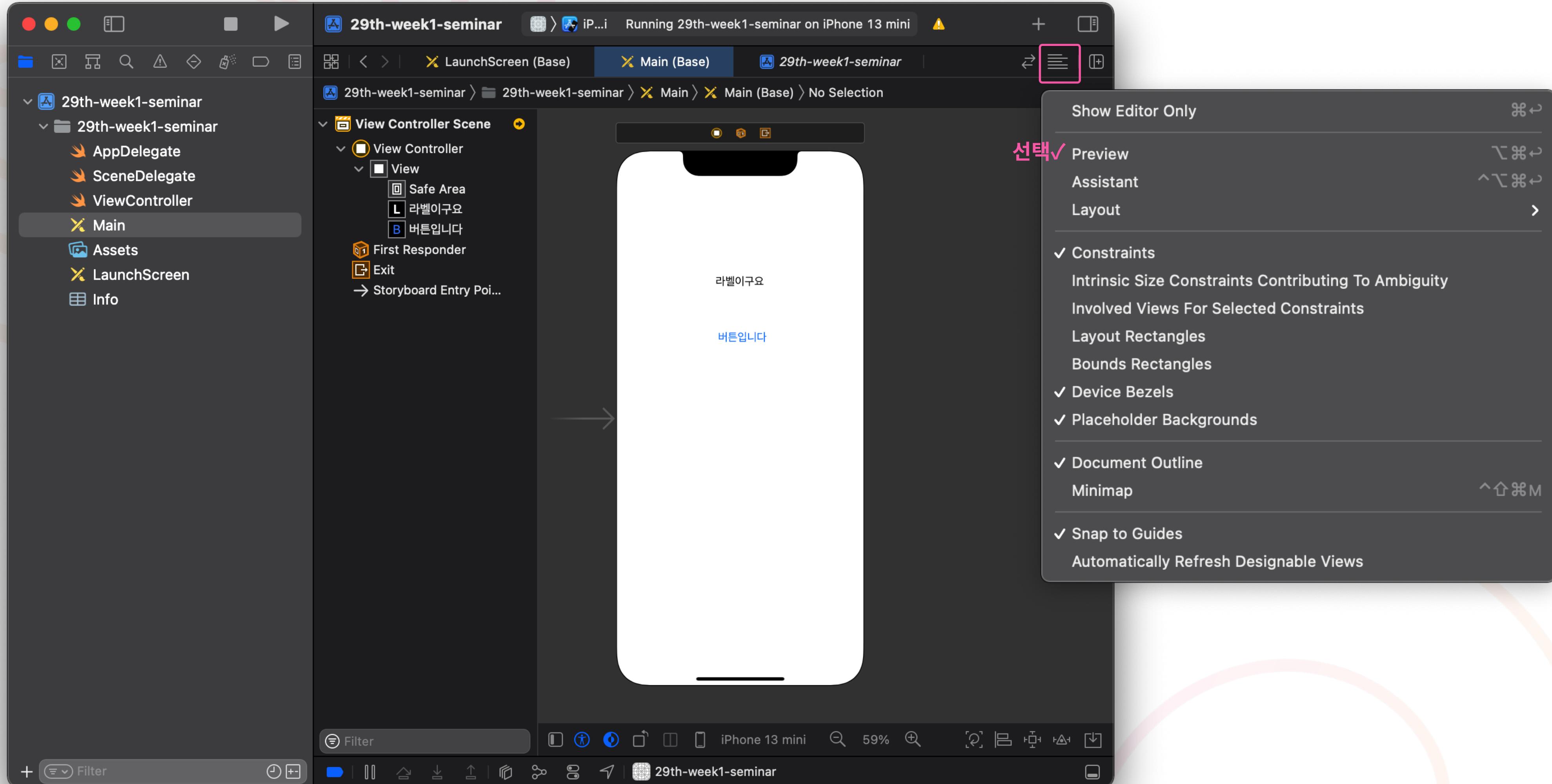


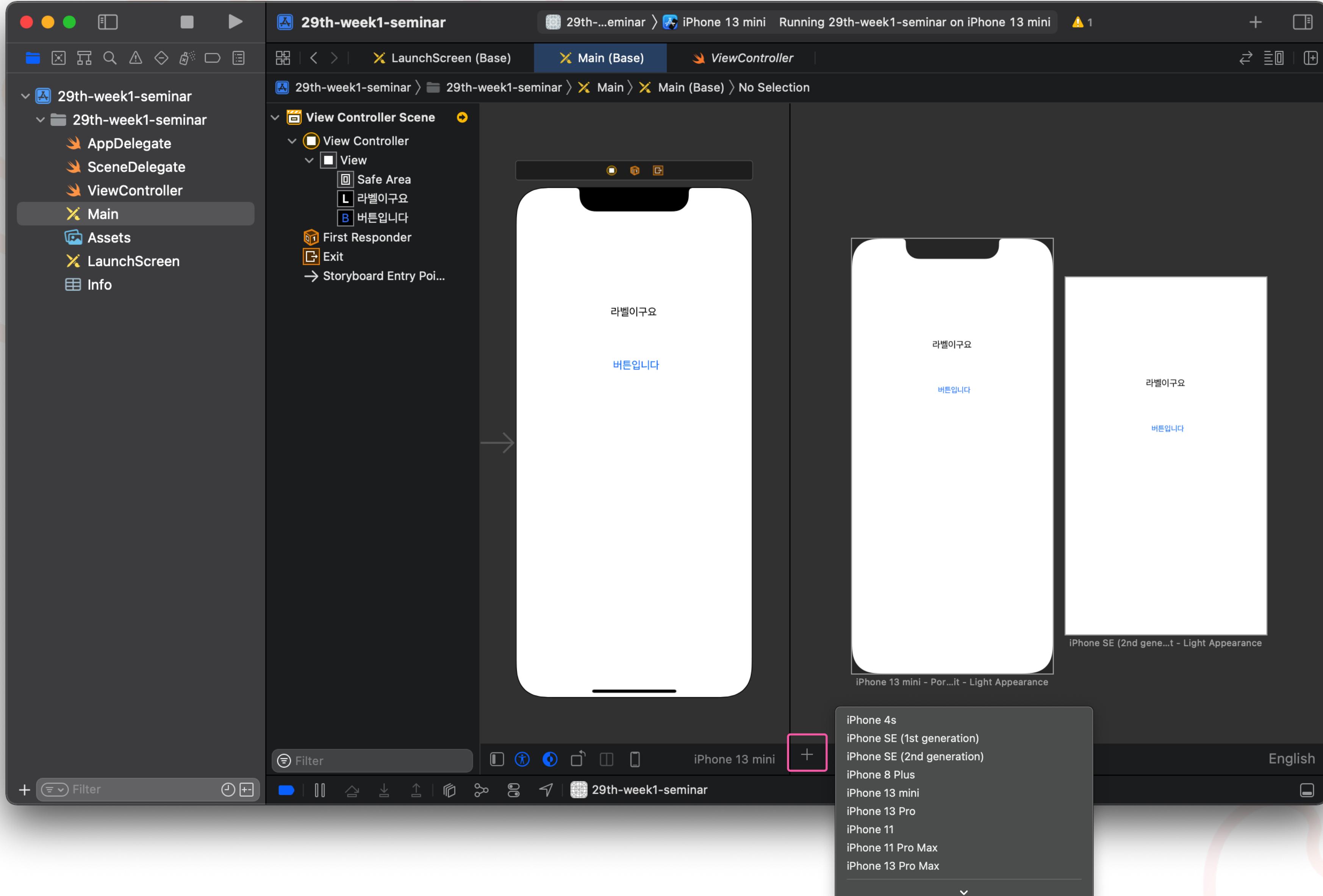
UI 구성하기 | 실습

버튼을 누르거나 command(⌘) + R을 눌러주세요



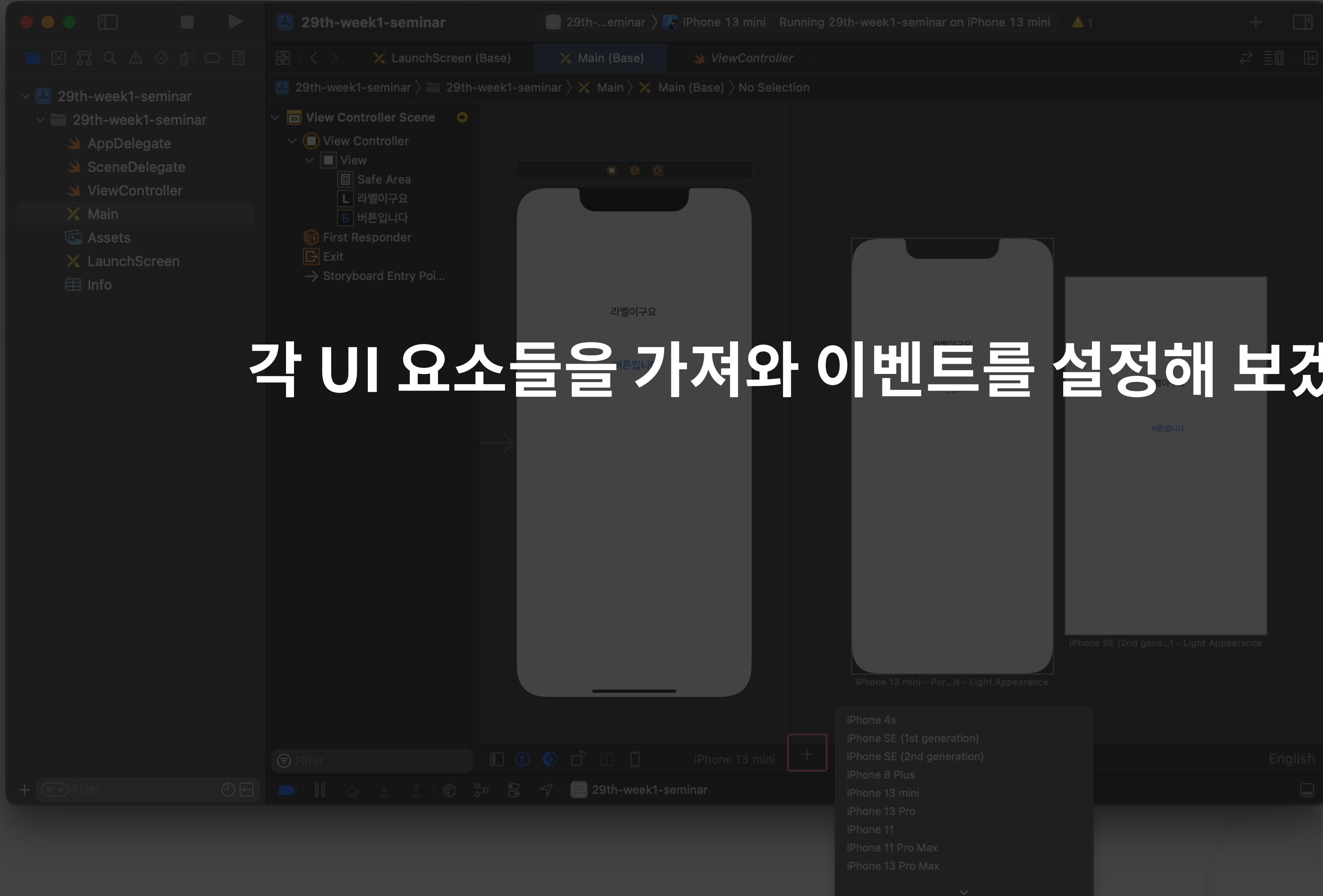
UI 구성하기 | 실습





시뮬레이터를
빌드하지 않더라도

Preview 기능을 활용해
각 기기에서 어떻게 보이는지
빠르게 파악이 가능

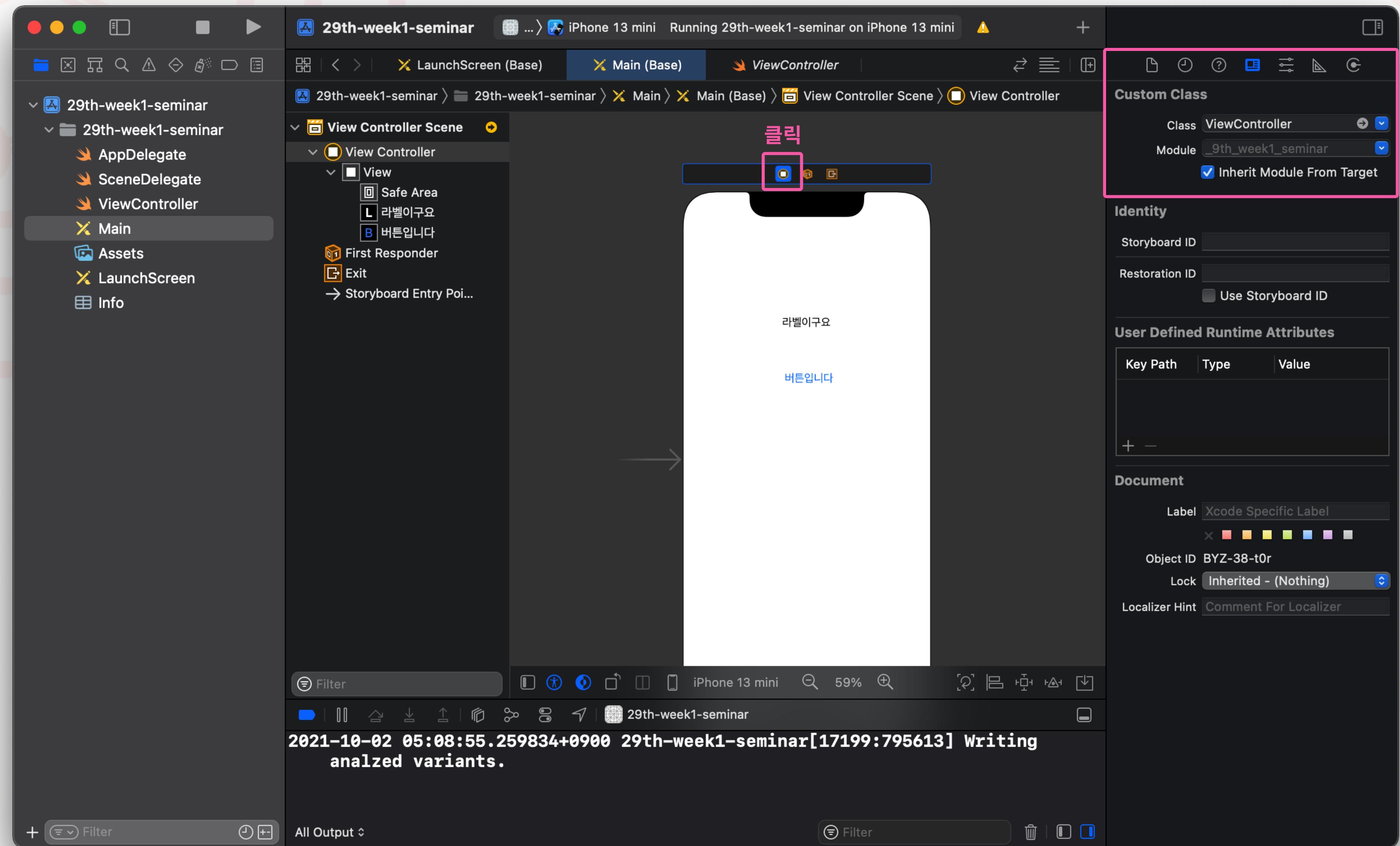


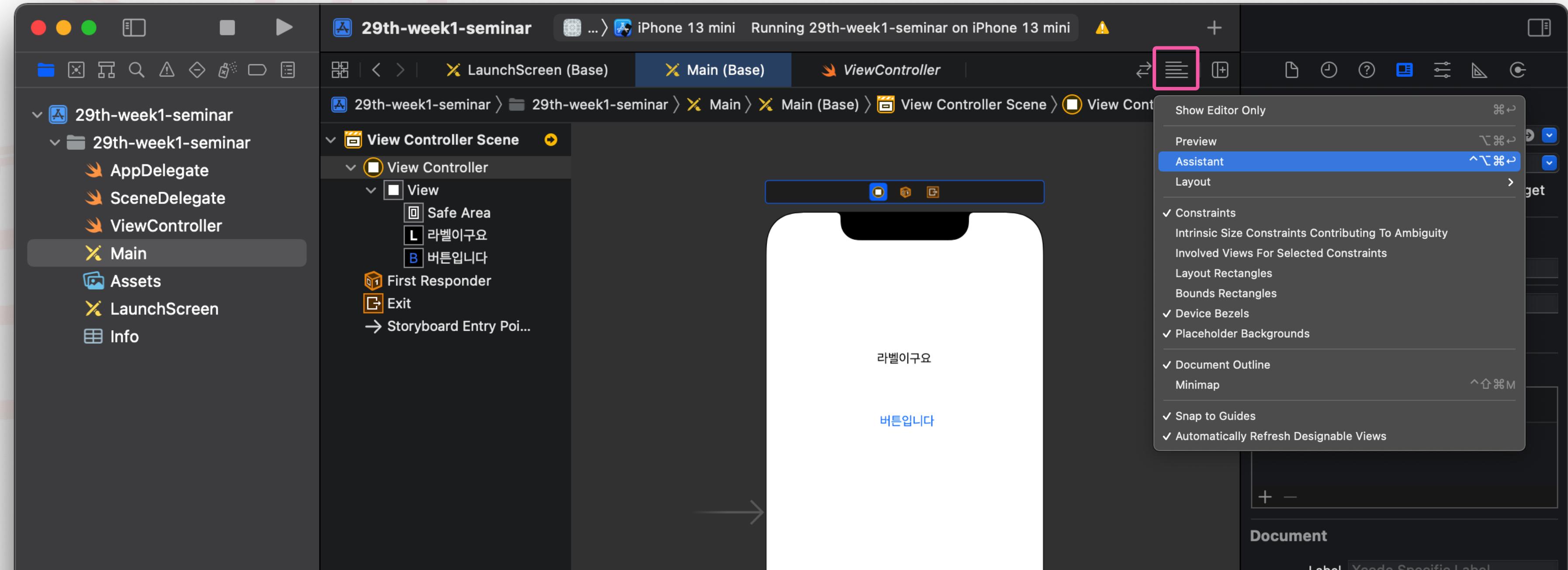
각 UI 요소들을 가져와 이벤트를 설정해 보겠습니다

시뮬레이터를
빌드하지 않더라도

Preview 기능을 활용해
각 기기에서 어떻게 보이는지
빠르게 파악이 가능

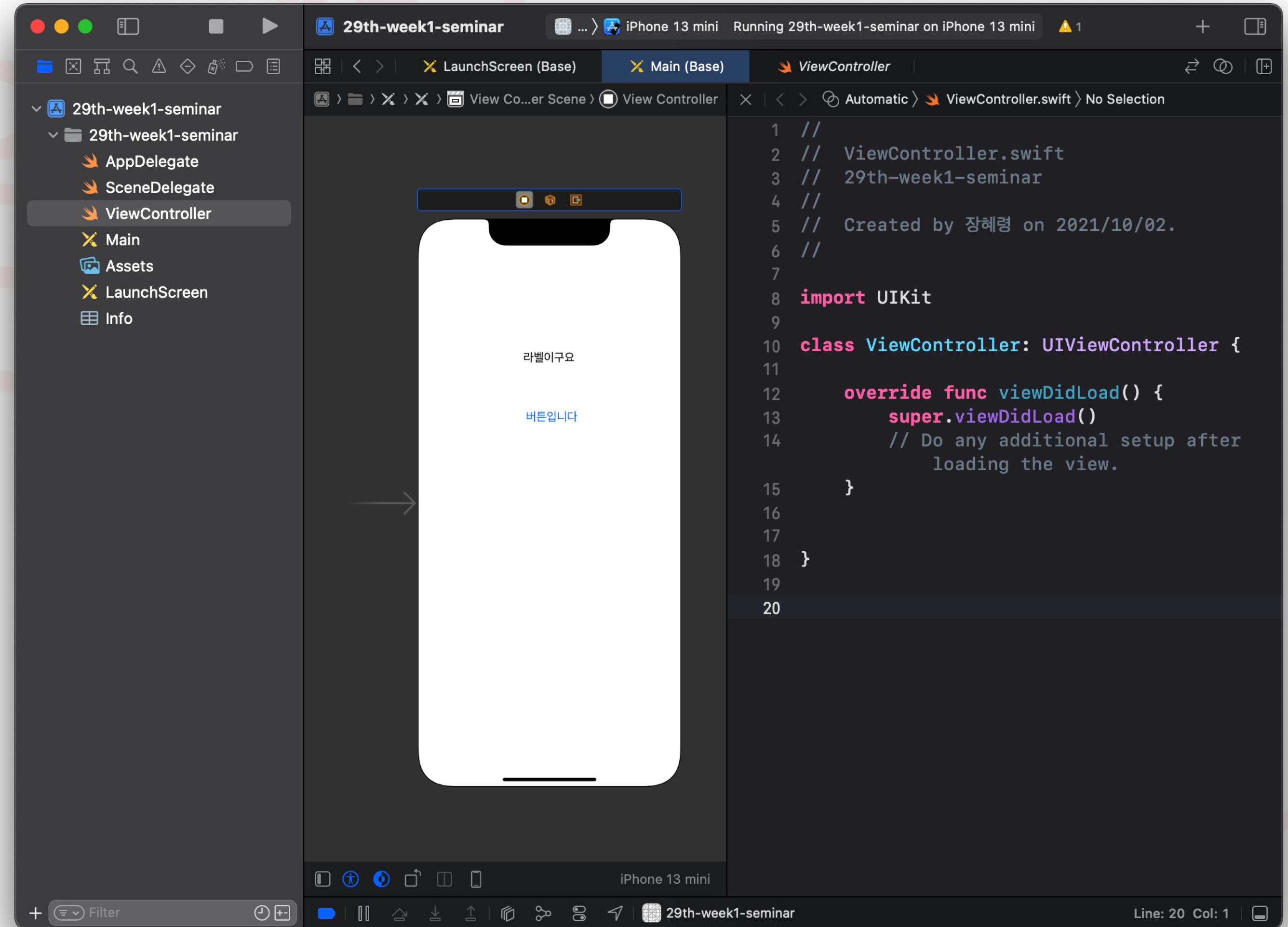
UI 구성하기 | 실습





Assistant Editor($\text{control}(\wedge)$ + $\text{option}(\text{\~})$ + $\text{command}(\text{\#})$ + enter)

UI 구성하기 | 실습



화면과 연결된
ViewController를 편집할 수
있는 창이 나타남

화면 상 요소 제어하기 위한 2가지 타입 (Interface Builder Annotation)

@IBOutlet

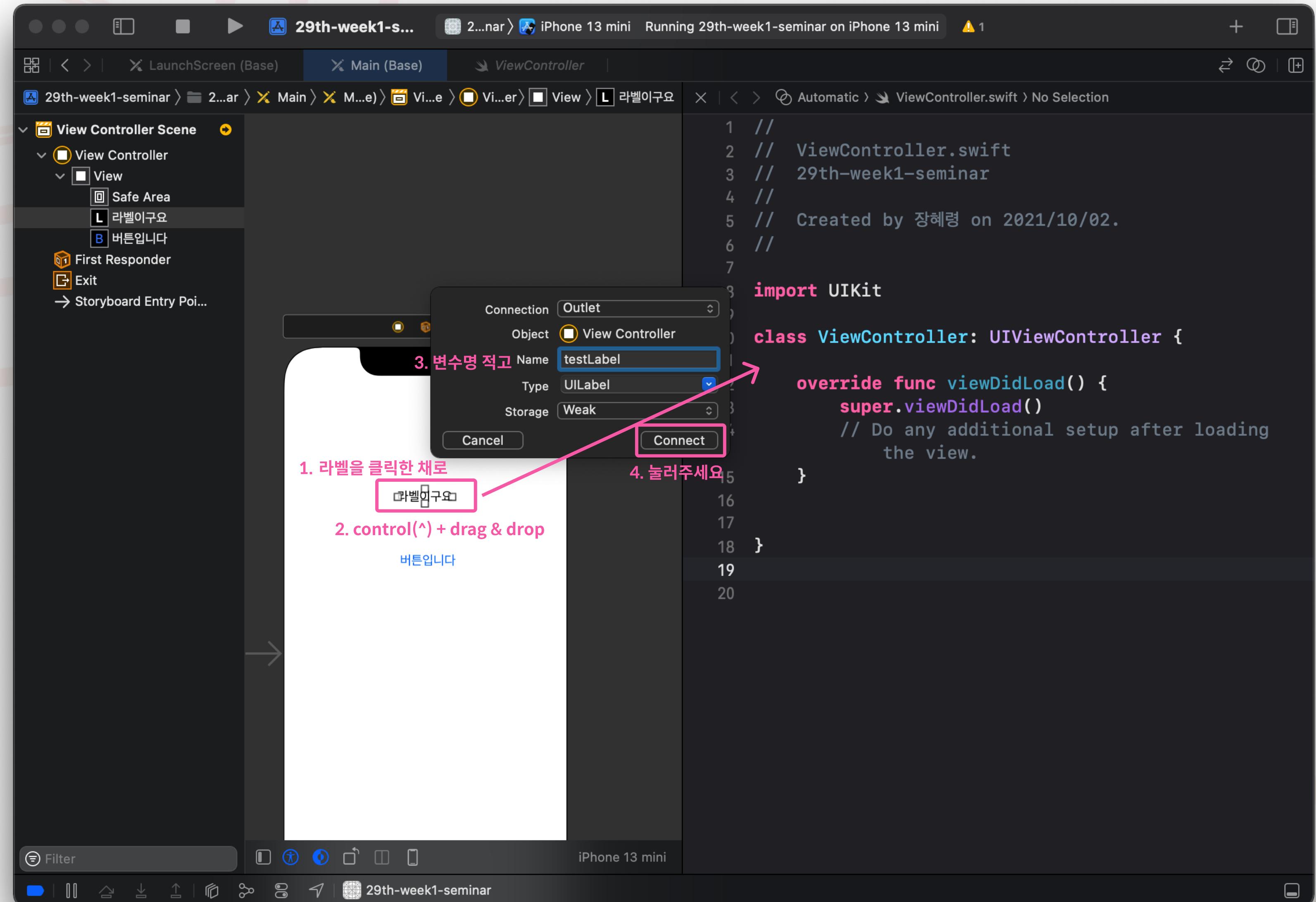
- 처리 결과를 View에 알리고 원하는 동작을 이끔
- View에 존재하는 요소와 Controller의 연결하기 위한 **변수**

