



iOS Seminar 0

금진섭

# 세미나장 소개

---

금진섭

와플스튜디오 18기



software engineer (NFT 메타버스 플랫폼)

Team urban - SNUTT 리뉴얼 iOS 개발

# 세미나 소개

---

iOS Development

Xcode12

Language : Swift5

UIKit + Storyboard

# 세미나 소개

---

0주차 : 세미나 개요 / iOS 소개 / MVC / Storyboard를 이용한 UI 개발

1주차 : TableView - To Do List 앱 / delegate 패턴

2주차 : Networking / Data storage - 로그인 구현하기

3주차 : 다양한 View class / animation

4주차 : CollectionView - 갤러리 앱 / 모듈화 / MVVM

5주차 : RxSwift, arc, multi thread, app delegate, scene delegate

# iOS 란?



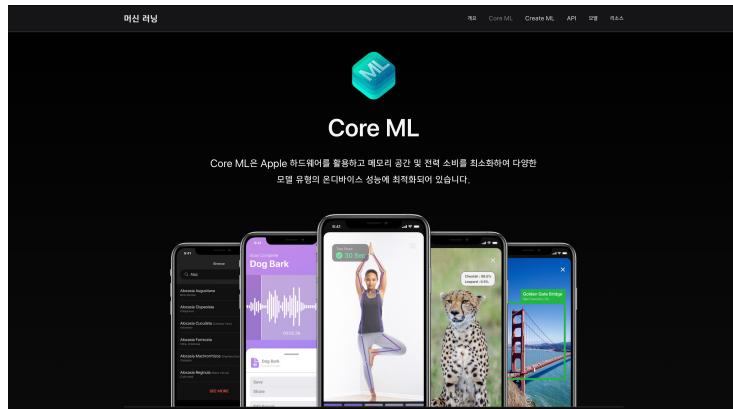
아이폰에서 사용하는 OS

Unix 계열 OS

애플이 개발하며 매년 버전이 업데이트됨. 올해 iOS 15 출시 예정

iOS 기반의 iPadOS, watchOS

# iOS 란?



iOS 개발을 통해 머신 러닝을 돌릴 수도 있고,  
AR도 구현 가능하고,  
게임도 만들 수 있음



\*99달러 개발자 멤버십에 가입해야 앱스토어에 배포 가능

# iOS layer

---

Cocoa Touch

Foundation, UIKit 등

Media

OpenGL, Core Audio, Core Animation, JPEG 등

Core Service

SQLite, Core Location, Networking 등

Core OS

File System, Certificates, KeychainAccess 등

# iOS Key Concepts

---

Swift - 객체 지향 언어

UIKit - iOS 빌드에 필요한 핵심 객체들 제공

Foundation - 데이터 타입, 알고리즘 등 Swift로 구현된 기본적인 객체들

Xcode - iOS 개발을 위해 필요한 editor

# Swift

---

2014년 Apple에서 소개된 언어 (그 이전에는 Objective-C)

객체 지향 언어 - iOS의 모든 코어 프로그램은 객체로 이루어져 있다

강타입 언어

Swift 5

<https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html>

Swift가 처음이신 분들은 꼭 위 Swift 공식 문서를 정독하는 걸 권합니다!

# UIKit vs SwiftUI

---

SwiftUI : 2019년 iOS 13과 함께 등장한 새로운 UI Framework

선언형 UI / MVVM

컴포넌트 중심의 React 개발 방식과 많이 유사한 방식. Android의 Compose

SwiftUI는 아직 프로덕트 용으로는 잘 사용하지 않음. 실험적

큰 프로젝트는 계속 UIKit 방식으로 진행됨. 하지만 앞으로 몇 년 후에는 SwiftUI가 상용화될 수 있을 것

# iOS 레퍼런스

---

Stanford iOS lecture

2021 - SwiftUI

<https://www.youtube.com/watch?v=bqu6BquVi2M&list=PLpGHT1n4-mAsxuRxVPv7kj4-dQYoC3VVu>