@IBAction

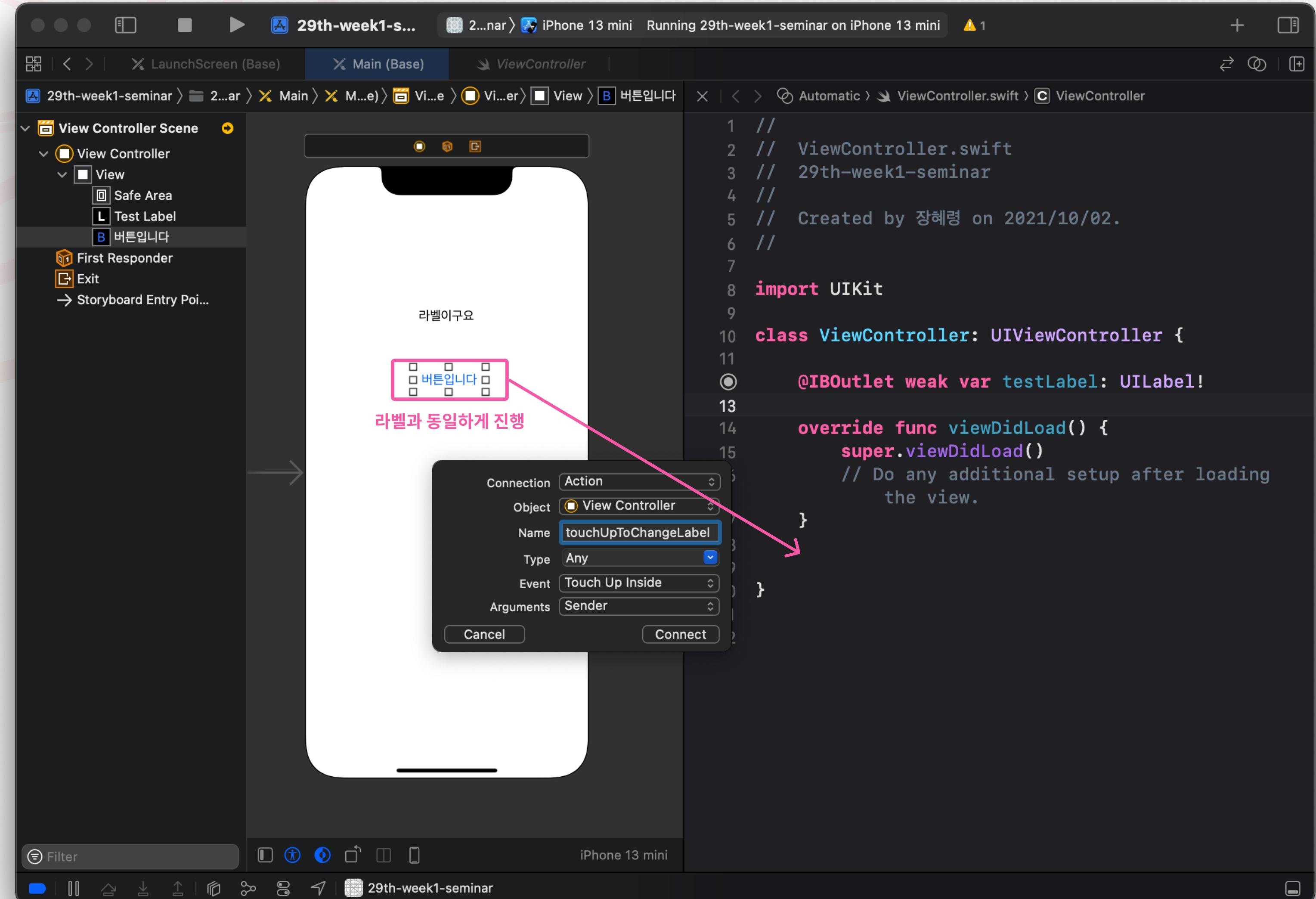
- 유저의 특정 이벤트를 감지해서 Controller에 알리는 역할
- 특정 이벤트 발생 시 실행 될 **동작**들을 정의

* IB는 Interface Builder의 약자

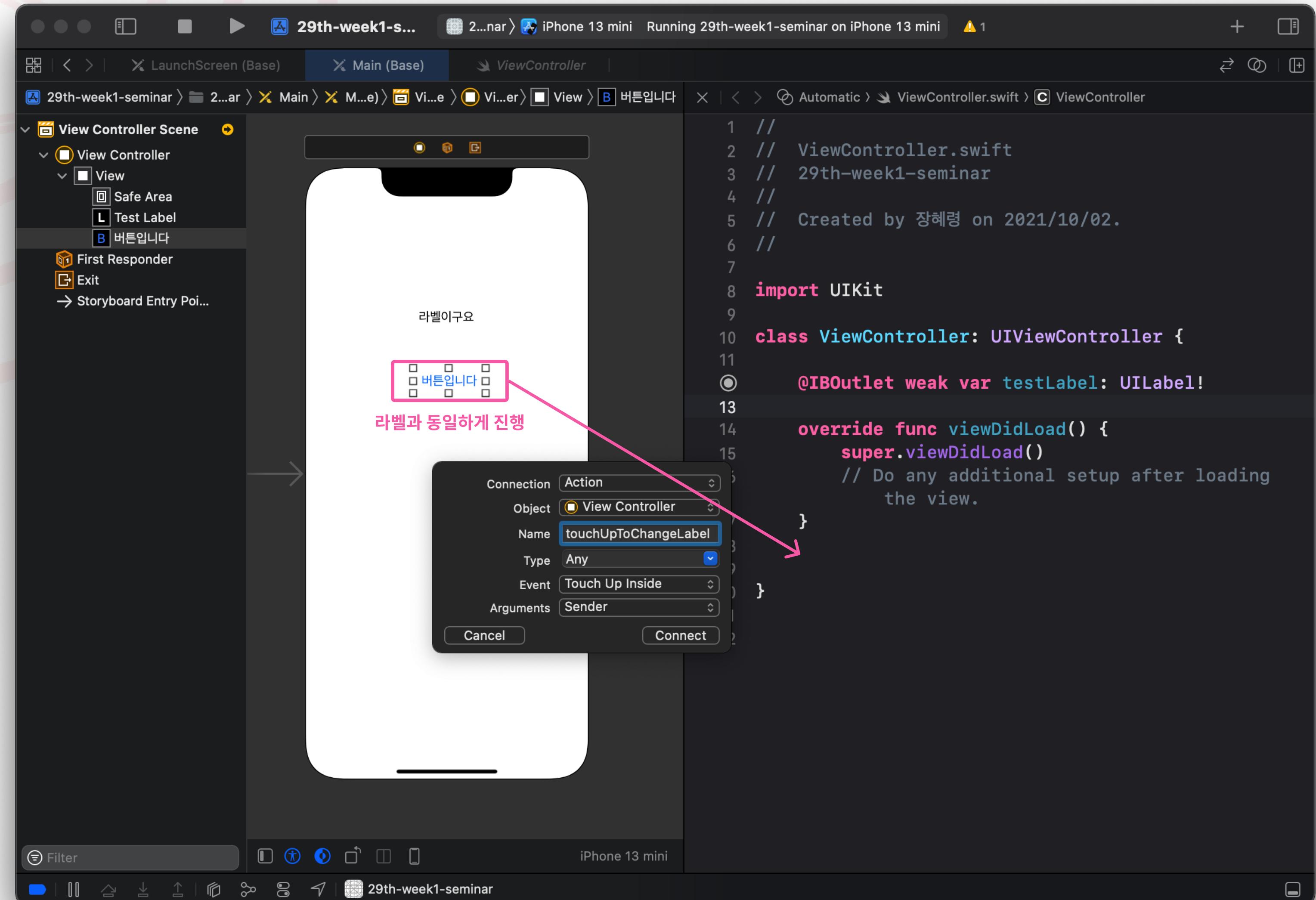
UI 구성하기 | 실습



UI 구성하기 | 실습



UI 구성하기 | 실습



```
8 import UIKit
9
10 class ViewController: UIViewController {
11
12     @IBOutlet weak var testLabel: UILabel!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view.
17     }
18
19     @IBAction func touchUpToChangeLabel(_ sender: Any) {
20
21     }
22 }
```

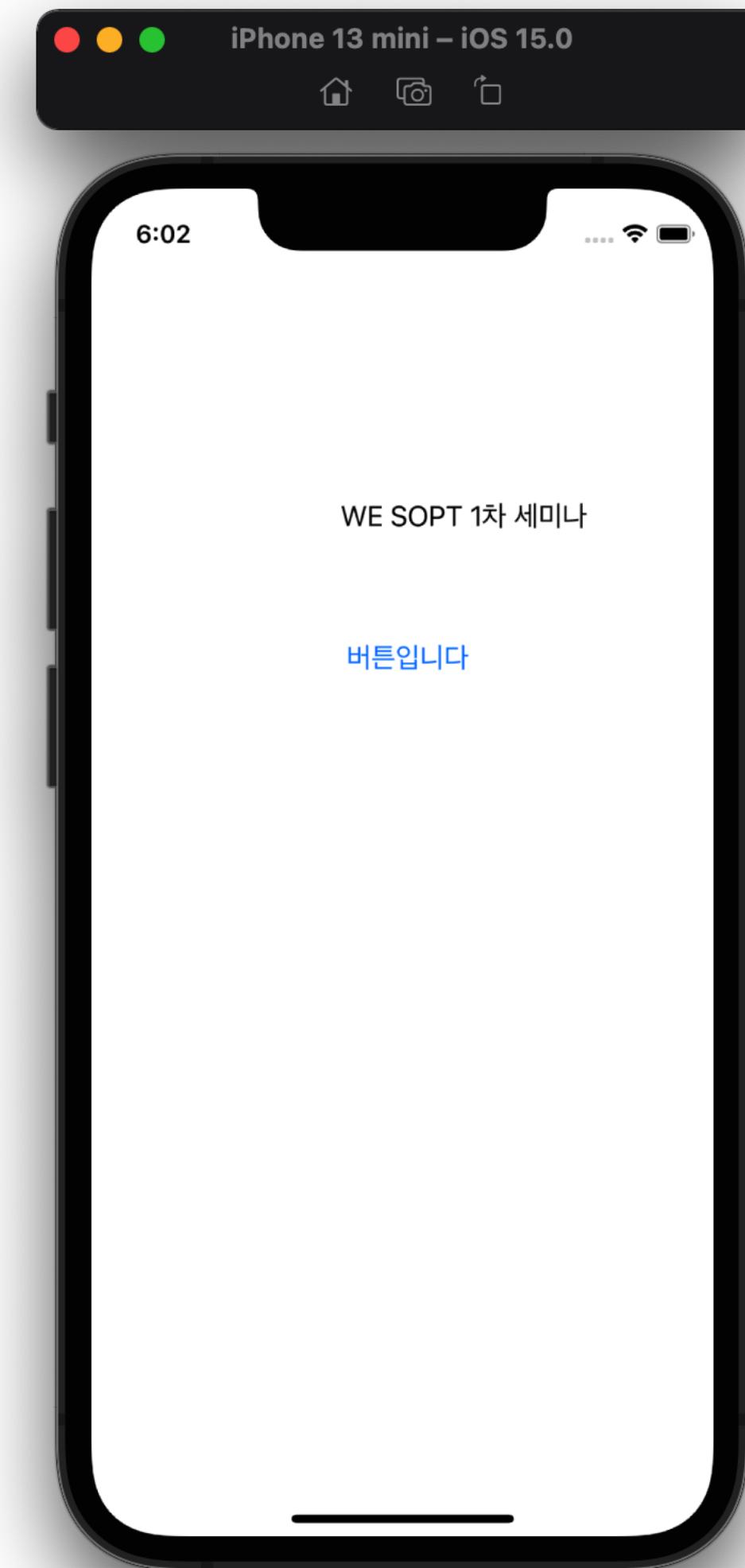
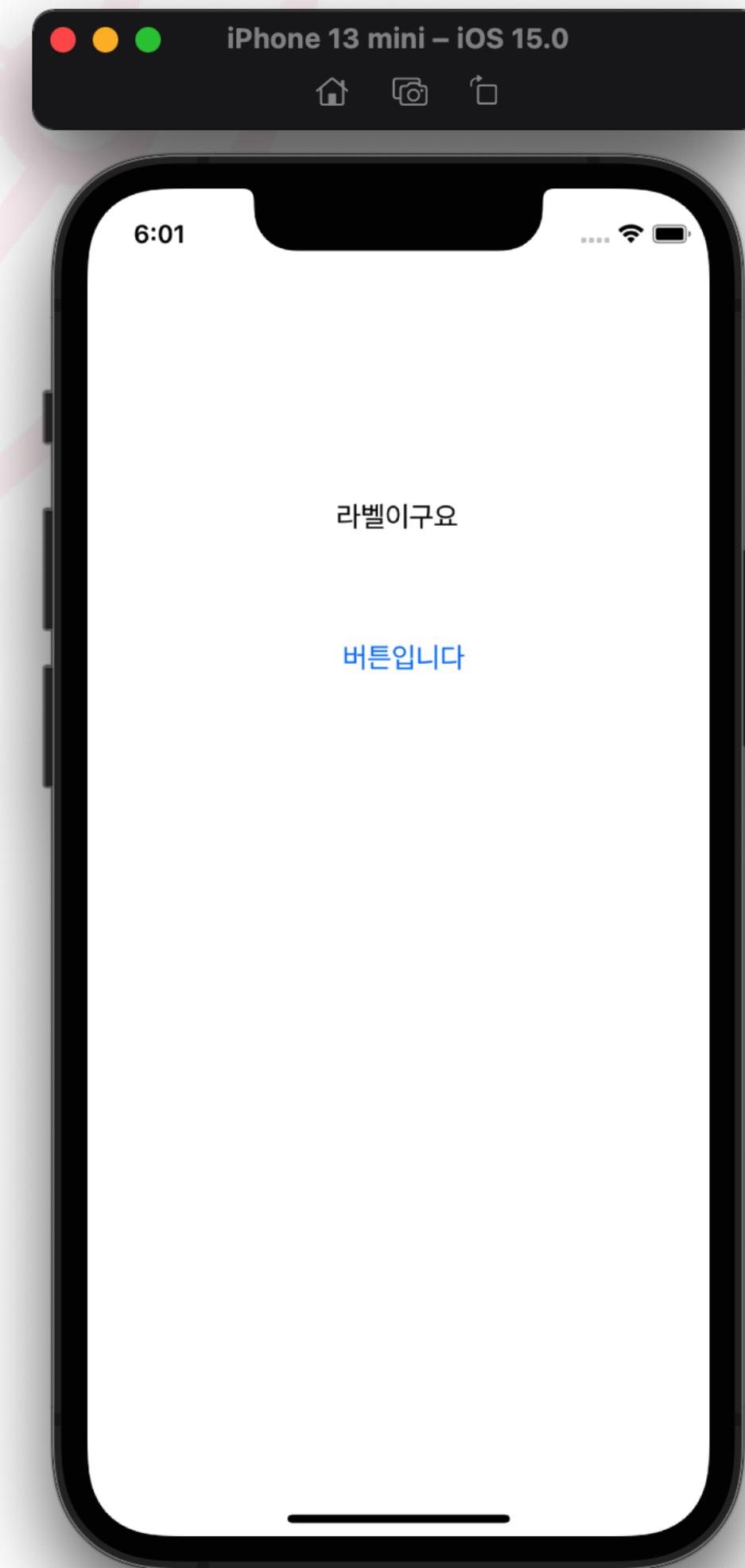
Storyboard에 넣었던 Label과 연결

Storyboard에 넣었던 Button의 touchUpInside라는 이벤트 감지
* touchUpInside : 터치한 컴포넌트에서 손을 뗐을 때 이벤트 발생

```
@IBAction func touchUpInside(_ sender: Any) {  
    testLabel.text = "WE SOPT 1차 세미나"  
    testLabel.sizeToFit() // 라벨 크기를 글자에 맞게 변경해주는 메소드  
}
```

- 버튼에 touchUpInside 이벤트가 발생하면
- 버튼과 연결한 touchUpToChangeLabel(_) 함수가 불림
- 이 함수 내부에서 실행 될 동작을 코드로 정의

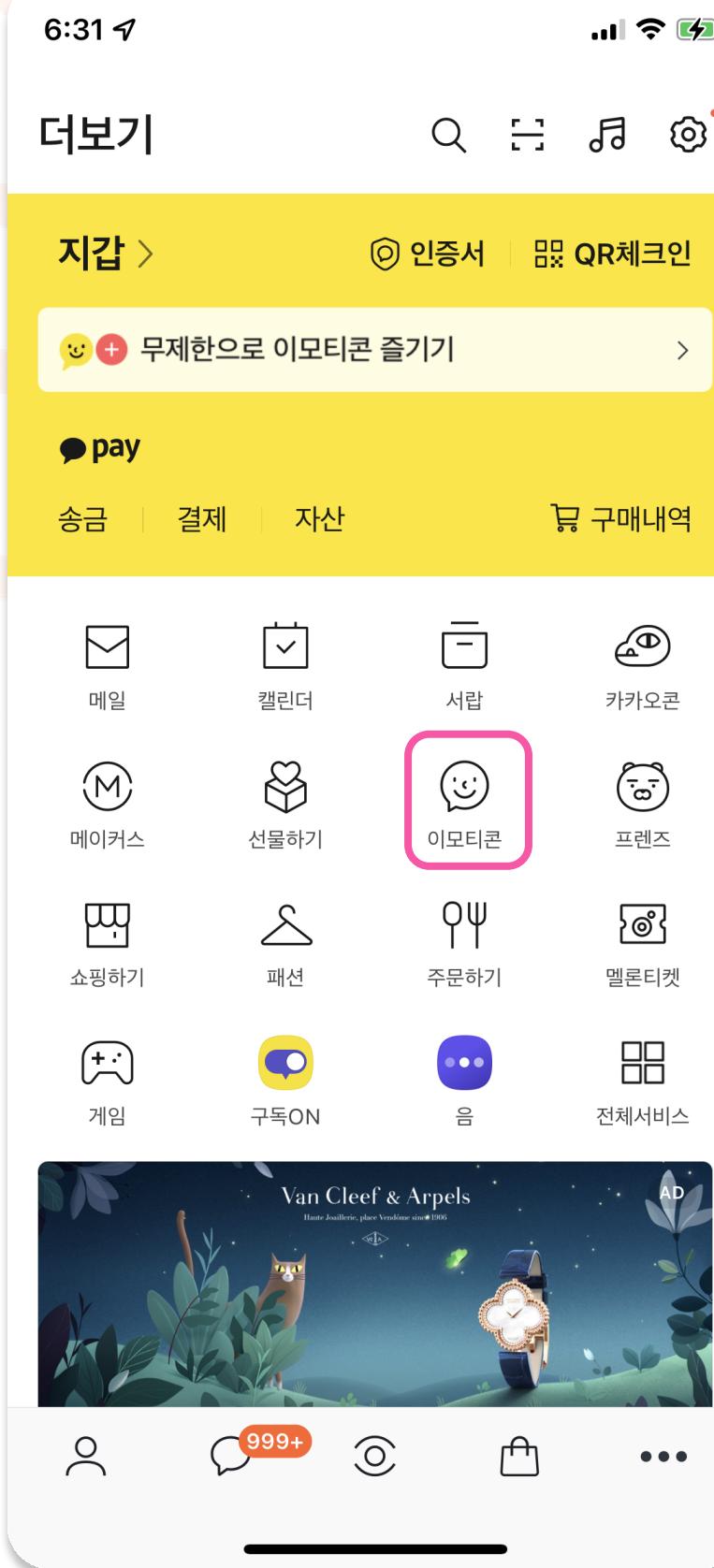
○ UI 구성하기 | 실습



04

화면 전환

수많은 APP들을 사용하면서 다양한 화면을 보게되는데
현재 보던 화면에서 다른 화면으로 이동하기 위해서는 **화면 전환**이 필요합니다!



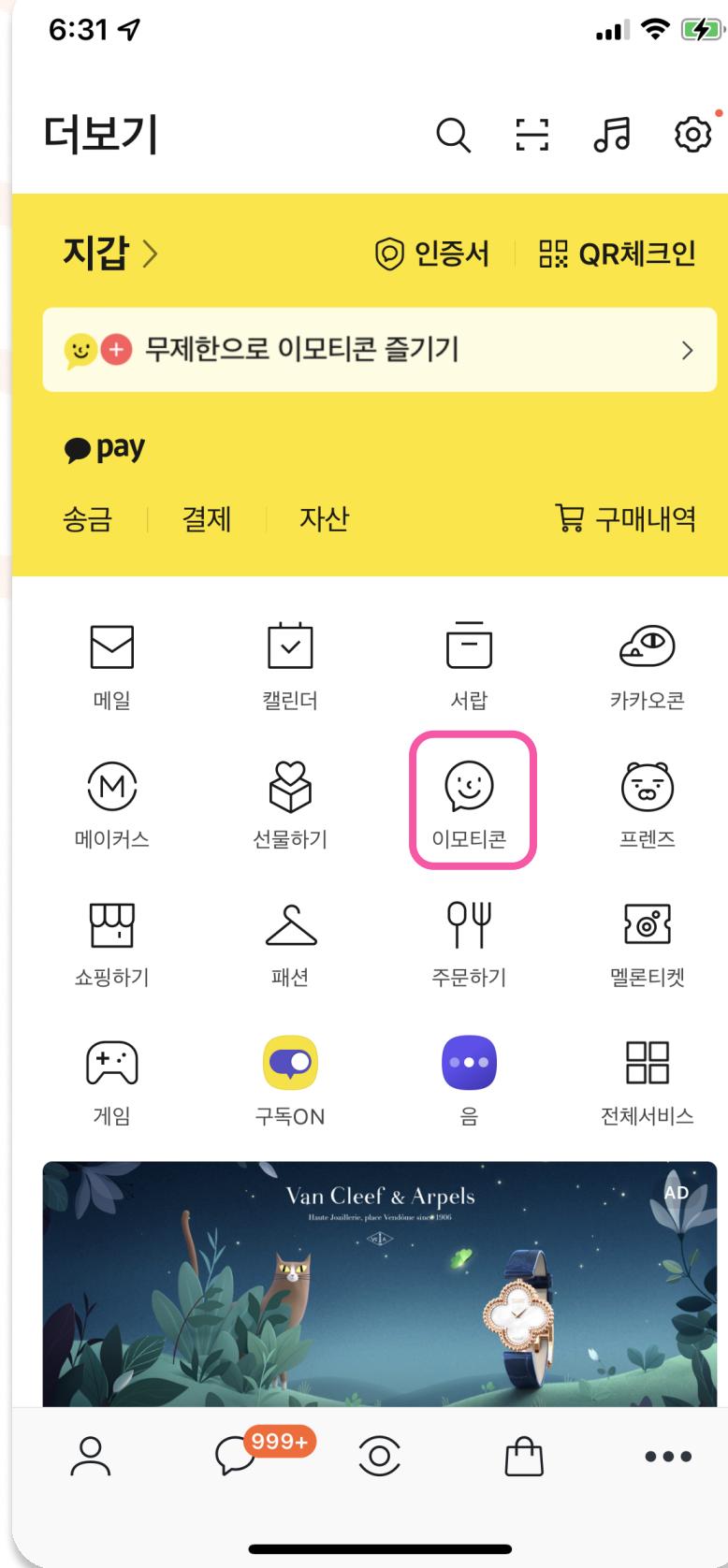
위로 속 올라오면서 하단에 있던
친구, 채팅 아이콘들이 사라지고
새로운 화면이 나왔습니다



이번에는 오른쪽으로 속 화면이 생기고
또 다른 화면이 나왔습니다



수많은 APP들을 사용하면서 다양한 화면을 보게되는데
현재 보던 화면에서 다른 화면으로 이동하기 위해서는 **화면 전환**이 필요합니다!