2017 - UIKit

<https://www.edwith.org/swiftapp/lecture/26619>

# iOS 레퍼런스

---

애플 개발자 문서

<https://developer.apple.com/documentation/>

WWDC

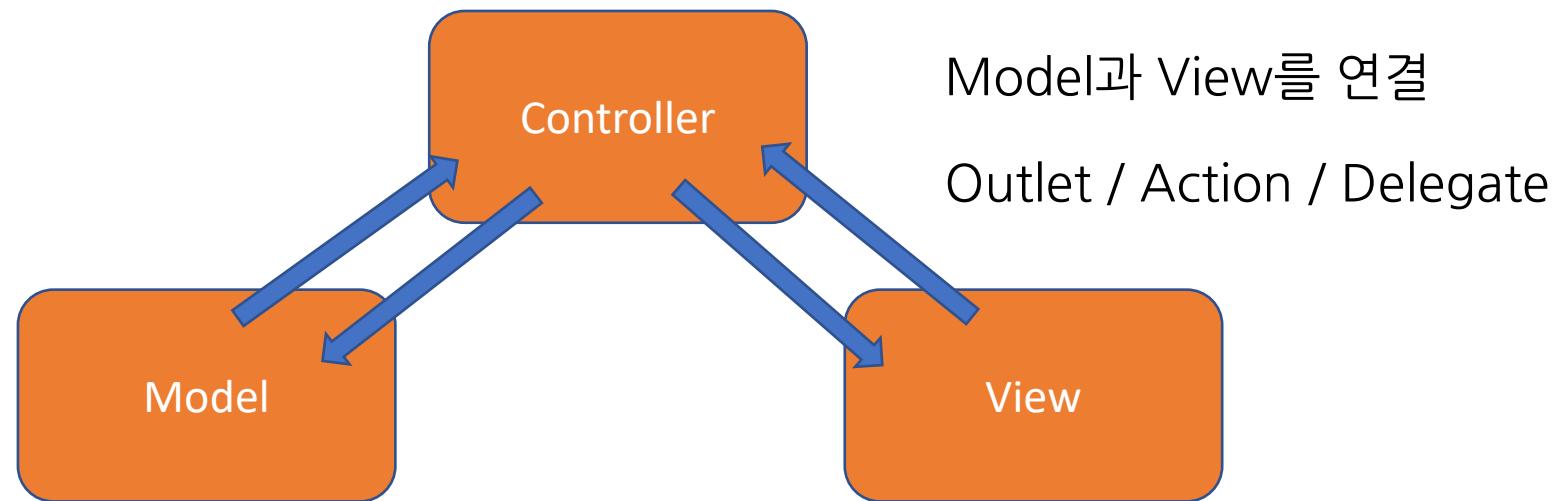
<https://developer.apple.com/wwdc21/>

Human Interface Guidelines

<https://developer.apple.com/design/human-interface-guidelines/>

# MVC Design Pattern

iOS UIKit 디자인 패턴



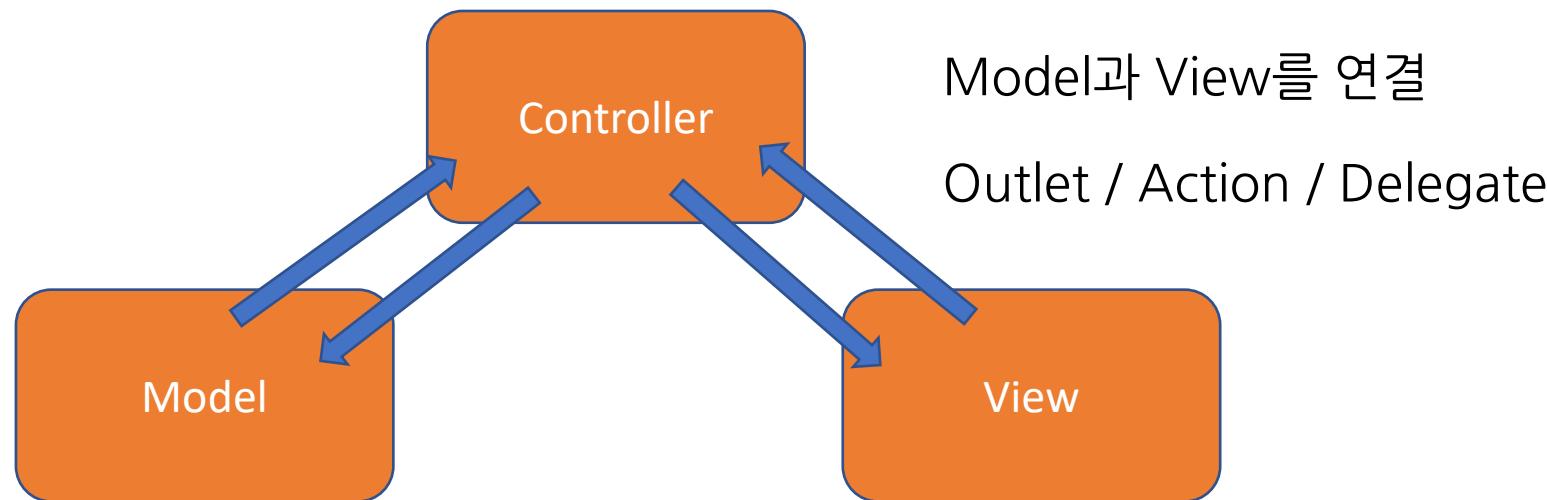
데이터를 정의하고 비즈니스 로직 처리

사용자 인터페이스에 보여지는 View 객체를 정의

# MVC Design Pattern

권장 사항이 아니라 반드시 따라야 하는 디자인 패턴

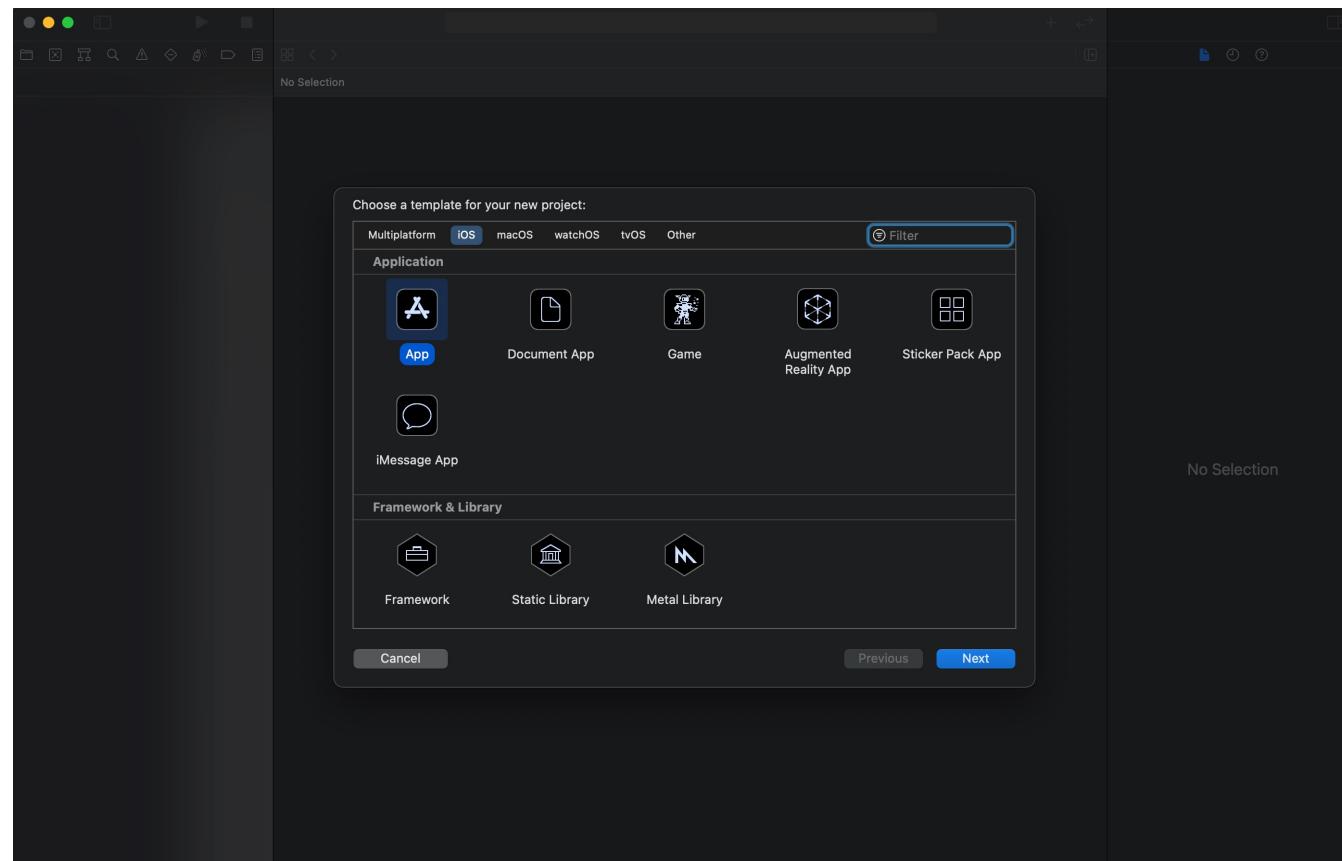
-> 새로운 도메인을 추가해 MVVM / Clean Architecture 등의 디자인 패턴으로 사용하기도 함