모달(Modal)

위로 속 올라오면서 하단에 있던
친구, 채팅 아이콘들이 사라지고
새로운 화면이 나왔습니다



네비게이션 (Navigation)

이번에는 오른쪽으로 속 화면이 생기고
또 다른 화면이 나왔습니다

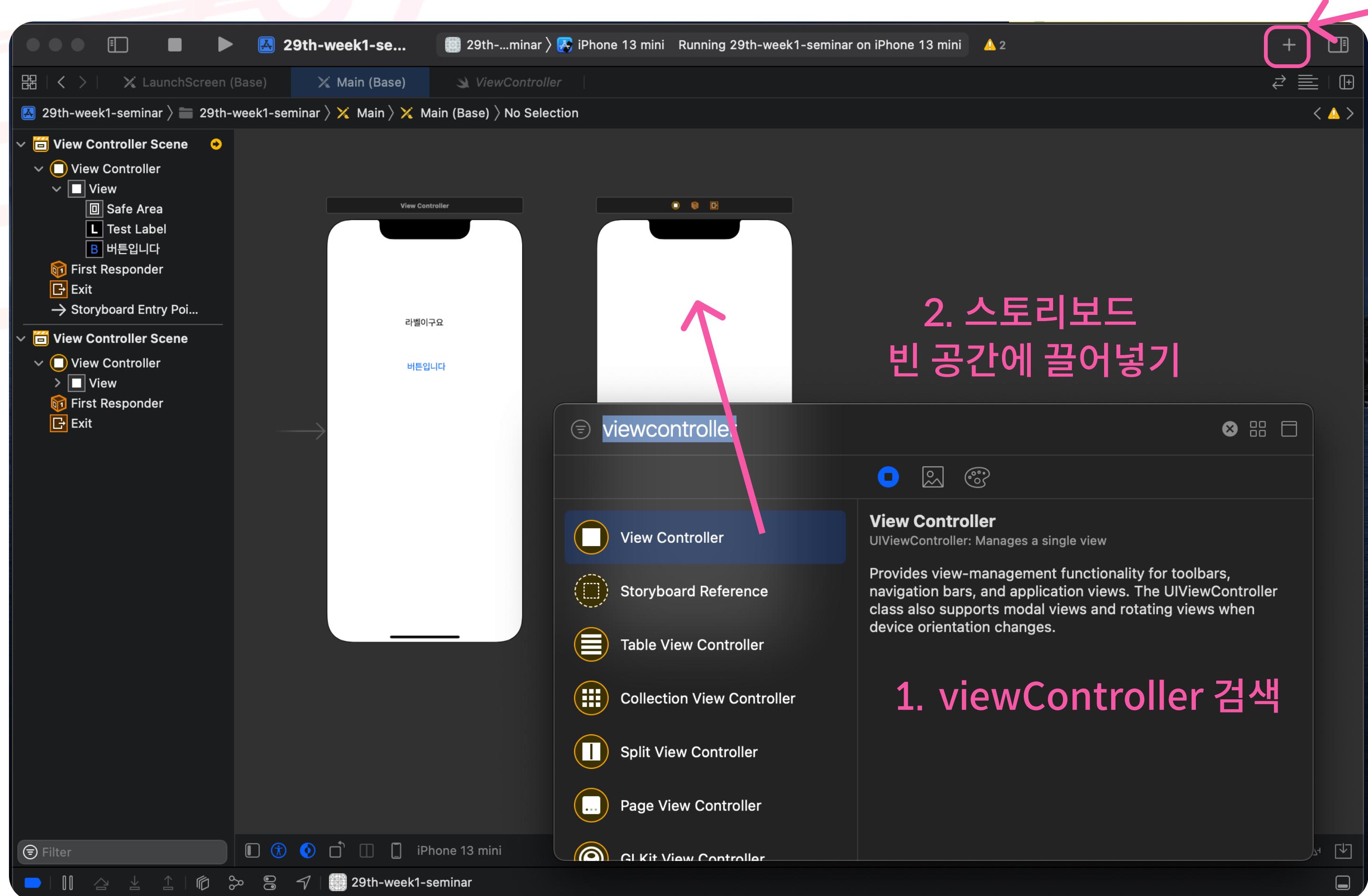


○ 화면 전환 | 모달(Modal)

● 모달(Modal)

- 사용자의 이목을 끌기 위해 사용하는 화면 전환 기법
- 화면을 전환한다기 보다는 이목을 집중해야 하는 화면을 다른 화면 위로 띄워(Presenting) 표현 하는 방식
- 흐름을 이어지는 컨텐츠를 담기보다는 흐름이 끊어져서 딱 눈에 들어와야 하는 컨텐츠를 담는데 사용

○ 화면 전환 | 모달(Modal)



2. 스토리보드 빈 공간에 끌어넣기

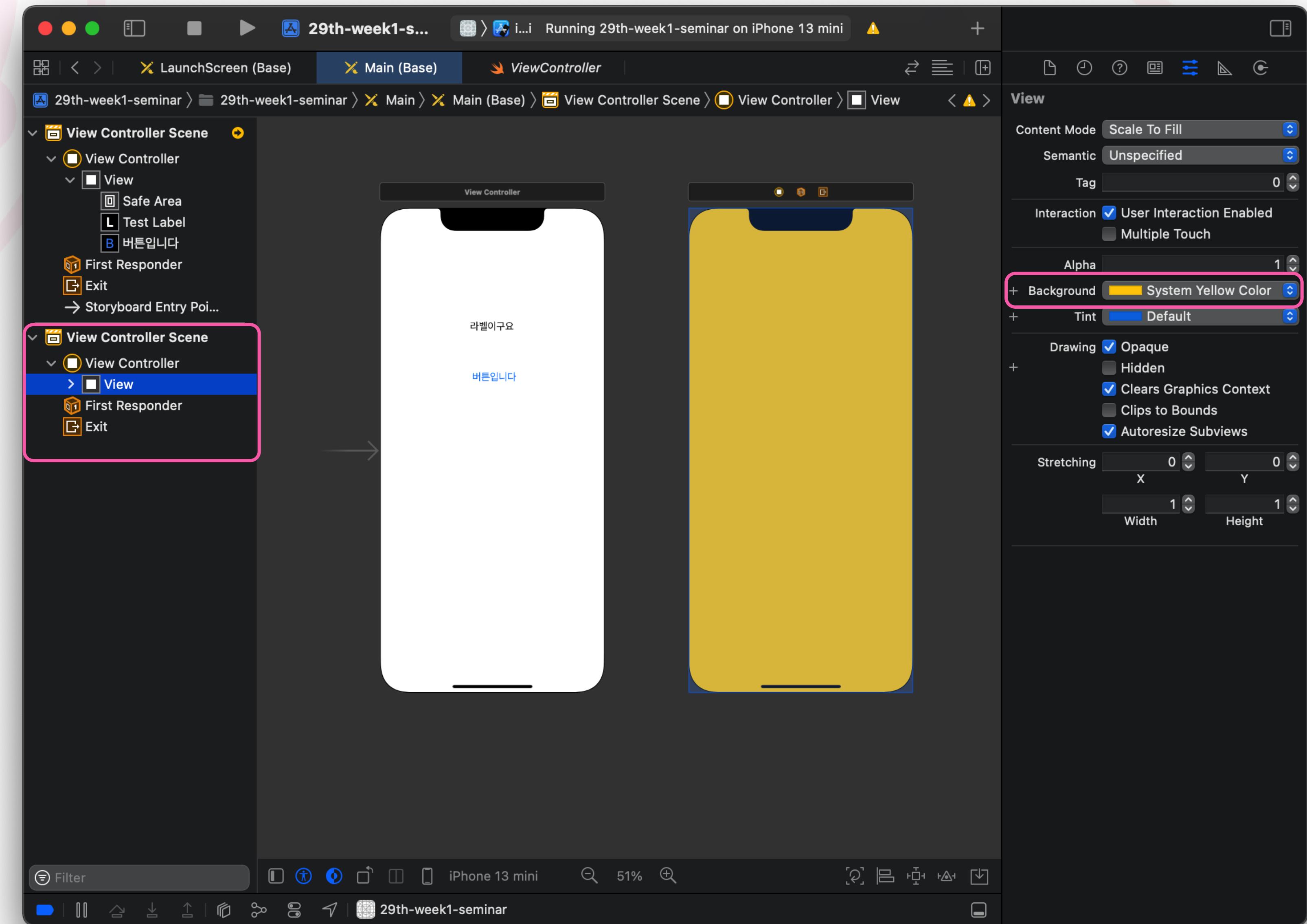
1. viewController 검색

0. 클릭 또는
shift (↑) + command (⌘) + L

○ 화면 전환 | 모달(Modal)

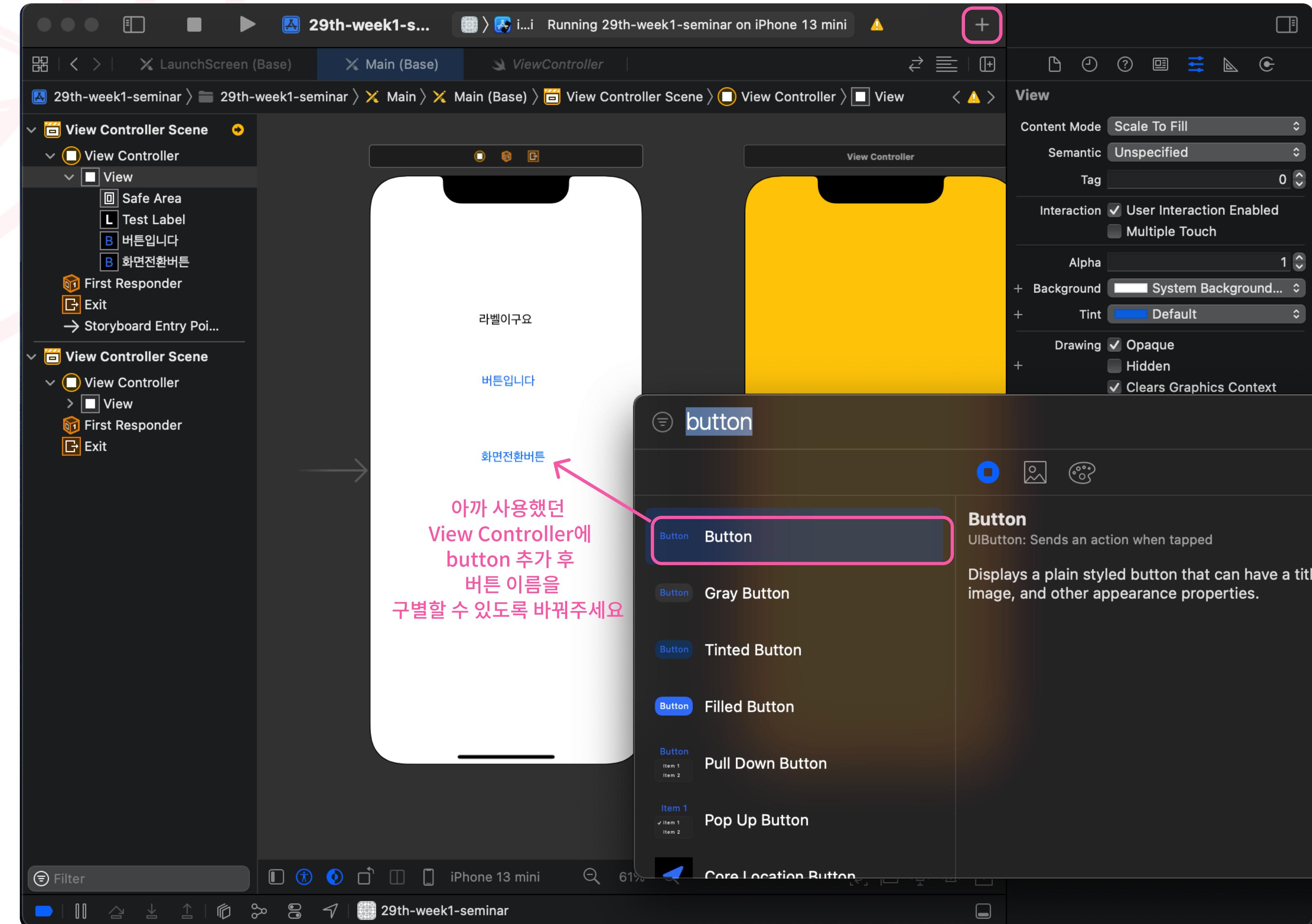
화면전환이 눈에 잘 띠도록

방금 만든 View controller에서
View를 선택한 후
우측 Inspector 메뉴에서
Background 속성을 변경해주세요
(색상은 자유롭게)



화면 전환 | 모달(Modal)

클릭 또는
shift(⇧) + command(⌘) + L

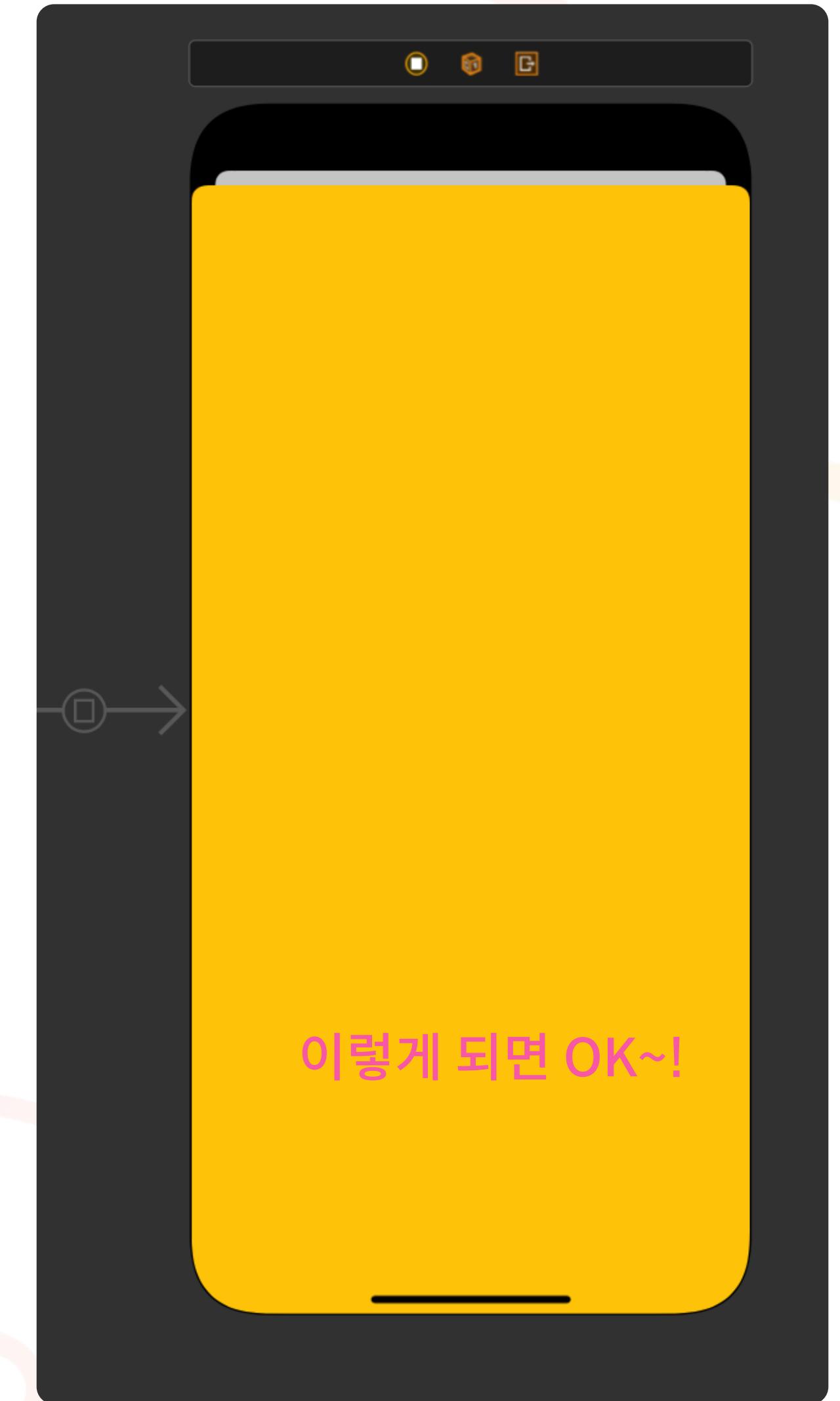


○ 화면 전환 | 모달(Modal)



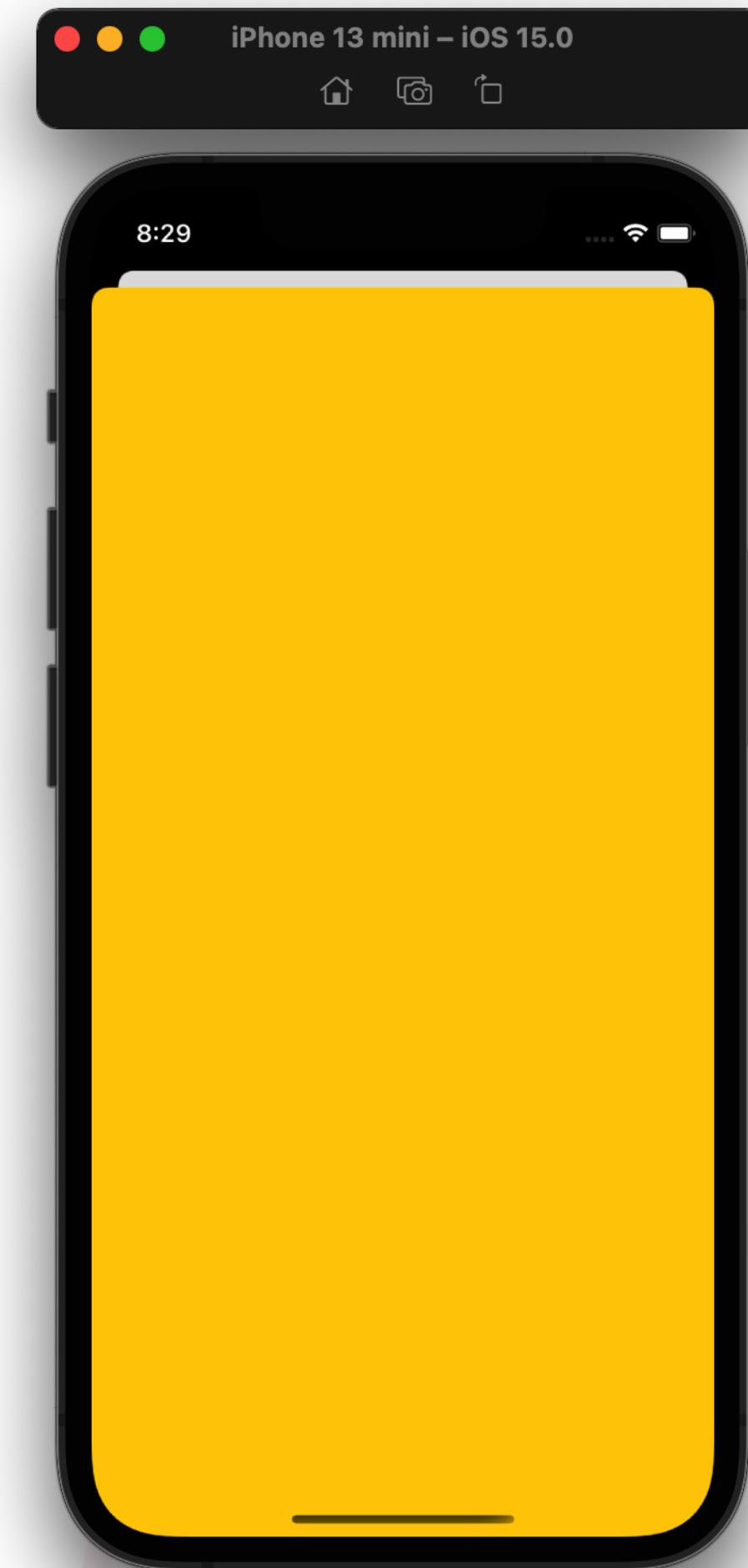
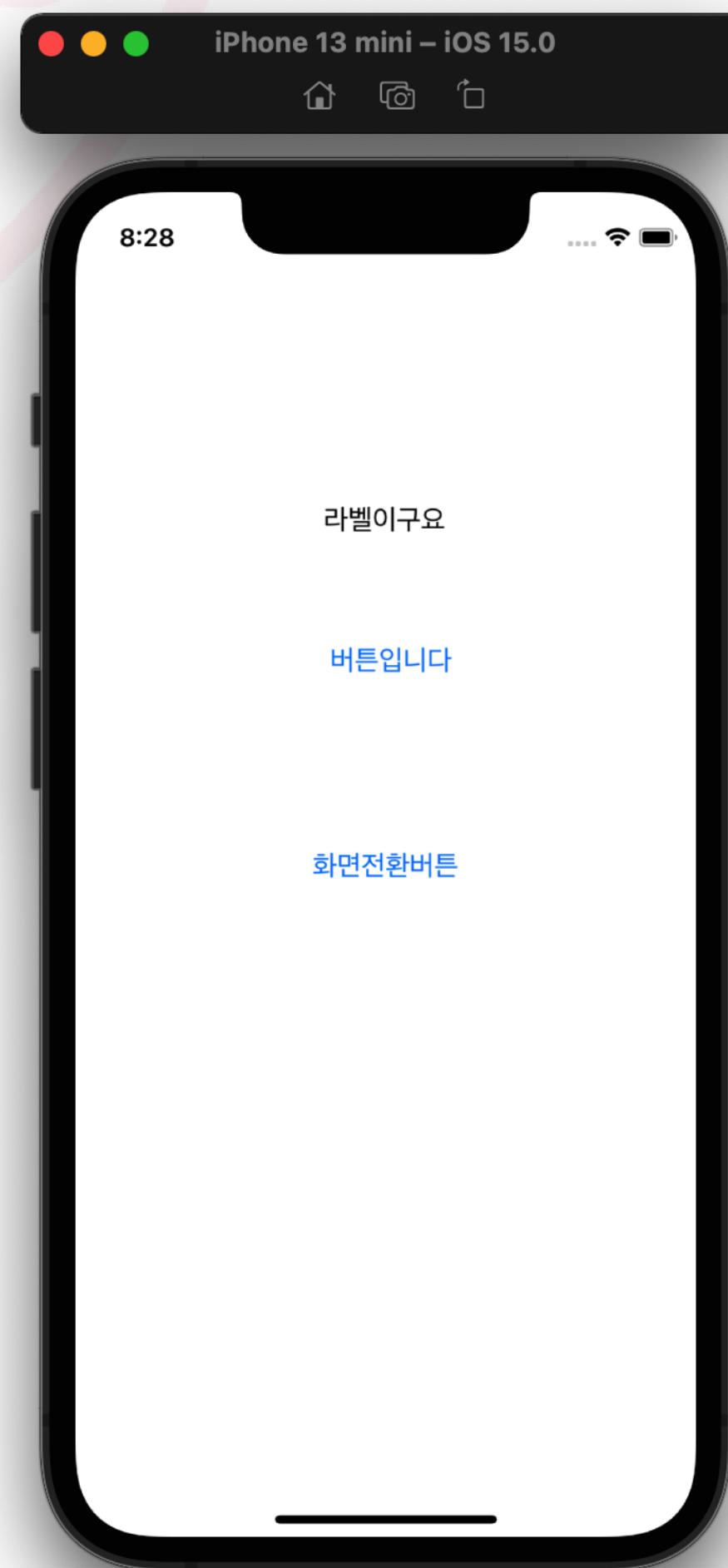
Action Segue
Show
Show Detail
Present Modally
Present AsPopover
Custom
Non-Adaptive Action Segue
Push (deprecated)
Modal (deprecated)

3. Present Modally 클릭



○ 화면 전환 | 모달(Modal)

빌드해서 확인해보세요! (command + R)



○ 화면 전환 | 모달(Modal)



세그웨이(Segue)로 모달을 구현한 방식

* 세그웨이(Segue)

iOS에서 화면 전환 방식에 사용되는 방법

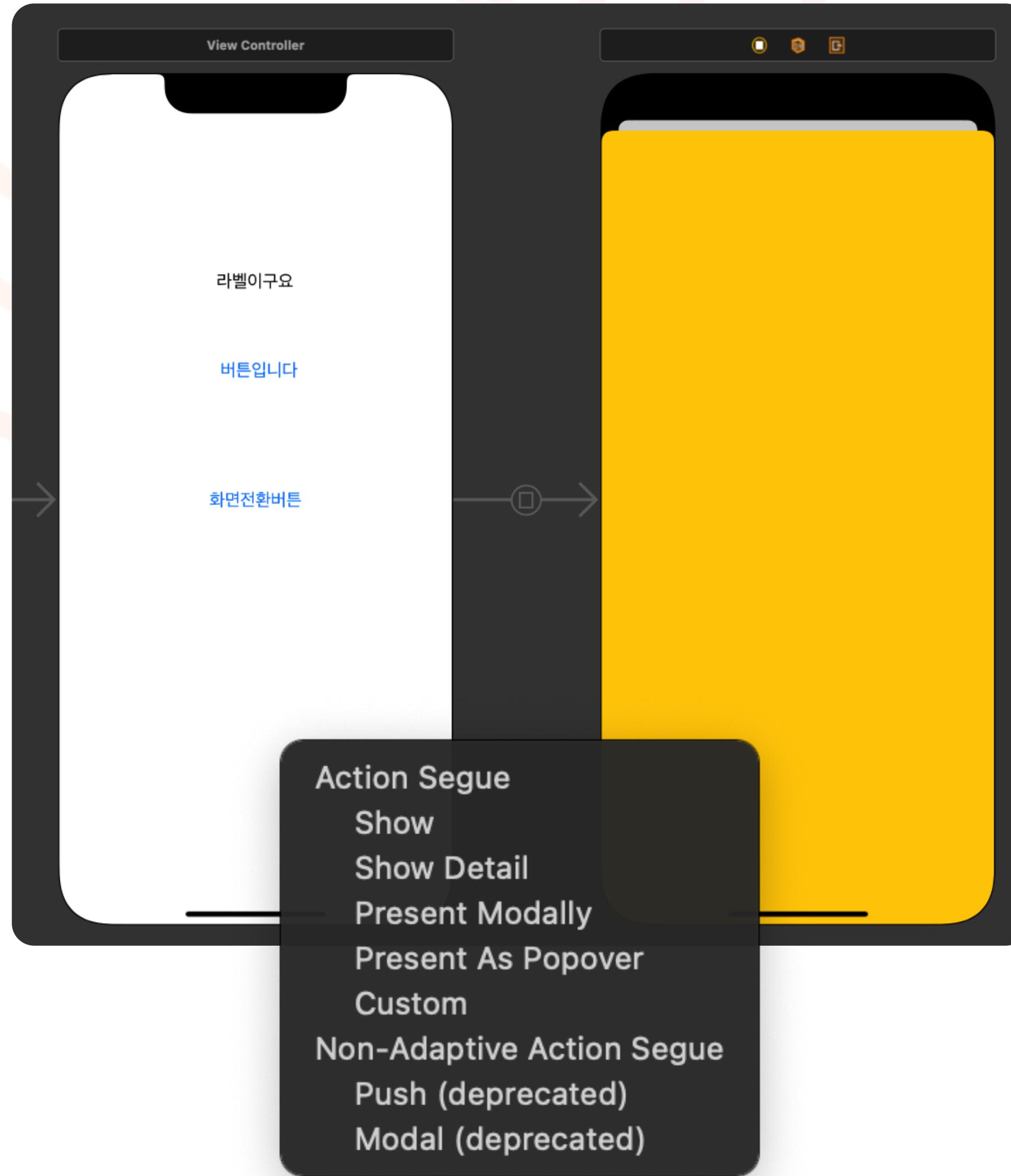
스토리보드를 통해 출발지와 목적지를 직접 지정하는 방식

을 세그웨이(segue)을 이용한 화면 전환이라고 하고,

두 개의 ViewController 사이 연결된

화면 전환 객체를 세그웨이이라고 합니다.

○ 화면 전환 | 모달(Modal)



세그웨이(Segue)로 모달을 구현한 방식

* 세그웨이(Segue)

iOS에서 화면 전환 방식에 사용되는 방법

스토리보드를 통해 출발지와 목적지를 직접 지정하는 방식을

세그웨이(segue)을 이용한 화면 전환이라고 하고,
두 개의 ViewController 사이 연결된 화면 전환 객체를 세그웨이라고 함.

○ 화면 전환 | 모달(Modal)

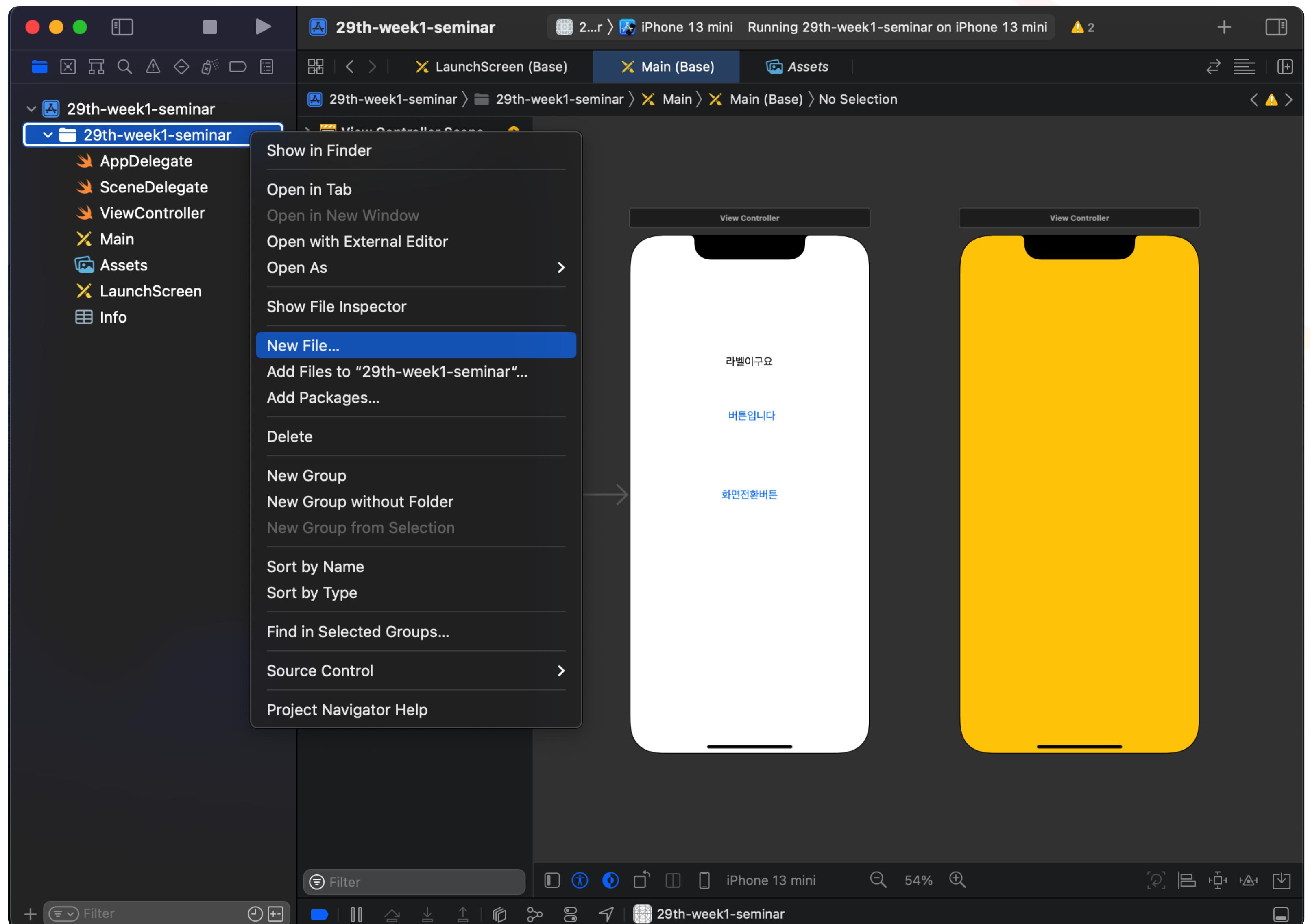


코드로 모달 구현하기

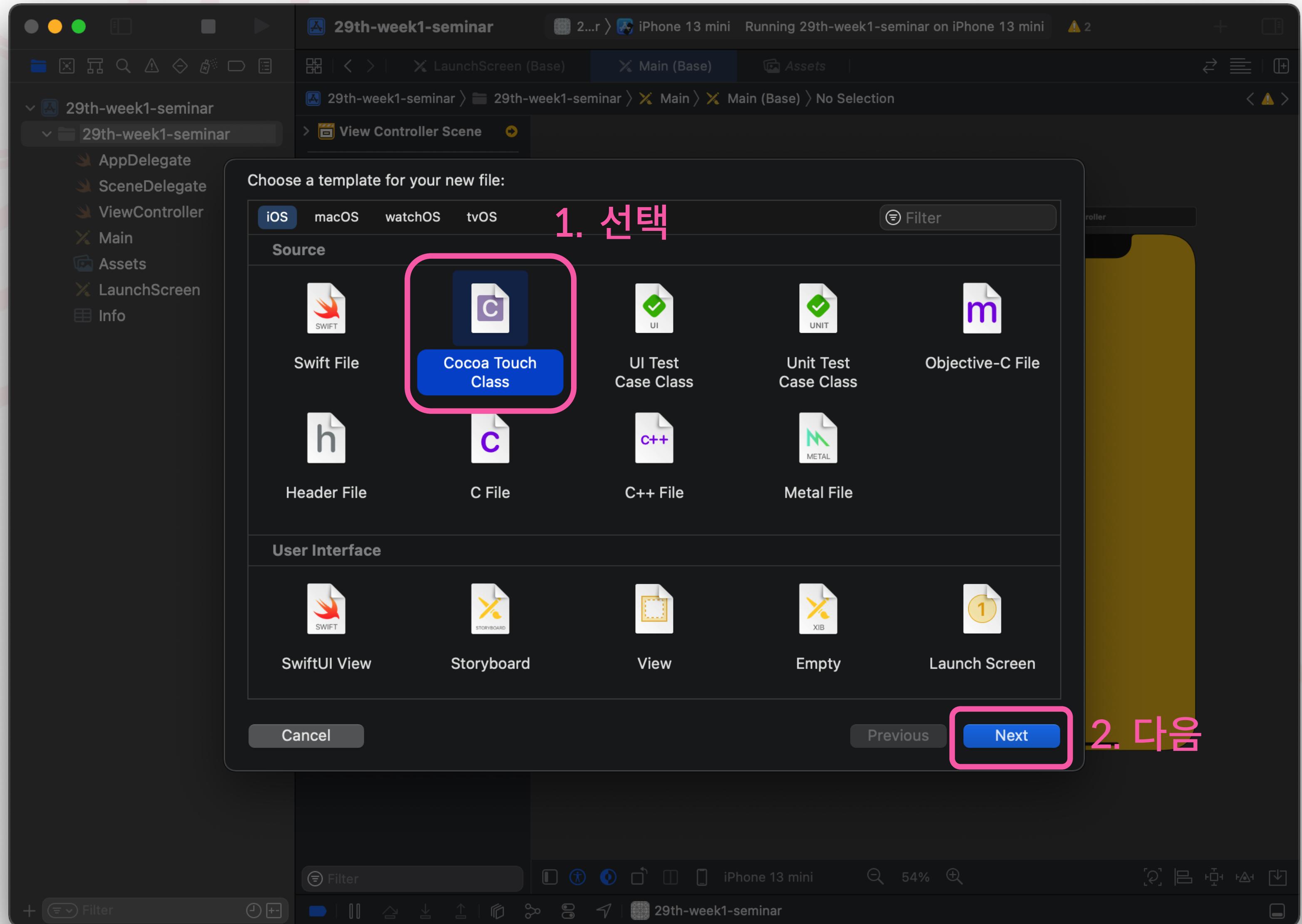
하기 전에는 먼저 세그를 삭제해주세요!
코드방식과 세그 방식을
동시에 사용하면 에러가 발생합니다

○ 화면 전환 | 모달(Modal)

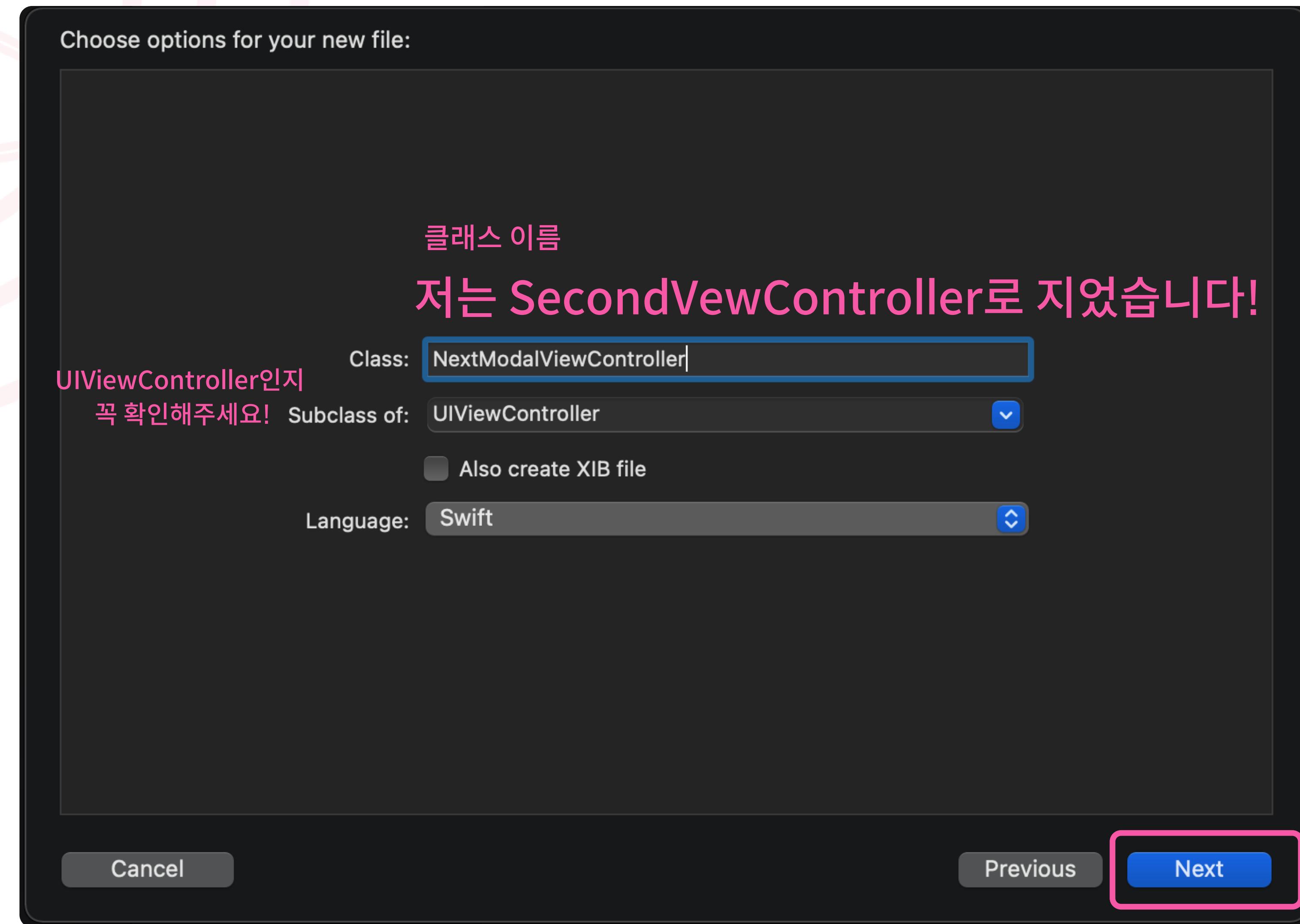
프로젝트 폴더를 선택하여
마우스 오른쪽 버튼을 누르거나,
command + N을 눌러
새로운 파일을 생성해주세요



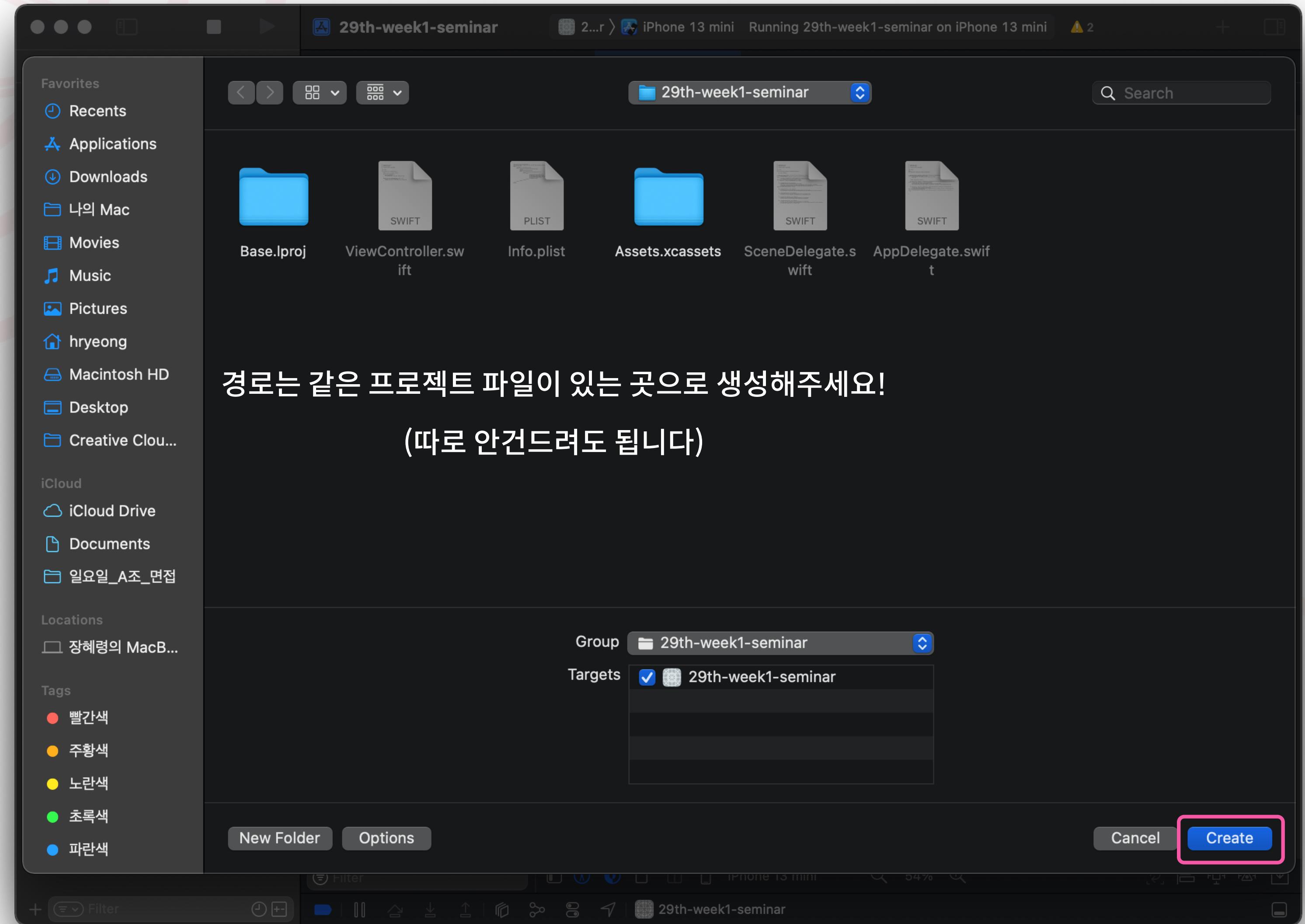
○ 화면 전환 | 모달(Modal)



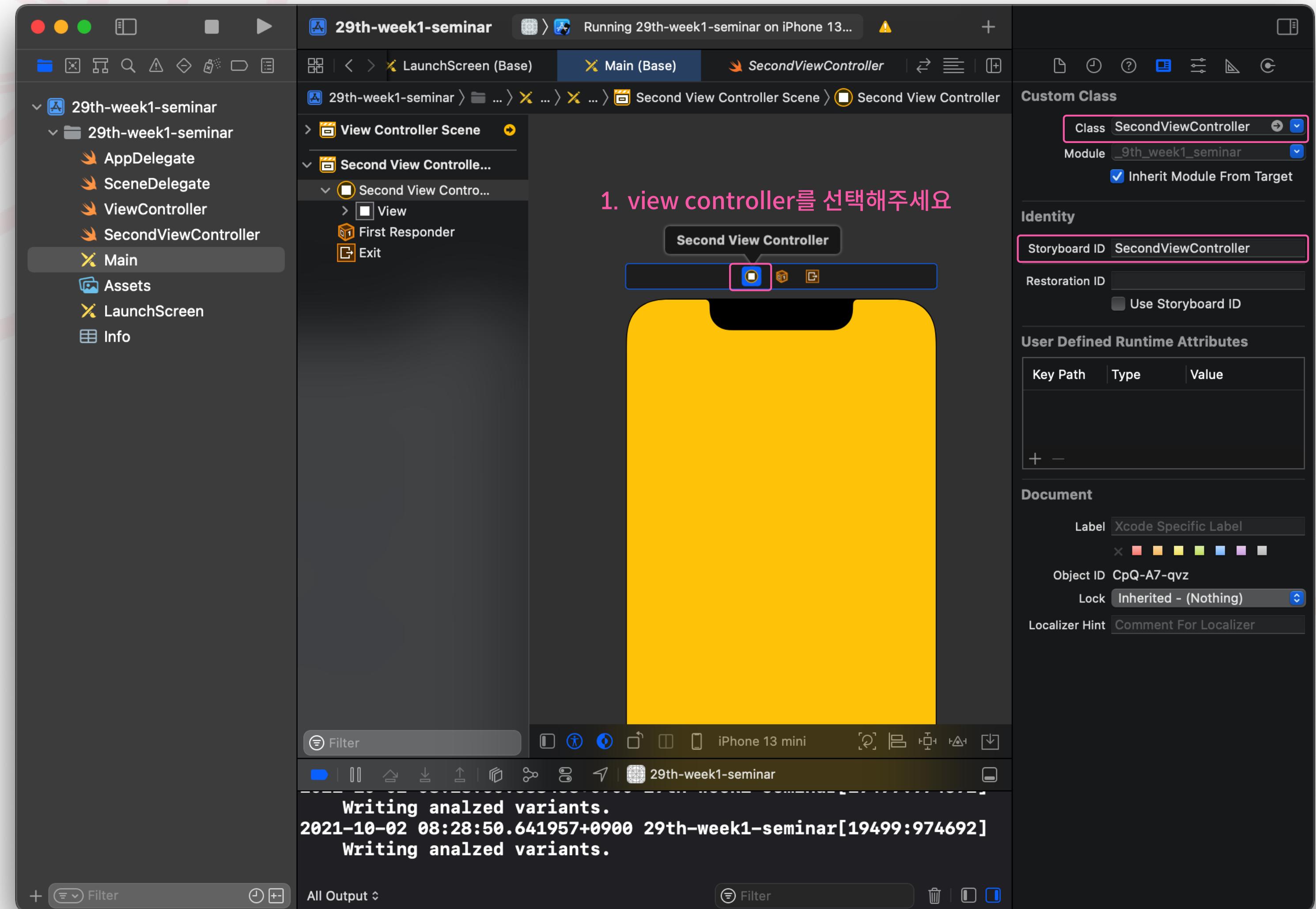
○ 화면 전환 | 모달(Modal)



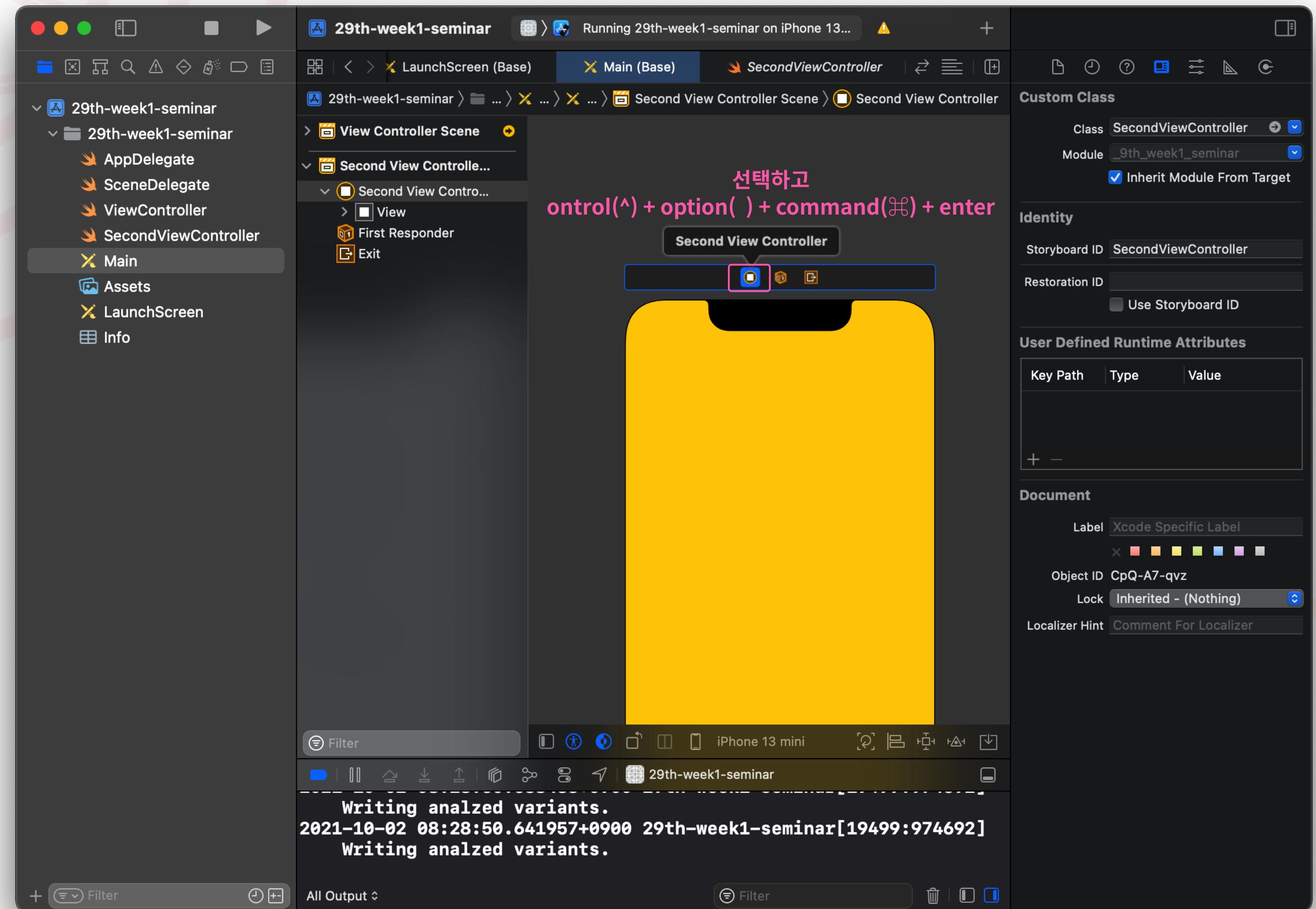
○ 화면 전환 | 모달(Modal)