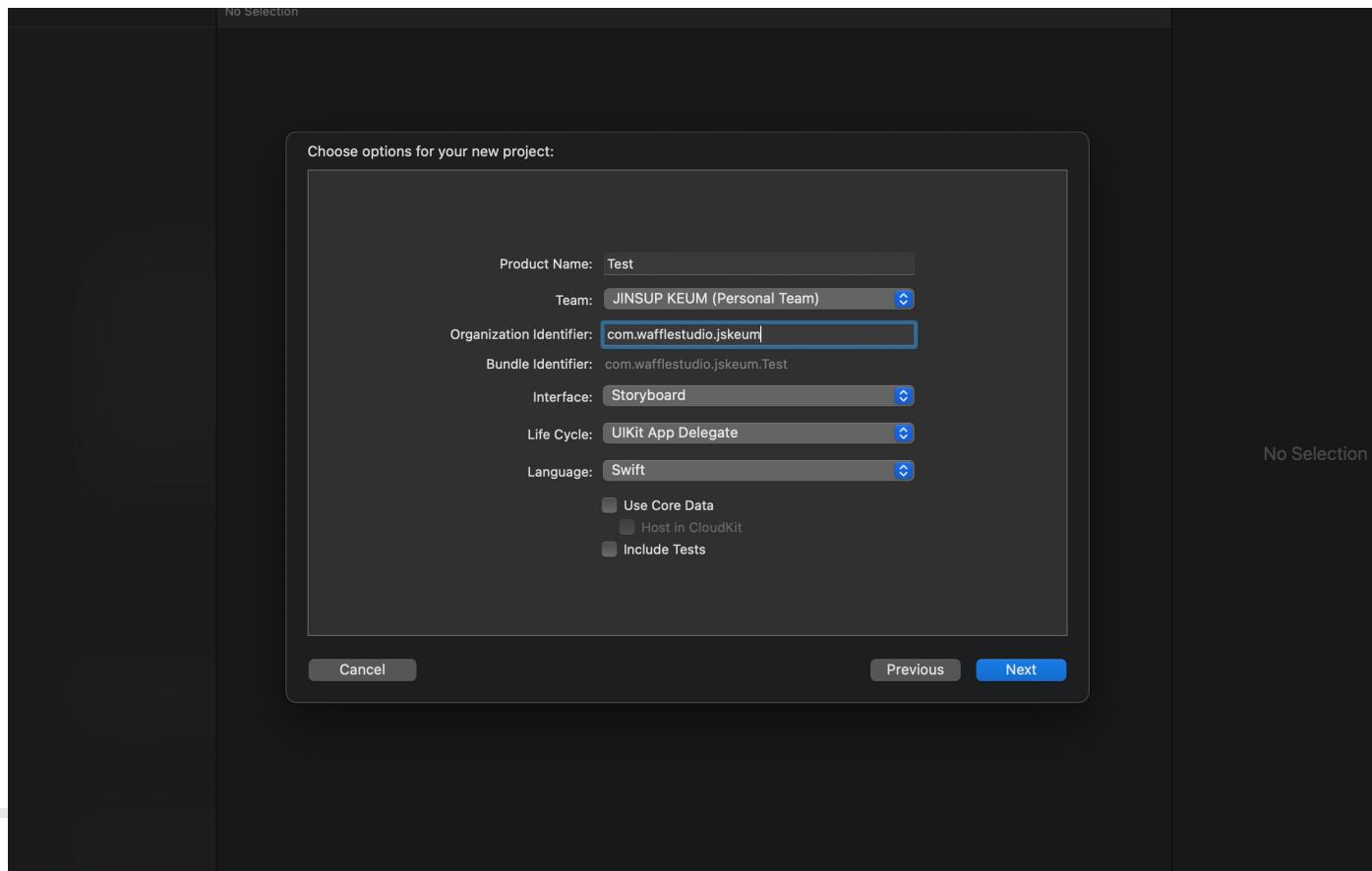
데이터를 정의하고 비즈니스 로직 처리

사용자 인터페이스에 보여지는 View 객체를 정의

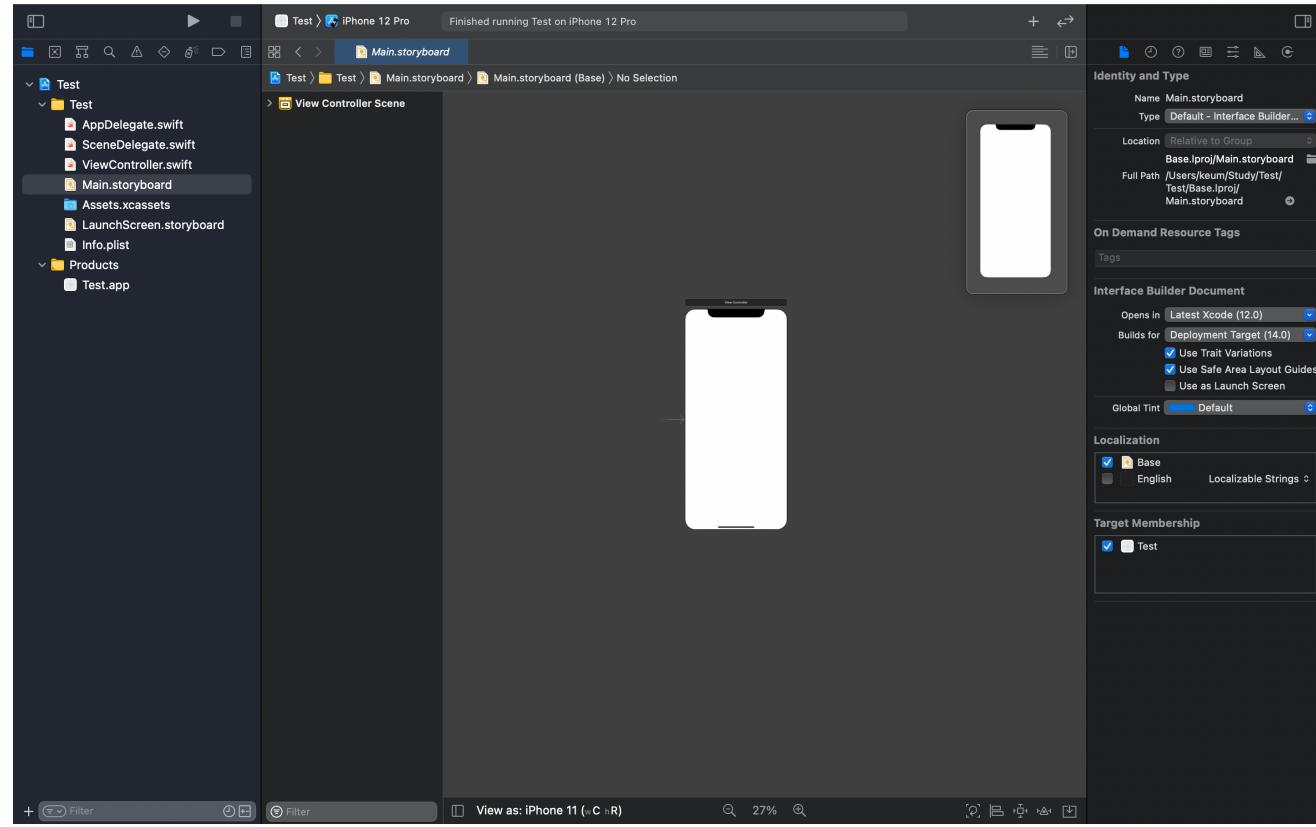