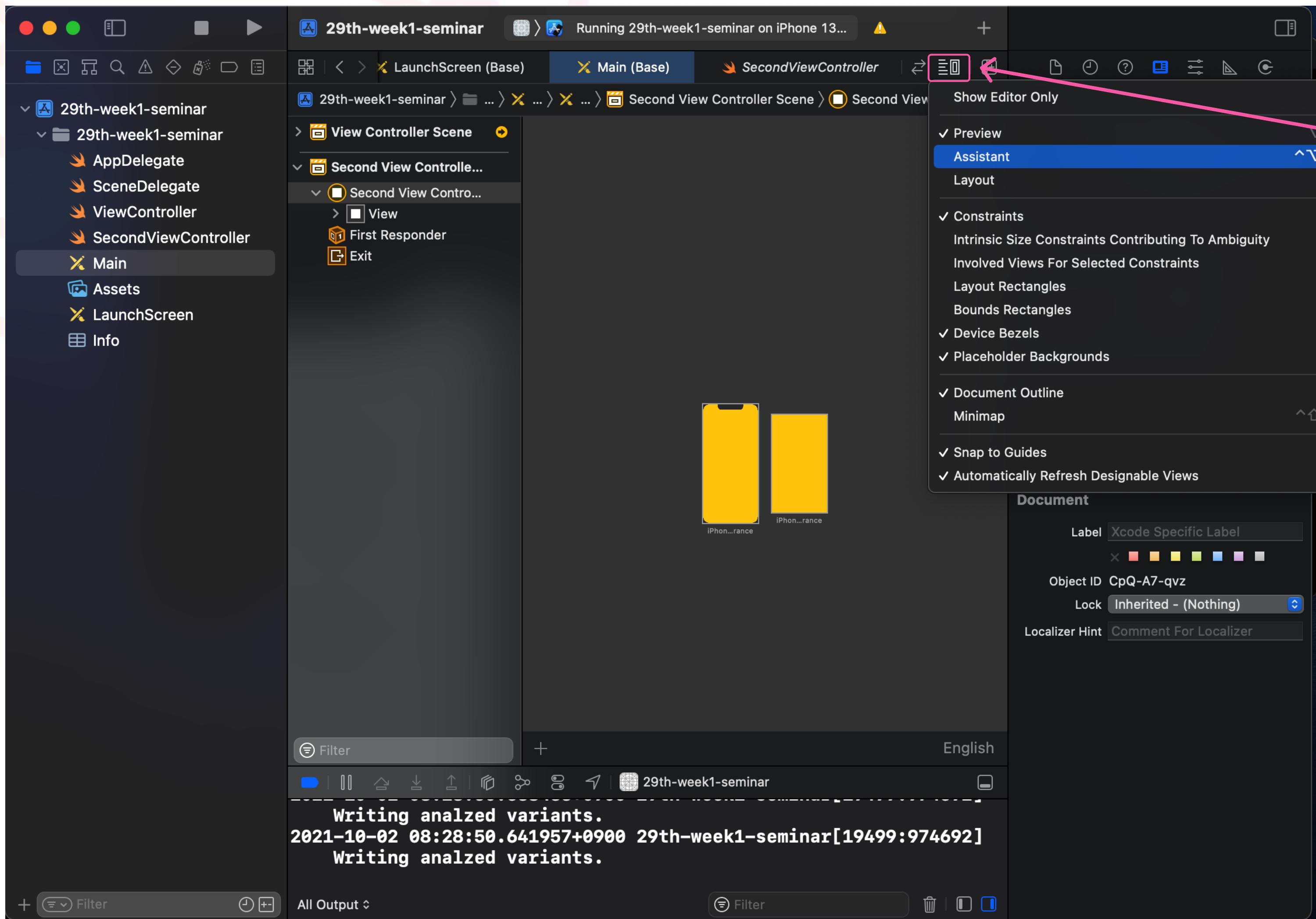
○ 화면 전환 | 모달(Modal)



○ 화면 전환 | 모달(Modal)



○ 화면 전환 | 모달(Modal)



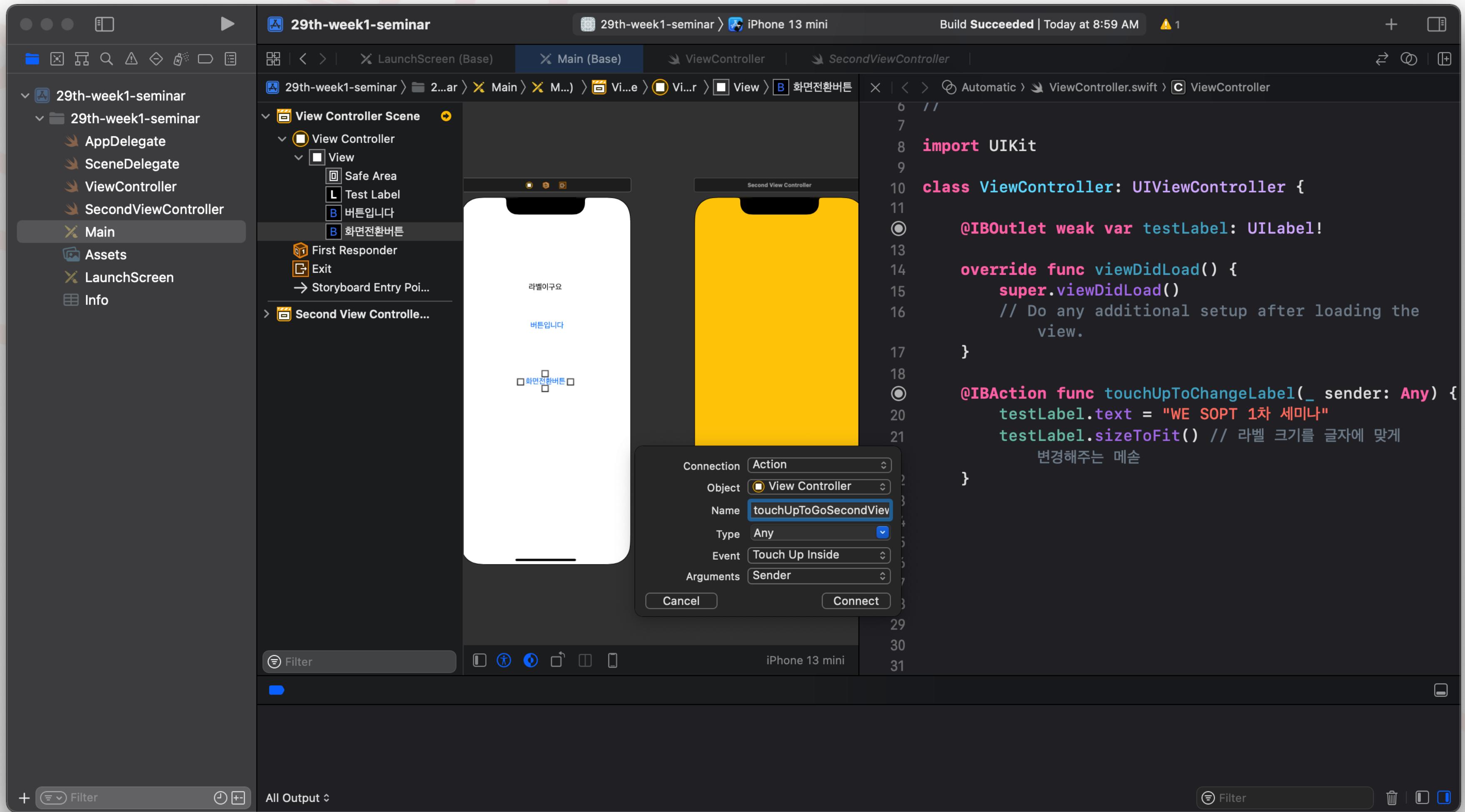
바로 어시스턴스 에디터가 열리지 않을 때!

어시스턴스 데이터가 아닌 다른 화면 나온다면
눌러서 assistant를 선택합니다

파일 연결이 잘 되었는지 확인한다.
SecondViewController.swift에 연결되었는지 확인합니다.

그래도 잘 안된다면?
한 번 꺼다가 켜주세요... 😢

○ 화면 전환 | 모달(Modal)



화면 전환버튼을 끌어다
Action을 만들어주세요!

○ 화면 전환 | 모달(Modal)

```
@IBAction func touchUpToGoSecondView(_ sender: Any) {  
    guard let nextVC = self.storyboard?.instantiateViewController(withIdentifier: "SecondViewController")  
    else {return}  
  
    self.present(nextVC, animated: true, completion: nil)  
}
```



guard (조건) else { 조건이 false 일 때 실행되는 부분}
// else 안에는 break나 return과 같이 종료 구문이 있어야 함

optional인 storyboard에서
SecondViewController라는 identifier를 가진
View controller instance를 생성해서
nextVC 상수에 담아줘

이를 실행하다가 무엇하나라도 잘못되면 else문에 걸린다면 return 되어
해당 touchUpToSecondView함수가 종료됩니다.

guard 구문을 빠져나온 nextVC 는 확실한 instance

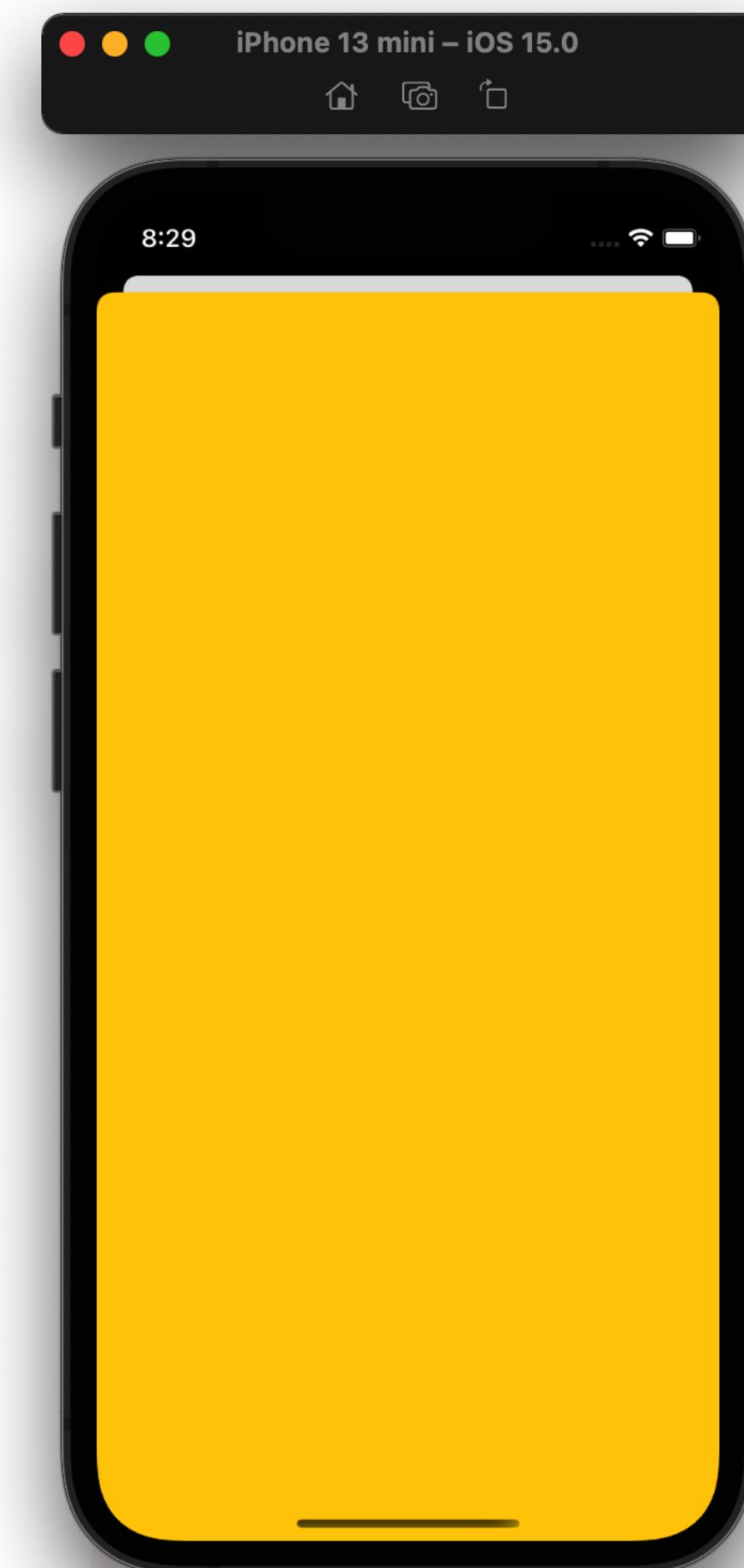
○ 화면 전환 | 모달(Modal)

```
@IBAction func touchUpToGoSecondView(_ sender: Any) {  
    guard let nextVC = self.storyboard?.instantiateViewController(withIdentifier: "SecondViewController")  
    else {return}  
  
    self.present(nextVC, animated: true, completion: nil)  
}
```

모달을 띄우는 present 메서드를 통해 nextVC를 띄웁니다.

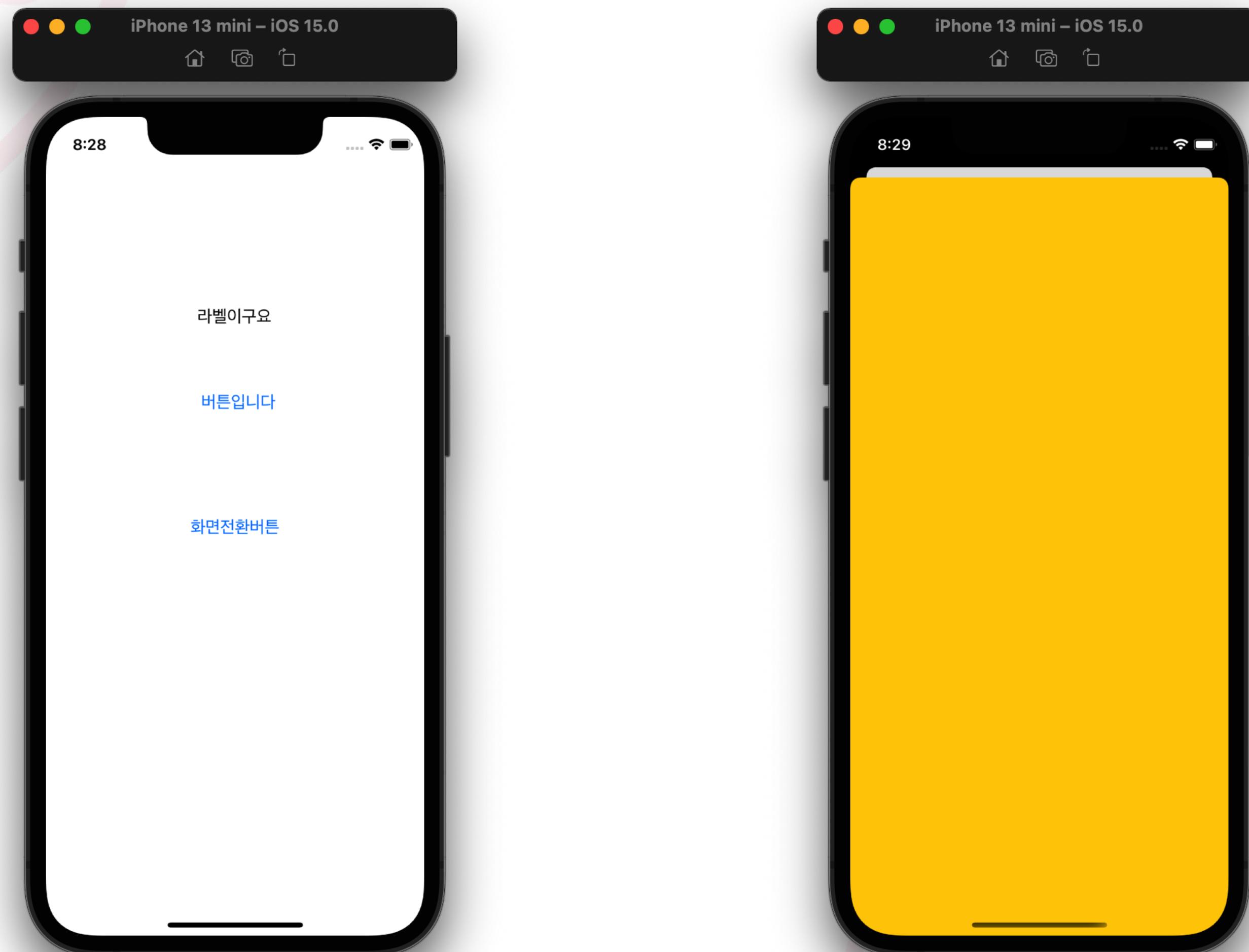
○ 화면 전환 | 모달(Modal)

빌드해서 확인해보세요! (command + R)



○ 화면 전환 | 모달(Modal)

빌드해서 확인해보세요! (command + R)



○ 화면 전환 | 모달(Modal)

```
@IBAction func touchUpToGoSecondView(_ sender: Any) {  
    guard let nextVC = self.storyboard?.instantiateViewController(withIdentifier: "SecondViewController")  
    else {return}  
  
    nextVC.modalPresentationStyle = .pageSheet  
    nextVC.modalTransitionStyle = .crossDissolve  
    self.present(nextVC, animated: true, completion: nil)  
}
```

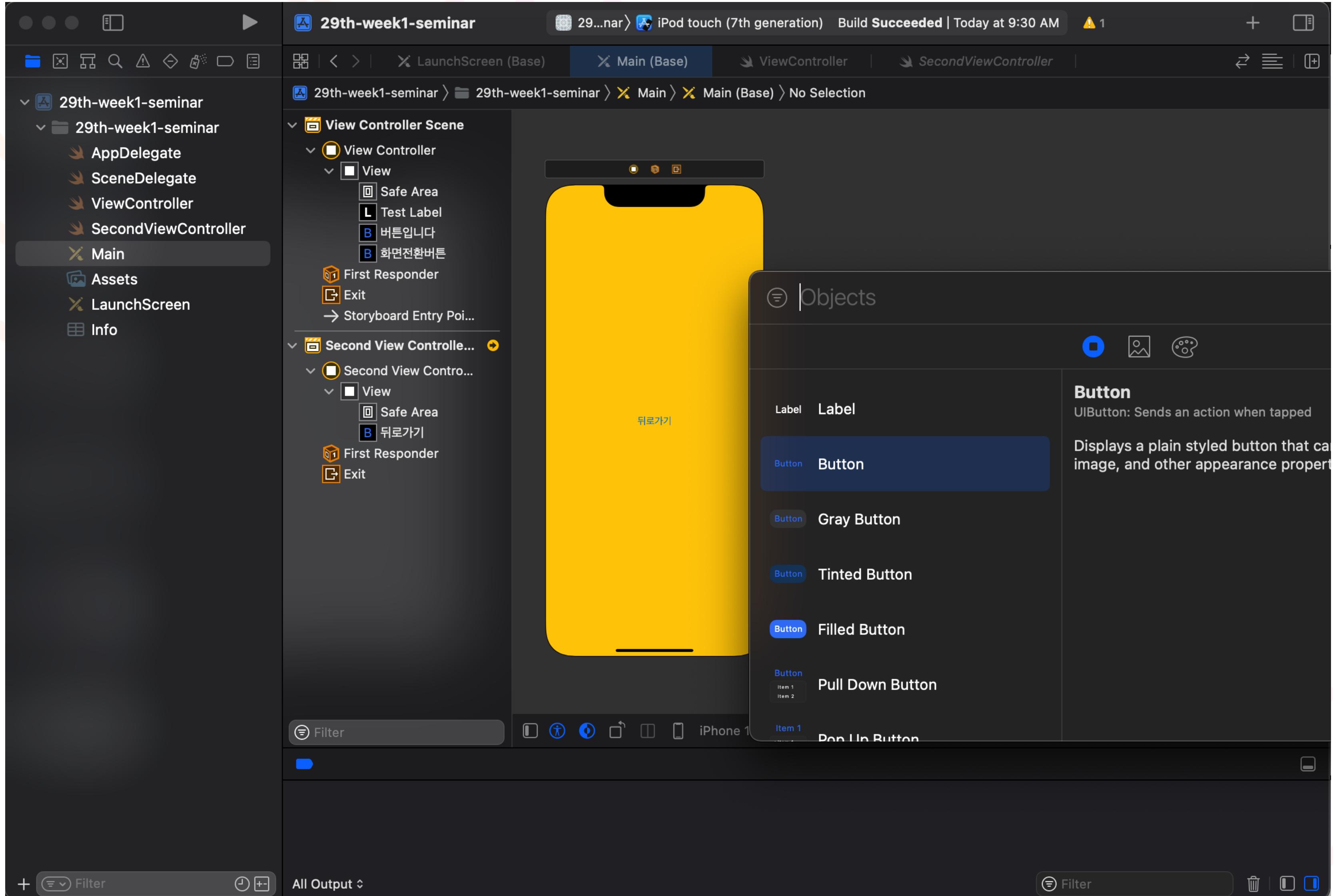
modalPresentationStyle
새로운 모달창을 어떻게 보여줄지

- automatic : 시스템에서 정한 기본 옵션 (현재 하는 방식)
- fullScreen : 위를 남기지 않고 새로운 모달로 창을 덮습니다.
- overCurrentContext : 새로운 모달창이 투명한 경우,
아래에 깔려있는 이전 뷰도 함께 보임

modalTransitionStyle
새로운 모달창을 어떻게 전환할지

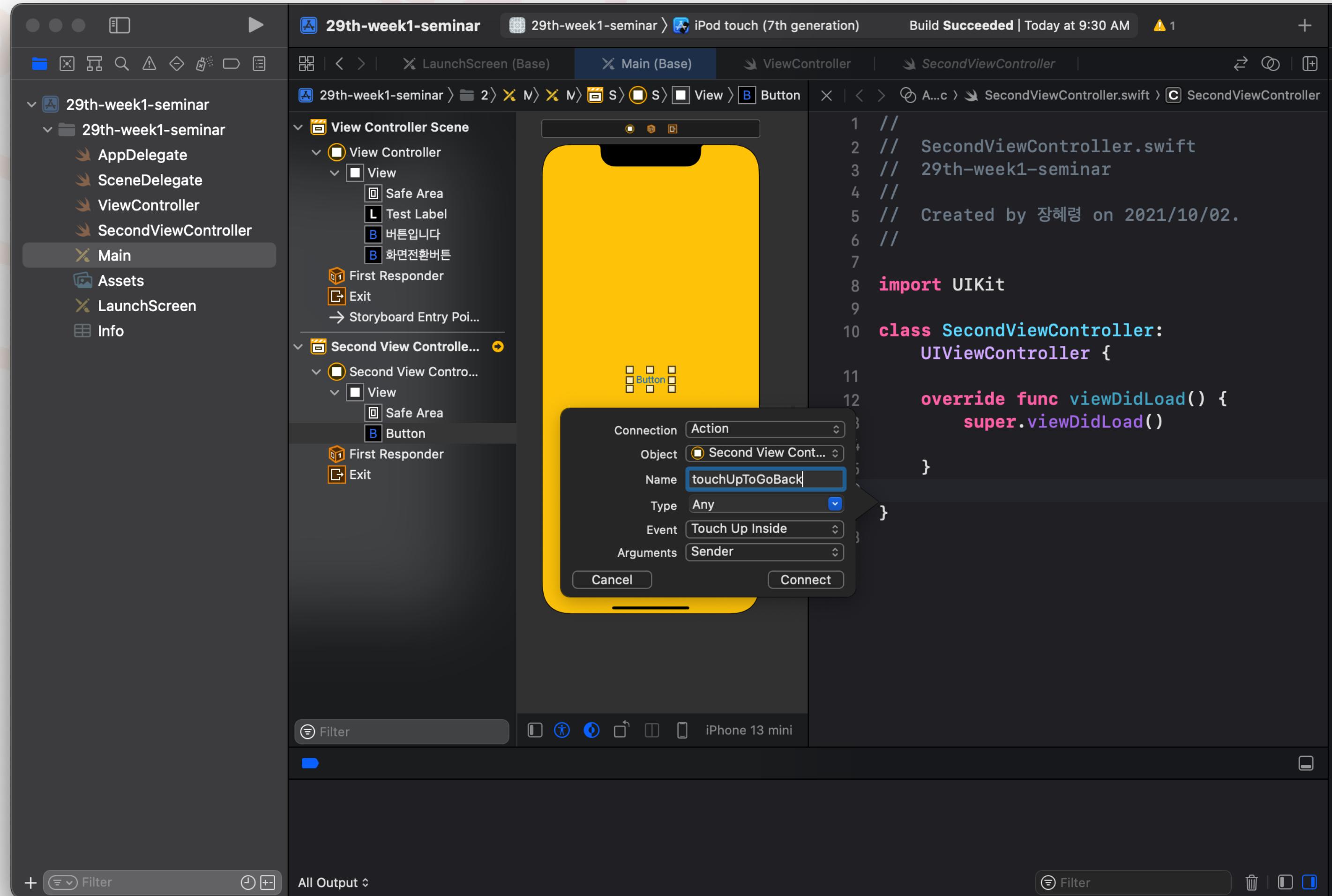
- overVertical : 우리가 처음 본 것처럼,
아래에서 나와 뷰를 올리는 형식
- crossDissolve : 뷰가 교차되면서 전환됨.

○ 화면 전환 | 모달(Modal)



**SecondViewController에
뒤로가기 버튼을 추가해주세요**

○ 화면 전환 | 모달(Modal)



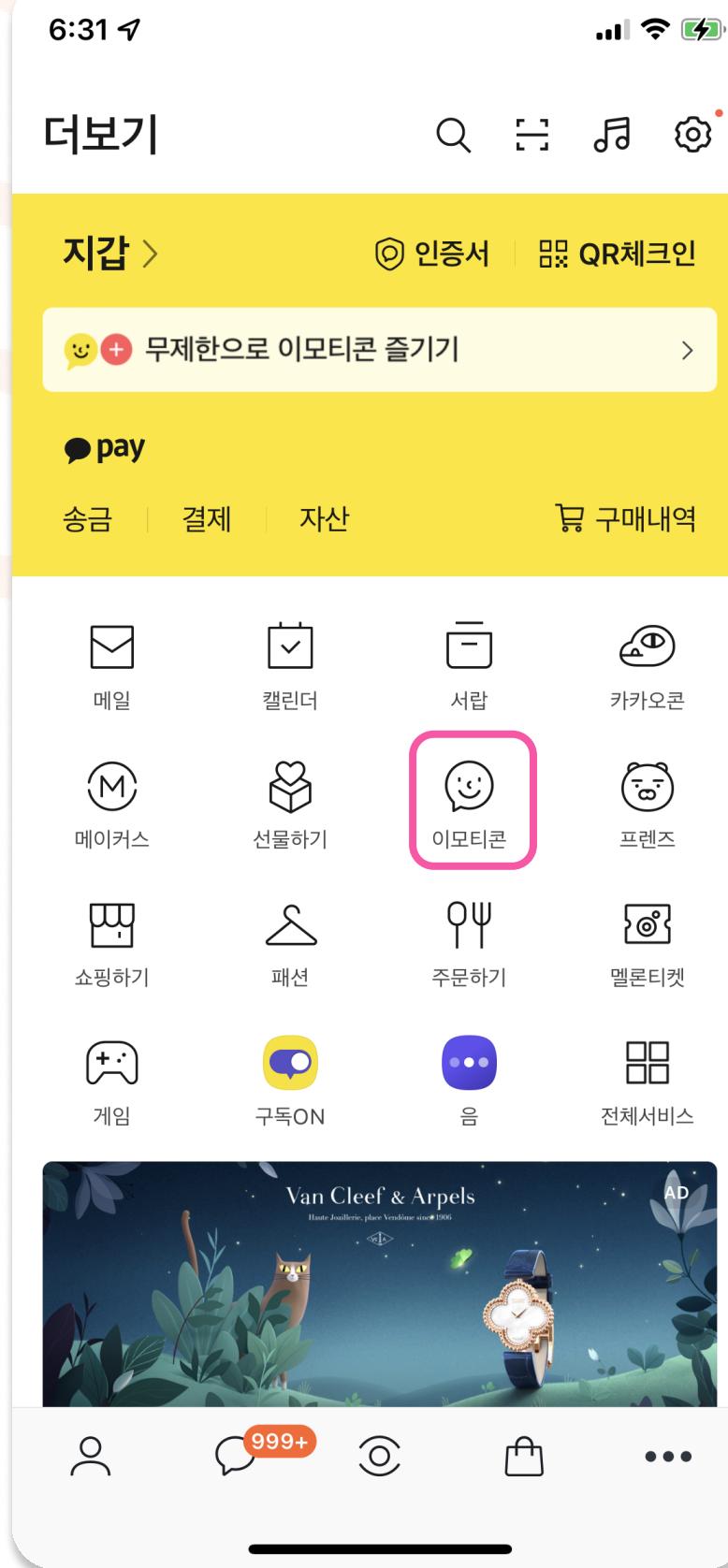
추가한 버튼에
IBAction도 추가해주세요:)

○ 화면 전환 | 모달(Modal)

```
8 import UIKit
9
10 class SecondViewController: UIViewController {
11
12     override func viewDidLoad() {
13         super.viewDidLoad()
14
15     }
16
17     @IBAction func touchUpToGoBack(_ sender: Any) {
18         self.dismiss(animated: true, completion: nil)
19     }
20 }
21
22 |
```

dismiss 메서드는
현재 present로 띄워진 뷰를
사라지게 하는 메서드

수많은 APP들을 사용하면서 다양한 화면을 보게되는데
현재 보던 화면에서 다른 화면으로 이동하기 위해서는 **화면 전환**이 필요합니다!



모달(Modal)

위로 속 올라오면서 하단에 있던
친구, 채팅 아이콘들이 사라지고
새로운 화면이 나왔습니다



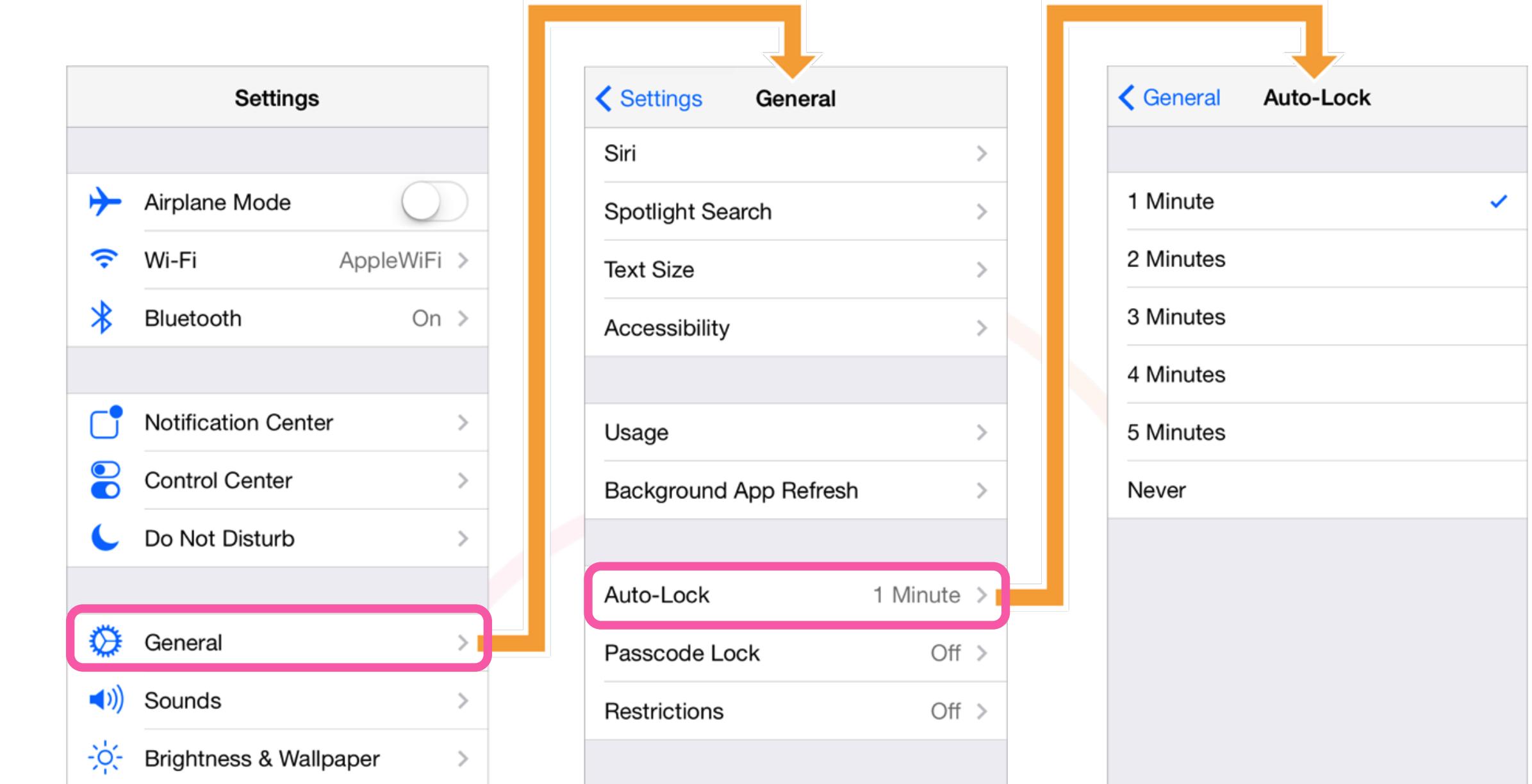
네비게이션(Navigation)

이번에는 오른쪽으로 속 화면이 생기고
또 다른 화면이 나왔습니다



📌 네비게이션 인터페이스

- iOS 화면 전환을 위해 사용 되는 기법 중 가장 대표적인 방법
- 위 이미지와 같이 계층적 구조의 뷰 워크플로우를 구현하기 위해 사용
- UINavigationController에 의해 관리



📌 Navigation Controller 란?

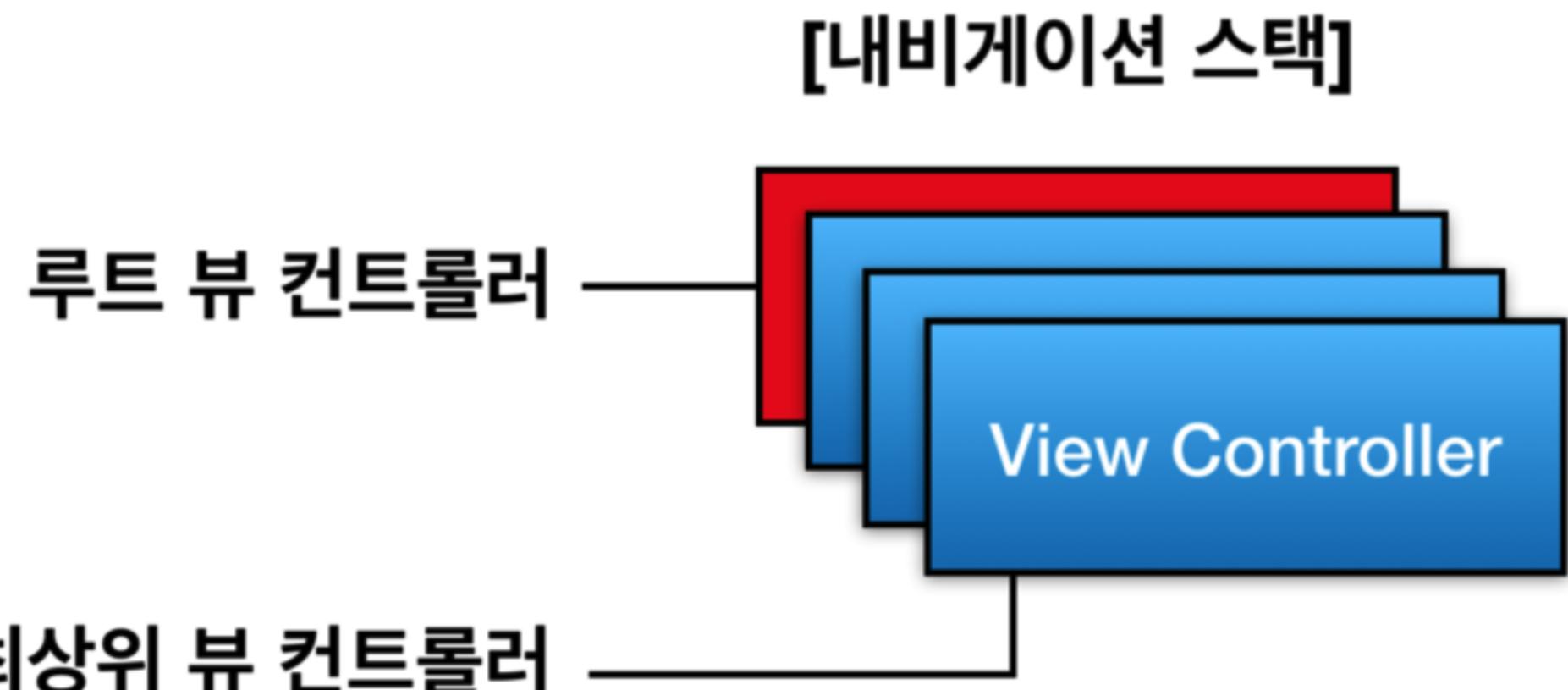
- ViewController 사이 계층 구조를 탐색할 수 있게 해주는 객체
- Present 방식과는 다르게 자식 뷰는 Stack으로 관리
- UIViewController를 담을 수 있는 Container ViewController
- 두 개의 뷰를 화면에 표시
 1. Navigation 스택에 포함된 최상위 ViewController
 2. 직접 관리하는 Navigation bar, Tool bar



📌 Navigation Controller 란?

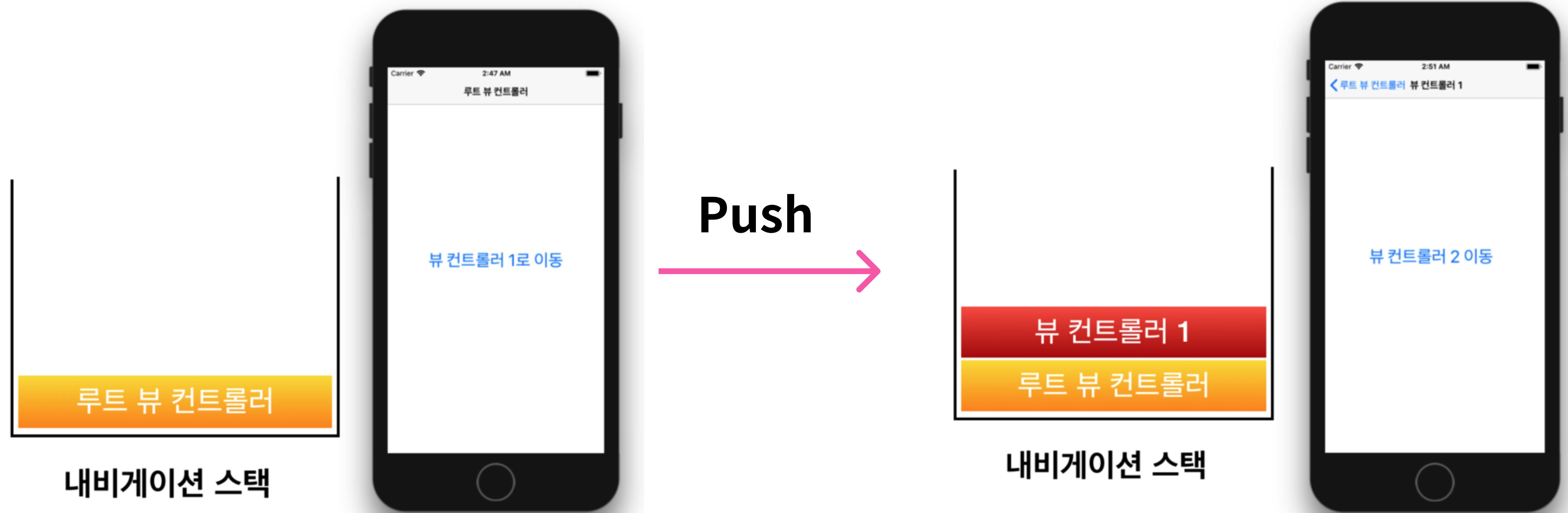
📌 Navigation Stack

- ViewController를 담을 수 있는 배열
- 가장 먼저 스택에 추가 된 ViewController가 ‘루트’ ViewController
- 가장 늦게 들어간 ViewController 가 ‘최상위’ ViewController가 된다.



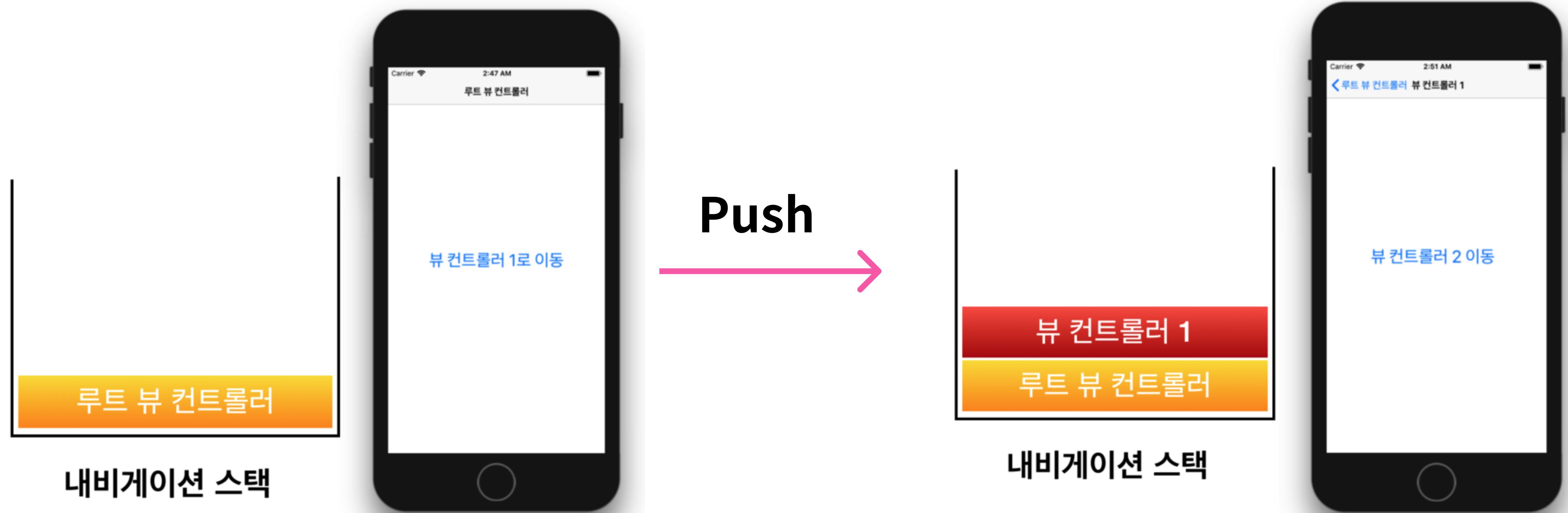
📌 **Navigation Controller 란?**

📌 **Navigation Stack**



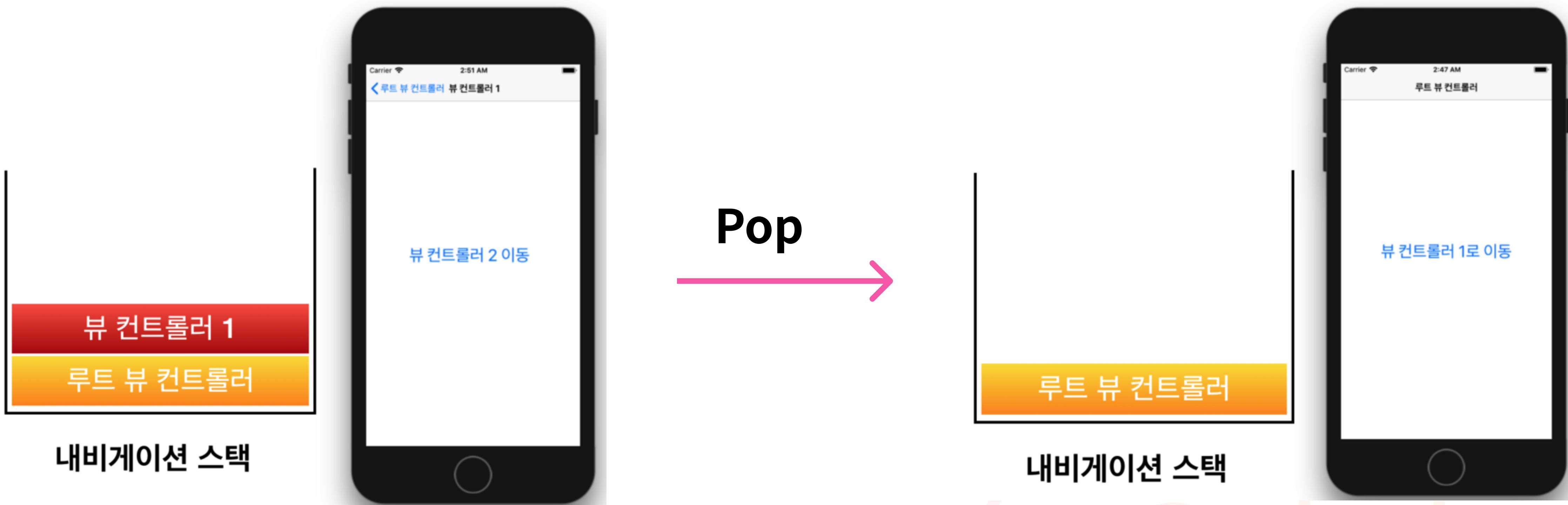
📌 **Navigation Controller 란?**

📌 **Navigation Stack**



📌 **Navigation Controller 란?**

📌 **Navigation Stack**



📌 Navigation Controller 란?

📌 Navigation Stack



○ 화면 전환 | Navigation

```
class ViewController: UIViewController {

    @IBOutlet weak var testLabel: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }

    @IBAction func touchUpToChangeLabel(_ sender: Any) {
        testLabel.text = "WE SOPT 1차 세미나"
        testLabel.sizeToFit() // 라벨 크기를 글자에 맞게 변경해주는 메소드
    }

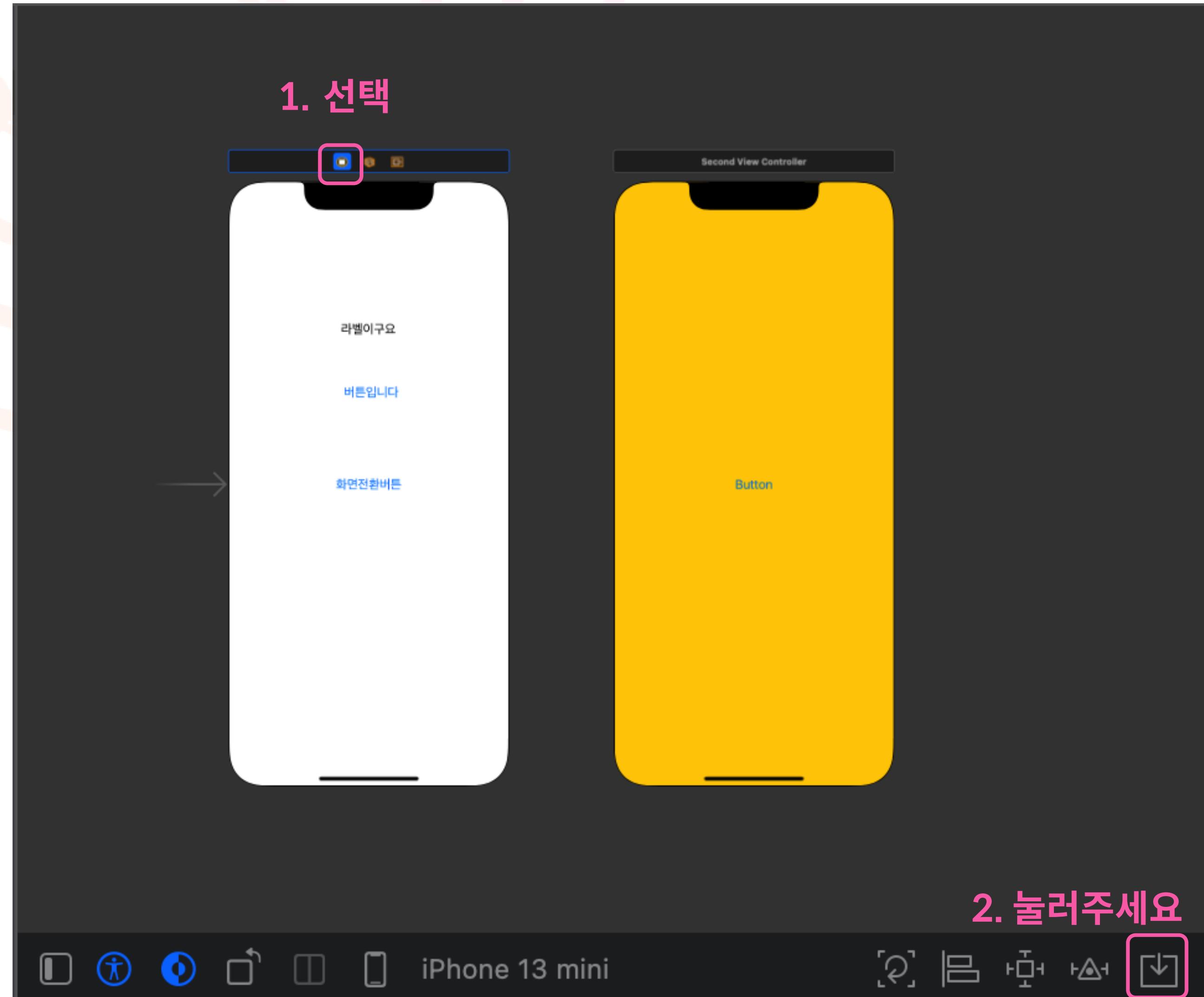
    @IBAction func touchUpToGoSecondView(_ sender: Any) {
        guard let nextVC =
            self.storyboard?.instantiateViewController(withIdentifier:
                "SecondViewController")
        else {return}

        // nextVC.modalPresentationStyle = .fullScreen
        // nextVC.modalTransitionStyle = .crossDissolve
        // self.present(nextVC, animated: true, completion: nil)
    }
}
```

모달로 띄웠던 코드를
먼저 주석처리 해주세요

드래그 한 후 한 번에
command + / 하면
주석처리 됩니다:)