# iOS 프로젝트 실습



# iOS 프로젝트 실습



# iOS 프로젝트 실습



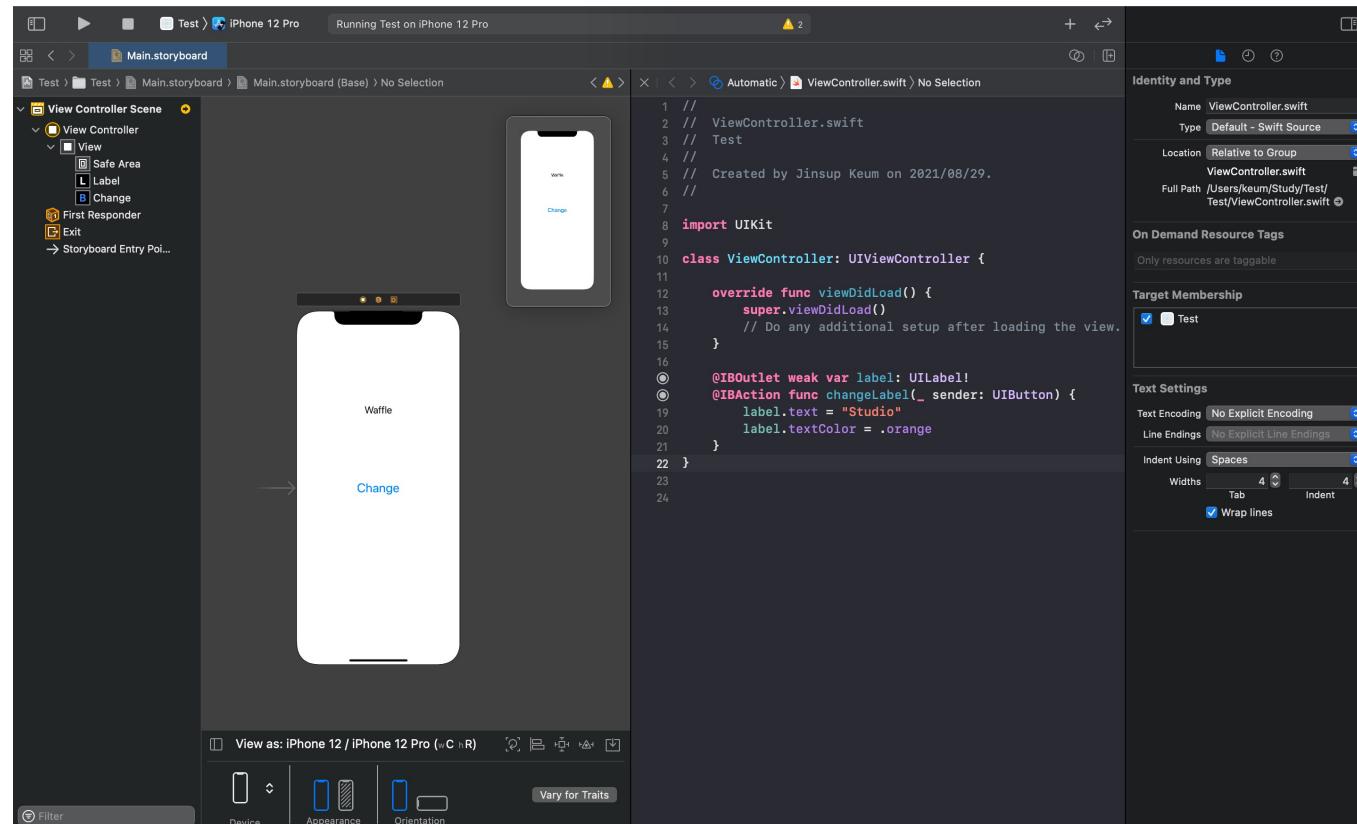
## Storyboard

: View 객체를 정의할 수 있도록  