○ 화면 전환 | Navigation



3. 네이게이션 컨트롤러를 선택해주세요

Embed In View

View

View Without Inset

ScrollView

Stack View

Embed In View Controller

Navigation Controller

Tab Bar Controller

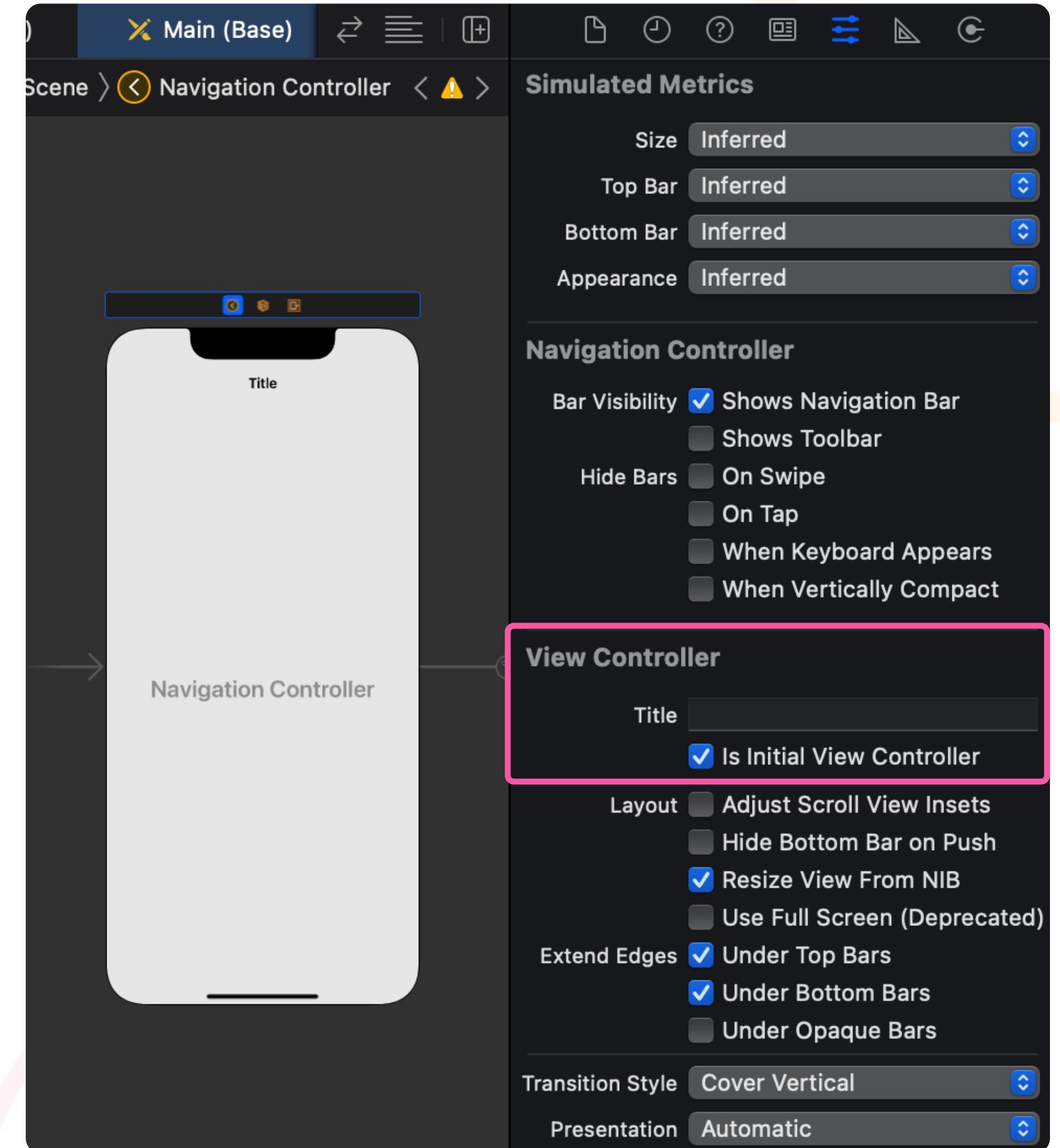
Unembed

○ 화면 전환 | Navigation



Is initial View Controller

: 스토리 보드의 진입 지점이 되는 뷰 컨트롤러



ViewController.swift

```
@IBAction func touchUpToGoSecondView(_ sender: Any) {  
    guard let nextVC =  
        self.storyboard?.instantiateViewController(withIdentifier:  
        "SecondViewController")  
    else {return}  
  
    self.navigationController?.pushViewController(nextVC, animated: true)  
}
```

SecondViewController.swift

```
@IBAction func touchUpToGoBack(_ sender: Any) {  
    //self.dismiss(animated: true, completion: nil)  
    self.navigationController?.popViewController(animated: true)  
}
```

05

데이터 전달 기초

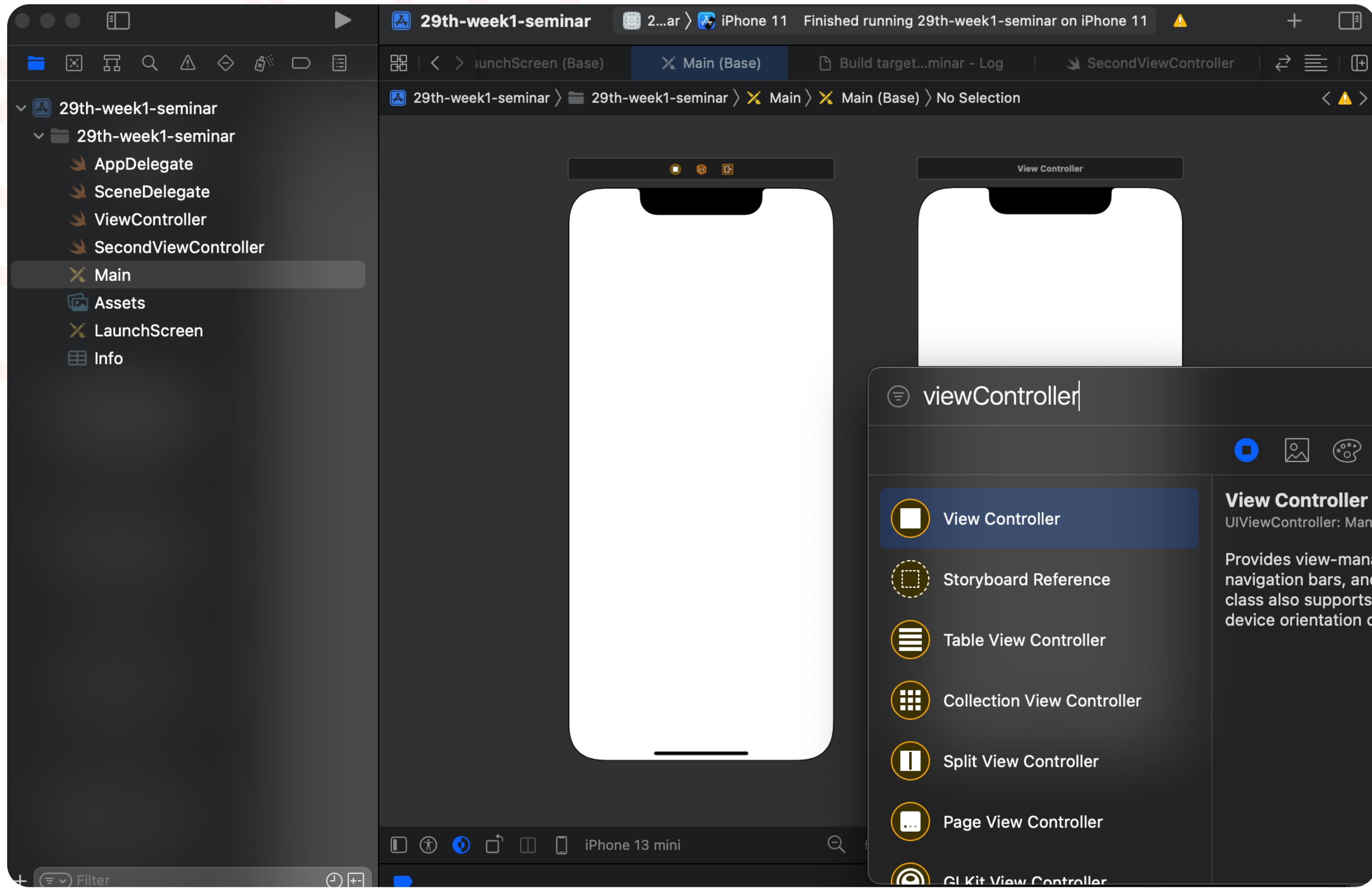
○ 데이터 전달 기초



이번에는 프로퍼티를 이용한
기초적인 뷰간 데이터 전달을 알아보려 합니다!

- ✓ 3주차 다양한 데이터 전달
- ✓ 7주차 데이터 전달 심화

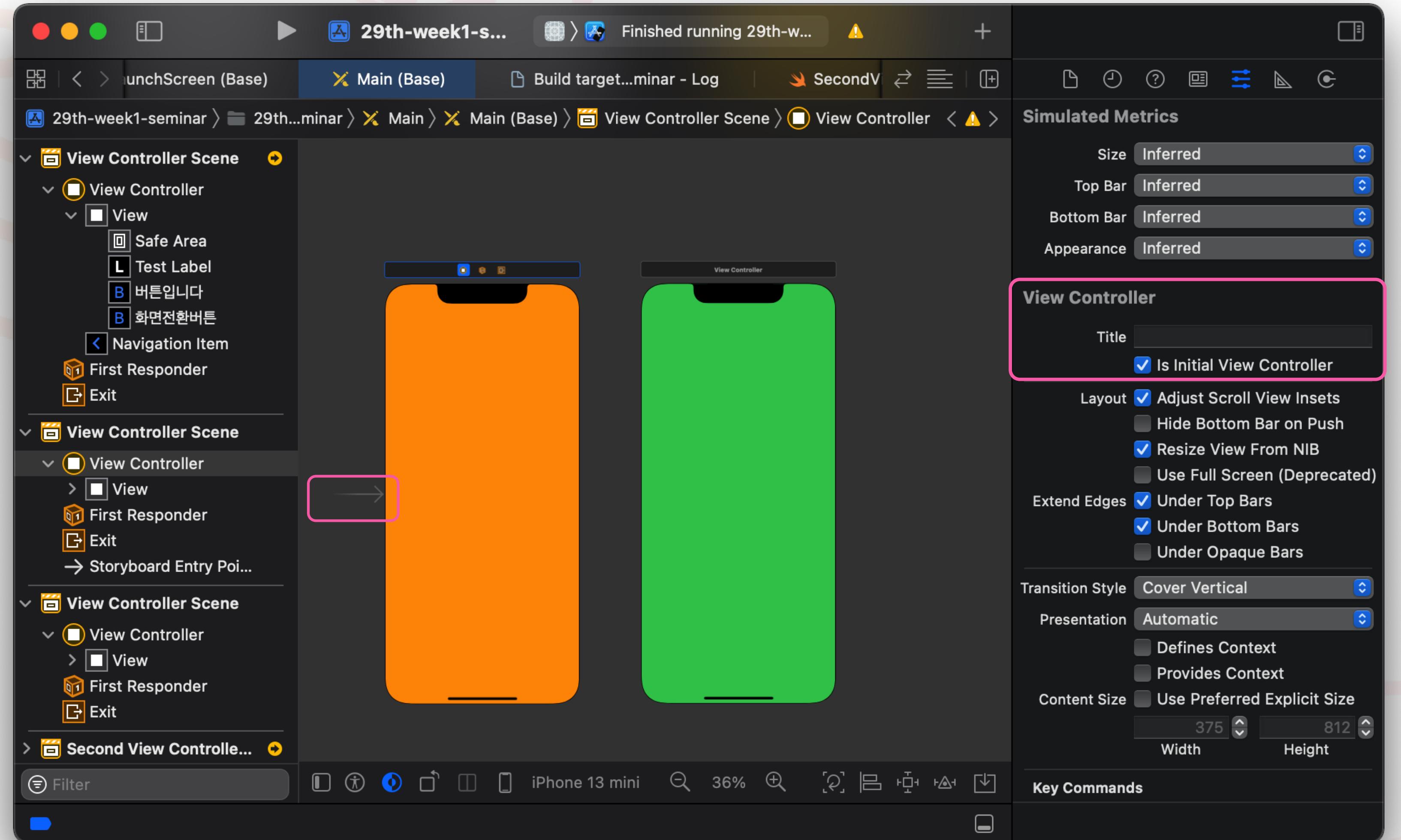
데이터 전달 기초



이제 조금씩 익숙해 지시고 계신가요??

cmd + shift + L을 눌러
View Controller 2개를
스토리보드에 놓아주세요!

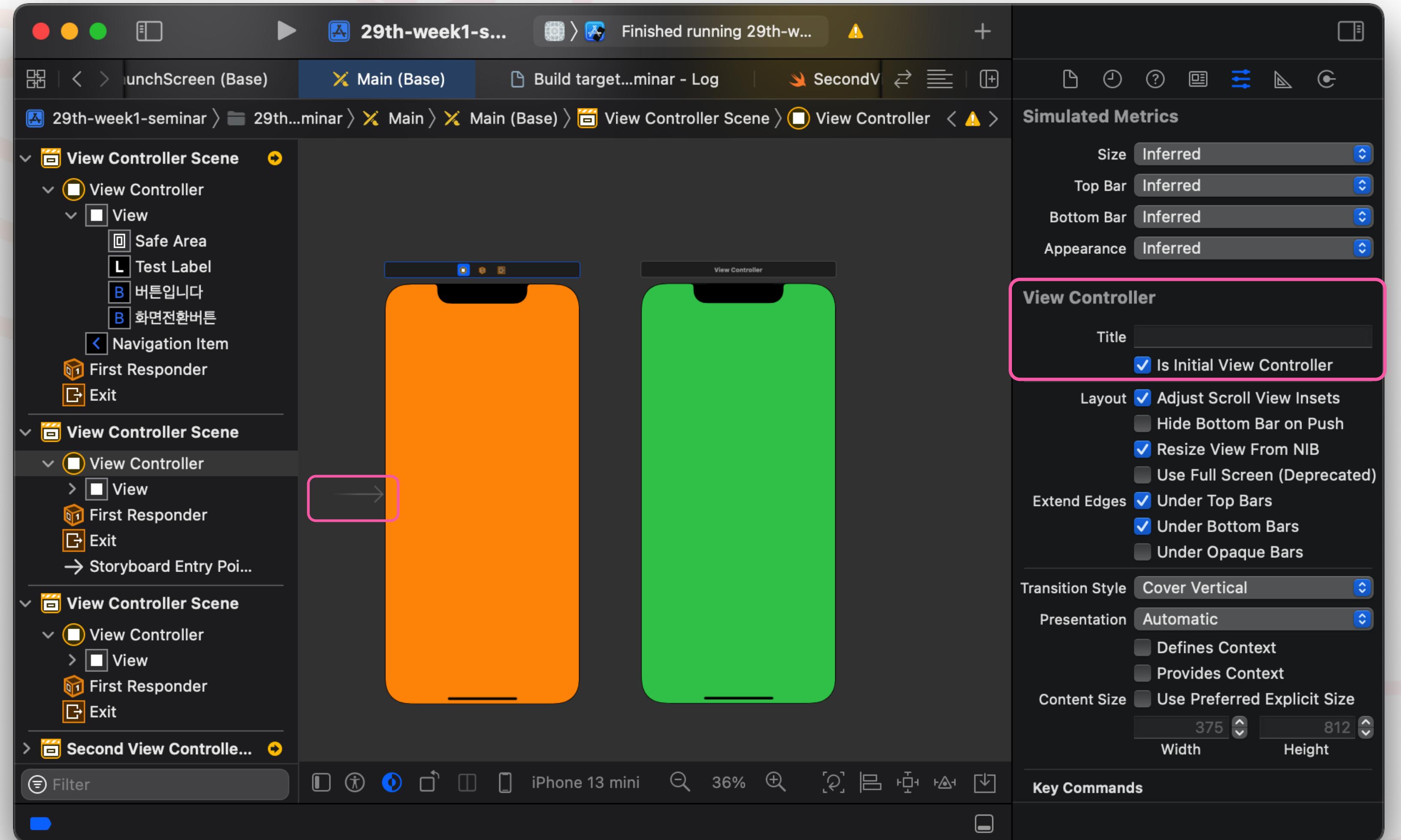
데이터 전달 기초



구분하기 쉽도록
각 view controller마다
view를 눌러 background를 변경해주세요!

그 후 스토리보드상에
왼쪽에 있는 View Controller를
Initial View Controller로 지정해주세요:)

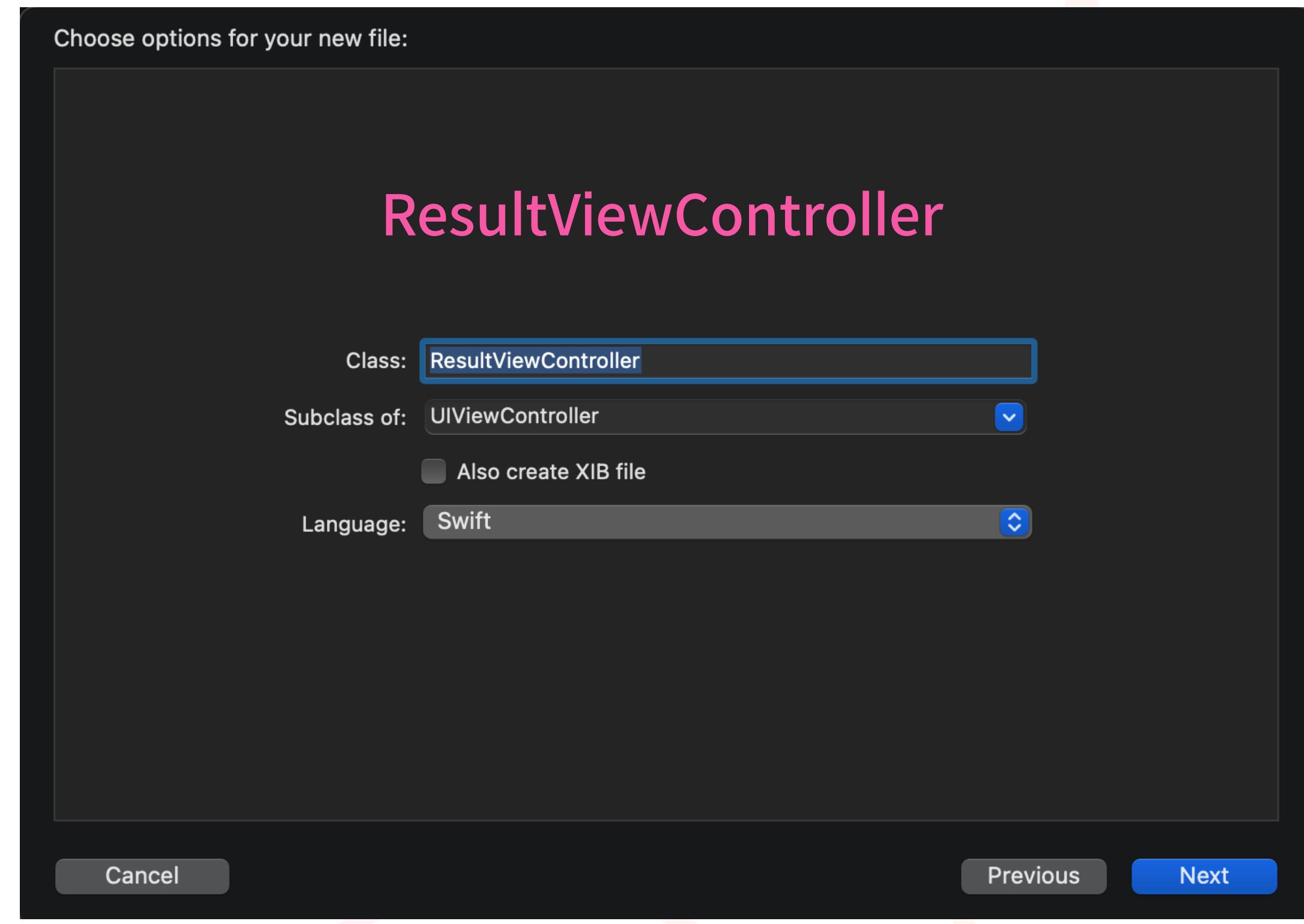
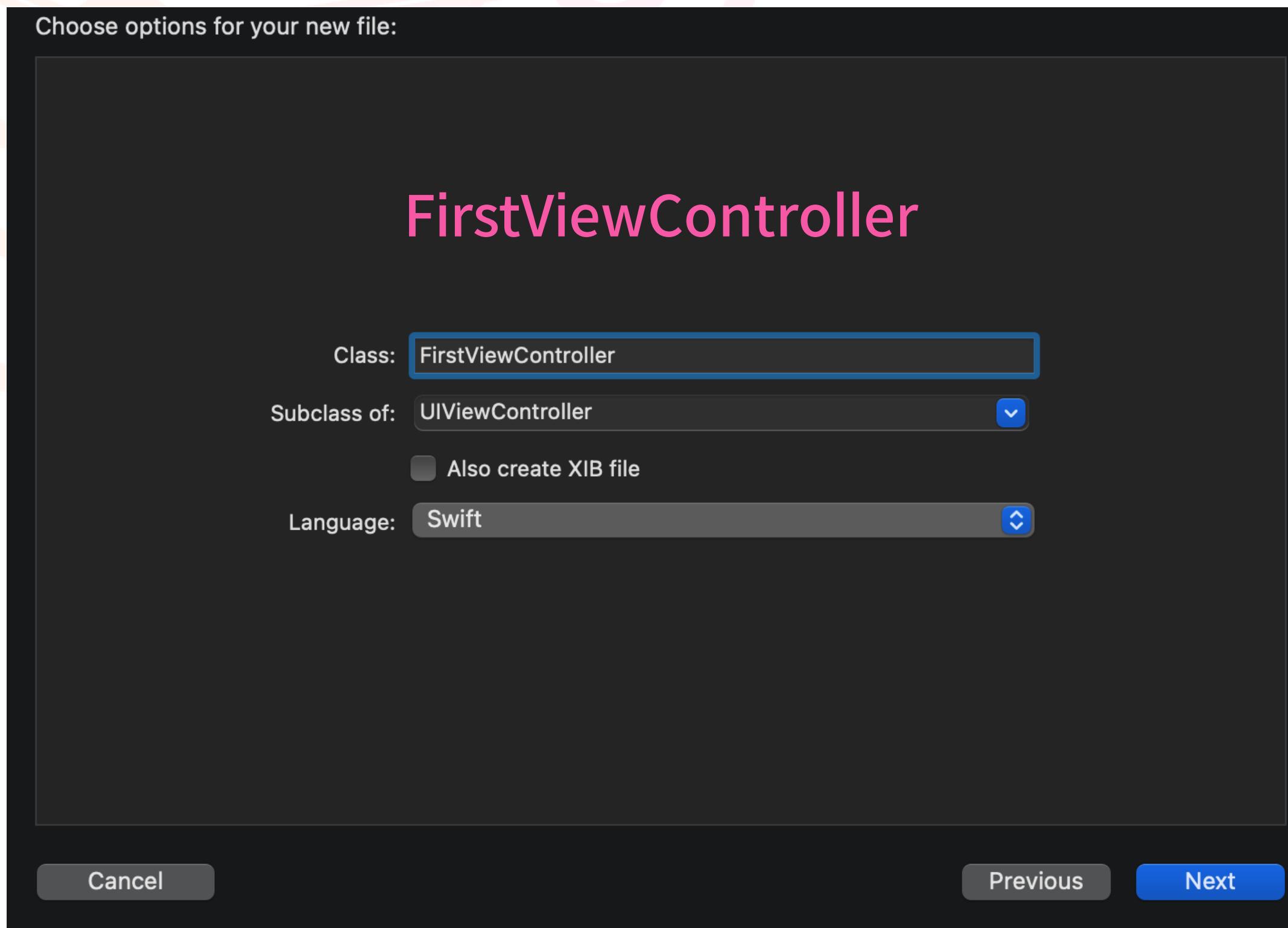
데이터 전달 기초



구분하기 쉽도록
각 view controller마다
view를 눌러 background를 변경해주세요!

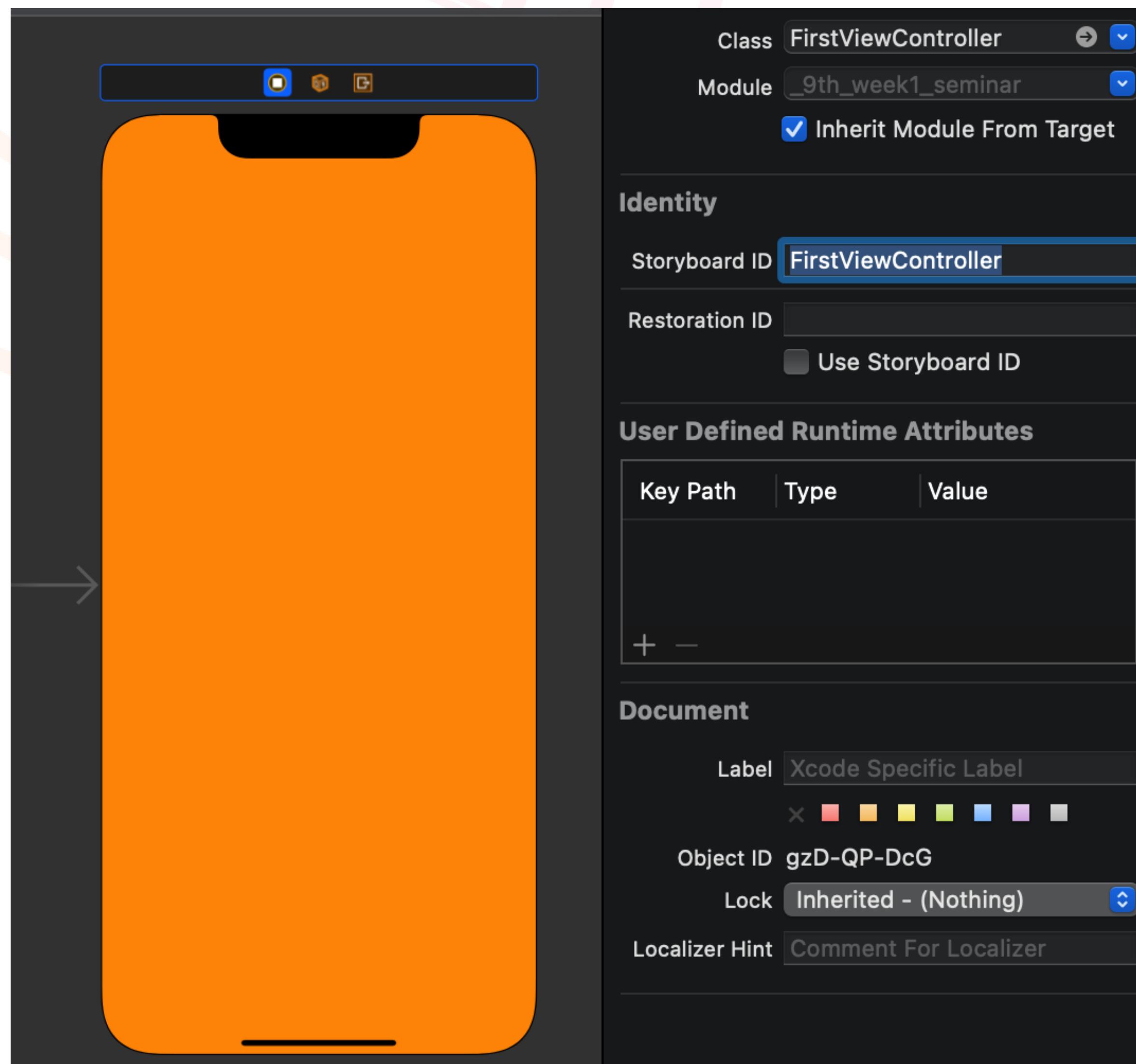
그 후 스토리보드상에
왼쪽에 있는 View Controller를
Initial View Controller로 지정해주세요:)

데이터 전달 기초

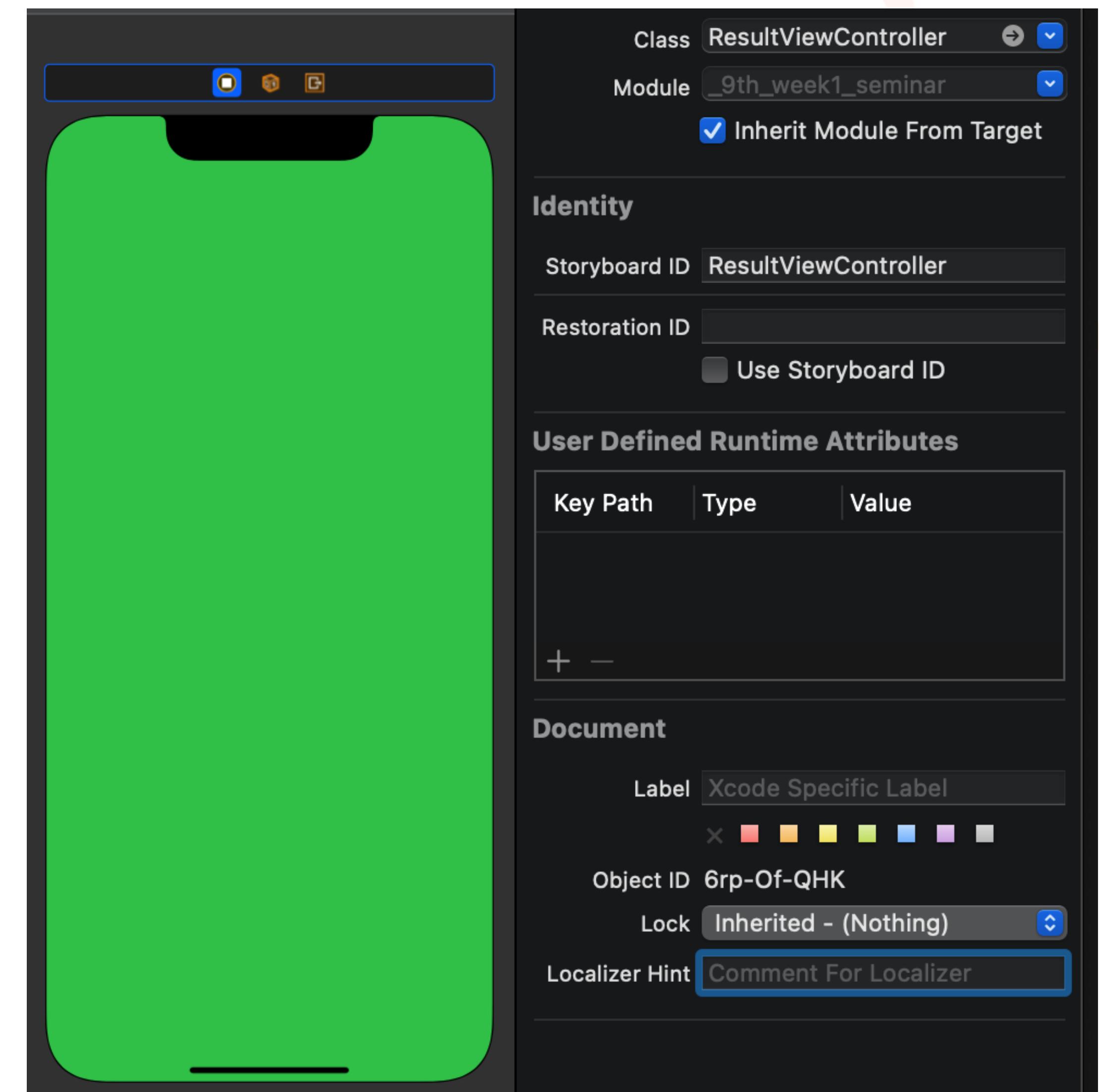


데이터 전달 기초

FirstViewController

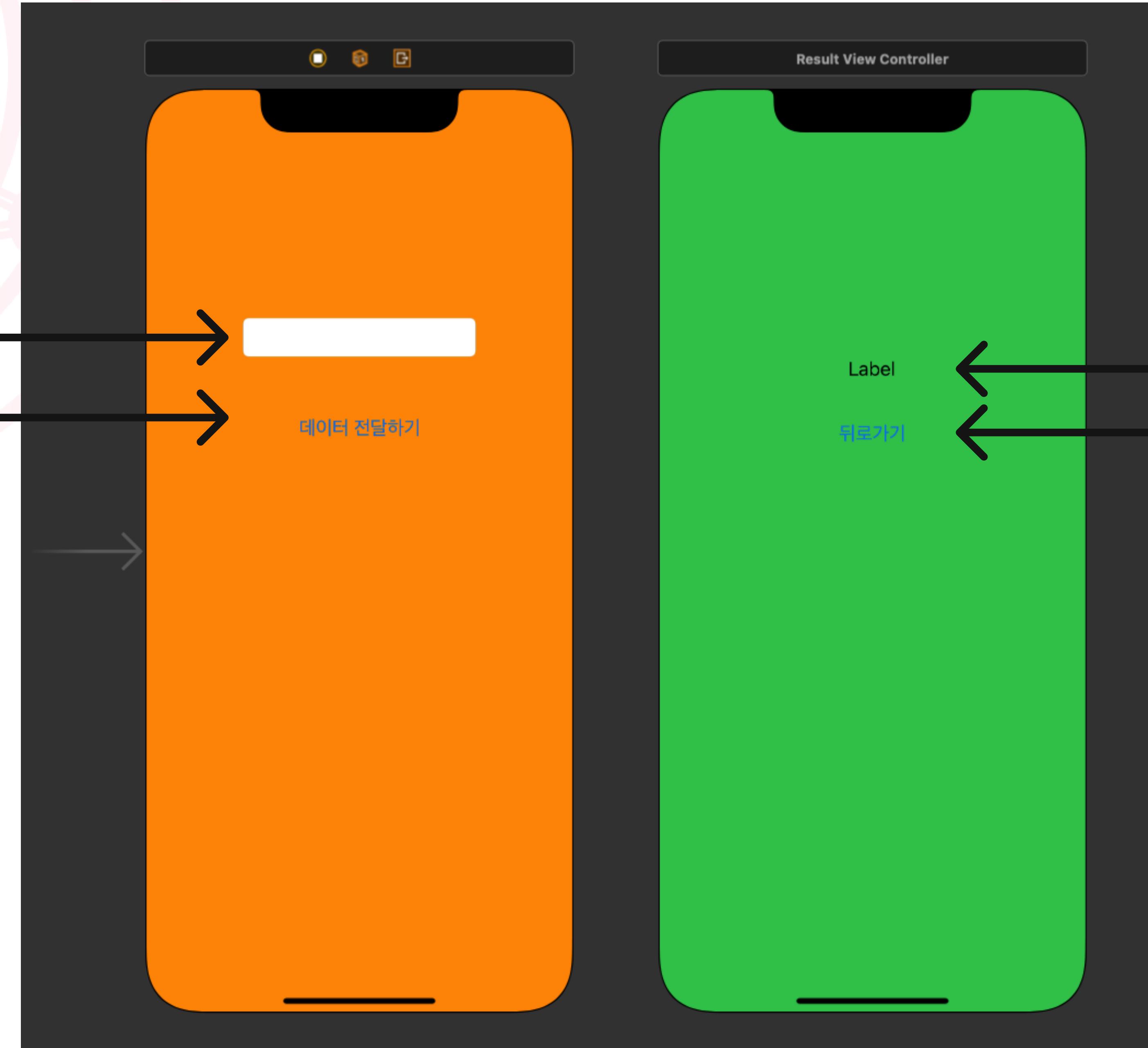


ResultViewController



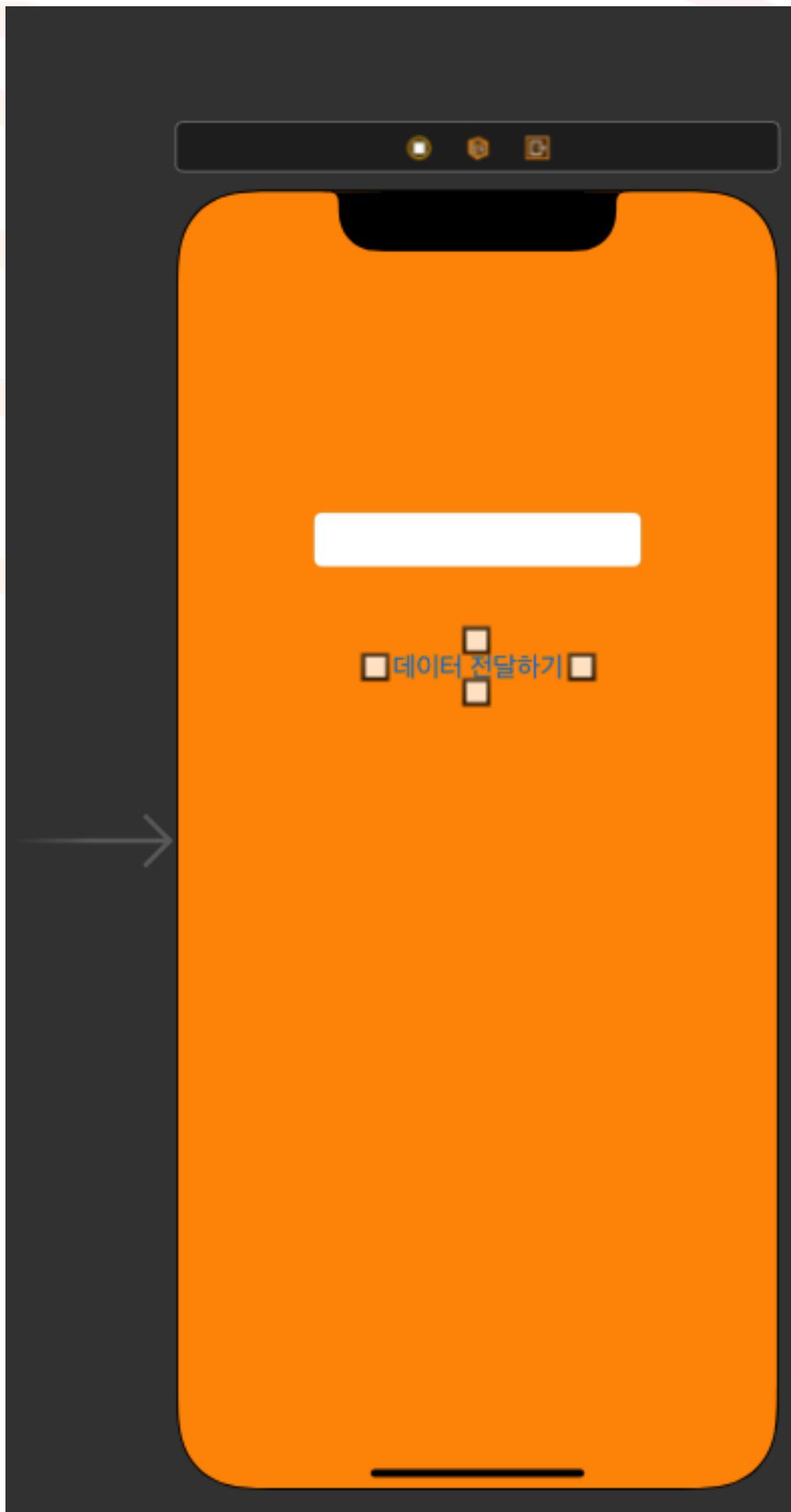
데이터 전달 기초

전달할 Text를 넣을
TextField
화면전환 버튼



전달받은 Text
나타낼 Label
뒤로가기
화면전환 버튼

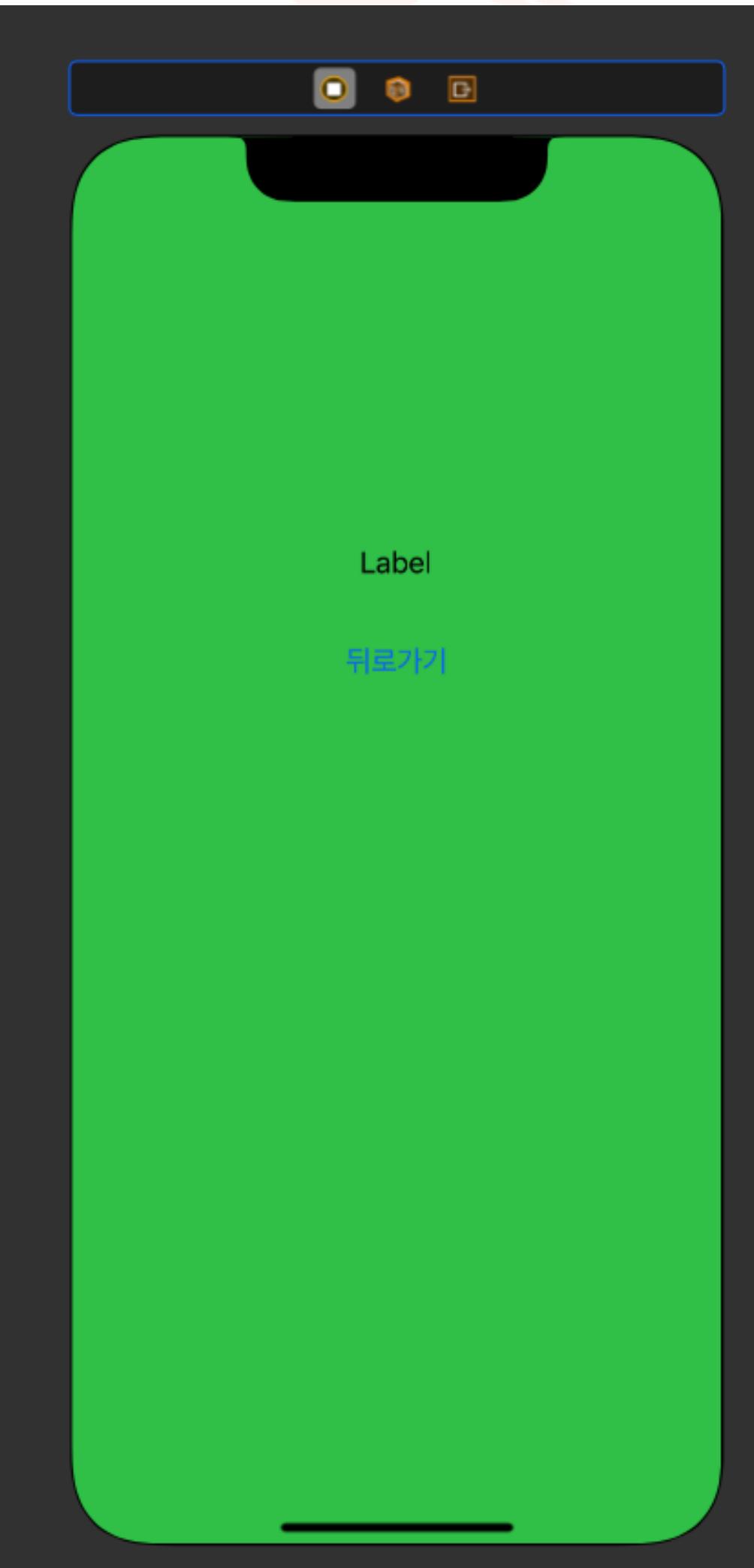
데이터 전달 기초



```
7
8 import UIKit
9
10 class FirstViewController: UIViewController {
11
12     @IBOutlet weak var dataTextField:
13         UITextField!
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17     }
18
19     @IBAction func touchUpToSendData(_
20         sender: Any) {
21
22     }
23 }
```

textField는 IBOutlet으로
button은 IBAction으로
연결해주세요:)

○ 데이터 전달 기초



```
5 // Created by 장혜령 on 2021/10/02.  
6 //  
7  
8 import UIKit  
9  
10 class ResultViewController: UIViewController {  
11  
12     @IBOutlet weak var dataLabel: UILabel!  
13  
14     override func viewDidLoad() {  
15         super.viewDidLoad()  
16     }  
17  
18     @IBAction func touchUpToGoBack(_ sender: Any) {  
19     }  
20  
21 }  
22
```

label은 IBOutlet으로
button은 IBAction으로
연결해주세요:)

데이터 전달 기초

```
6 //  
7  
8 import UIKit  
9  
10 class ResultViewController: UIViewController {  
11     @IBOutlet weak var dataLabel: UILabel!  
12     var message: String?  
13  
14     override func viewDidLoad() {  
15         super.viewDidLoad()  
16     }  
17  
18     @IBAction func touchUpToGoBack(_ sender: Any) {  
19     }  
20  
21 }  
22  
23
```

- 이전 화면에서 전달되는 값을 받기 위해 **프로퍼티를 추가하는 작업**
- IBOutlet 변수는 외부에서 값을 직접 대입할 수 없는 변수
- IBOutlet 변수는 외부에서 직접 참조할 수 없음.
- 그렇기 때문에, 두번째 화면으로 값을 대입하기 위해
따로 값을 지정해 주어야 함.

○ 데이터 전달 기초

```
class ResultViewController: UIViewController {  
  
    @IBOutlet weak var dataLabel: UILabel!  
    var message: String?  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        setMessageInLabel()  
    }  
  
    func setMessageInLabel(){  
        if let msg = message {  
            dataLabel.text = msg  
            dataLabel.sizeToFit()  
        }  
    }  
  
    @IBAction func touchUpToGoBack(_ sender: Any) {  
        self.dismiss(animated: true, completion: nil)  
    }  
}
```



옵셔널 바인딩으로
안전하게 값을 꺼낸 뒤
dataLabel의 text로 넣습니다

○ 데이터 전달 기초

```
@IBAction func touchUpToSendData(_ sender: Any) {  
    guard let nextVC =  
        self.storyboard?.instantiateViewController(withIdentifier:  
        "ResultViewController") else {return}
```

Summary

No overview available.

Declaration

```
let nextVC: UIViewController
```

Declared In

FirstViewController.swift

nextVC의 타입 : UIViewController

```
@available(iOS 5.0, *)  
open class UIStoryboard : NSObject {  
  
    public /*not inherited*/ init(name: String, bundle  
        storyboardBundleOrNil: Bundle?)  
  
    open func instantiateInitialViewController() -> UIViewController?  
  
    open func instantiateViewController(withIdentifier identifier: String)  
        -> UIViewController  
}
```

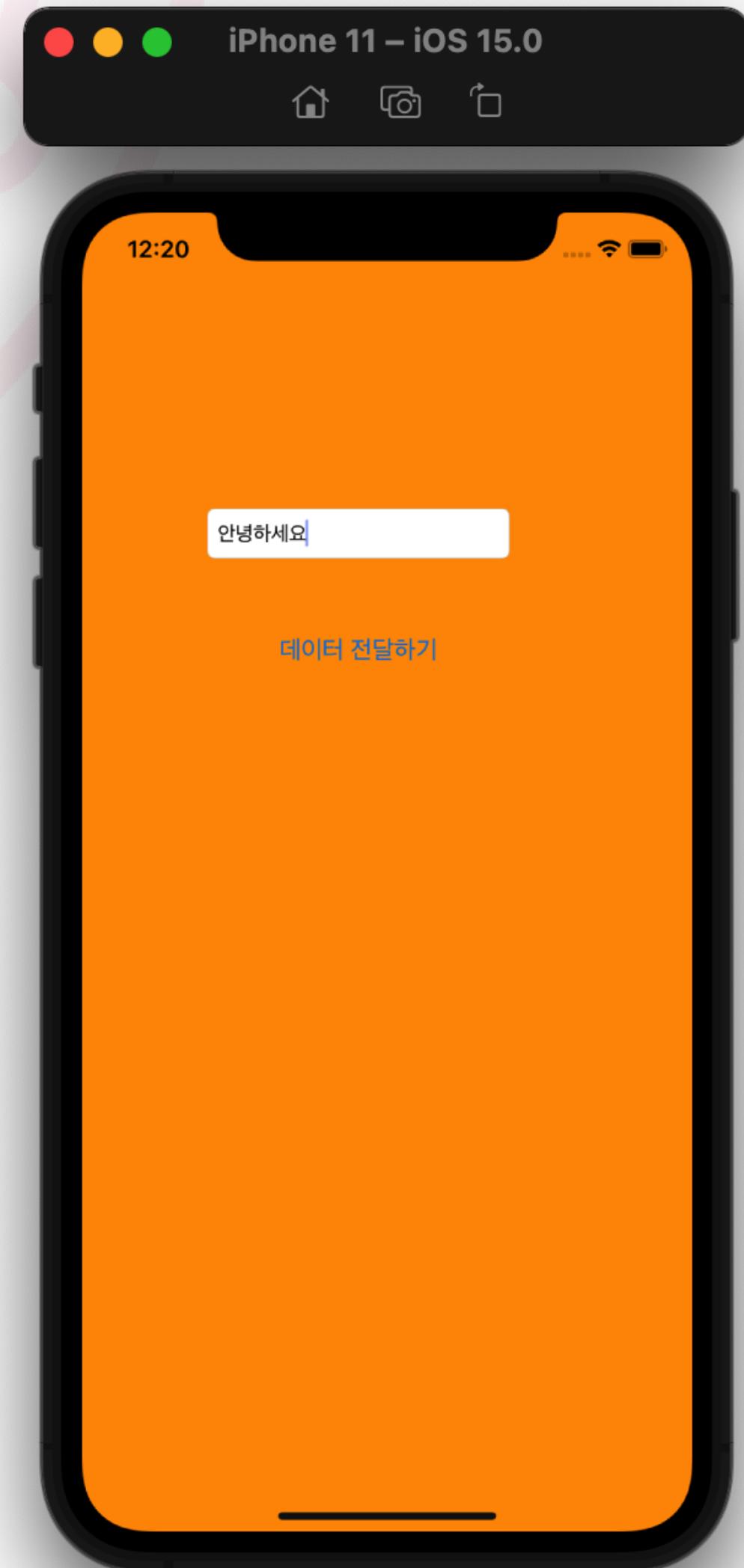
storyboard의 instantiateViewController의 return값이 ViewController 이기 때문입니다!

```
@IBAction func touchUpToSendData(_ sender: Any) {  
    guard let nextVC = self.storyboard?.instantiateViewController(withIdentifier:  
        "ResultViewController") else {return}  
  
    nextVC.message  
}  
✖ Value of type 'UIViewController' has no member 'message'
```

우리는 ViewController를 상속받은 ResultViewController에 message란 프로퍼티를 넣었는데
부모인 ViewController에게 자식의 프로퍼티를 묻는 것과 동일합니다.

```
@IBAction func touchUpToSendData(_ sender: Any) {  
    guard let nextVC = self.storyboard?.instantiateViewController(withIdentifier:  
        "ResultViewController") as? ResultViewController else {return}  
  
    nextVC.message = dataTextField.text  
    nextVC.modalPresentationStyle = .fullScreen  
    self.present(nextVC, animated: true, completion: nil)  
}
```

○ 데이터 전달 기초



○ 감사합니다

다음주, 2차 세미나에서 만나요!

긴 시간동안 너무 고생많으셨습니다!
맛있는 저녁 드세요!



피드백

1차 세미나 피드백

더 좋은 iOS 파트를 위해 소중한 의견 꼭 부탁드리겠습니다 ❤️



피드백

1차 세미나 피드백

더 좋은 iOS 파트를 위해 소중한 의견 꼭 부탁드리겠습니다 ❤️