하는 Interface

협업 시 Storyboard는 충돌되기  
쉽고, diff를 확인하기 어려움.

1인 1 Storyboard 또는 No  
Storyboard를 하기도 함

# iOS 프로젝트 실습



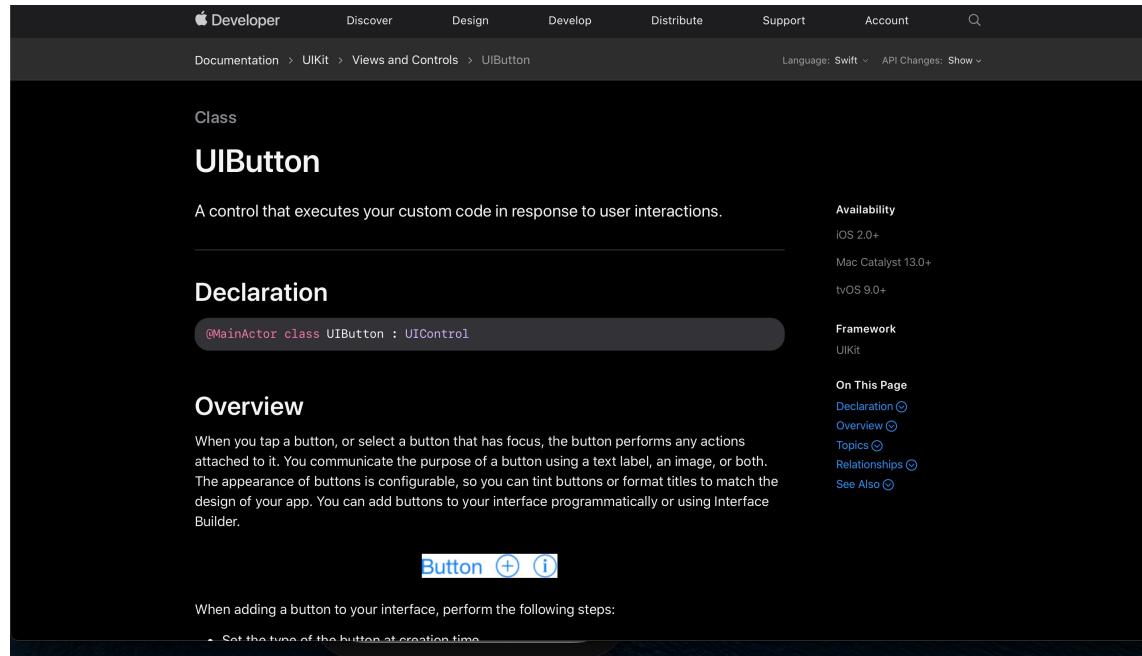
## Action

: 특정한 이벤트(터치와 같은)가 발생했을 때 실행되는 함수

## Outlet

: Storyboard에 정의된 객체에 대한 참조

# iOS 프로젝트 실습



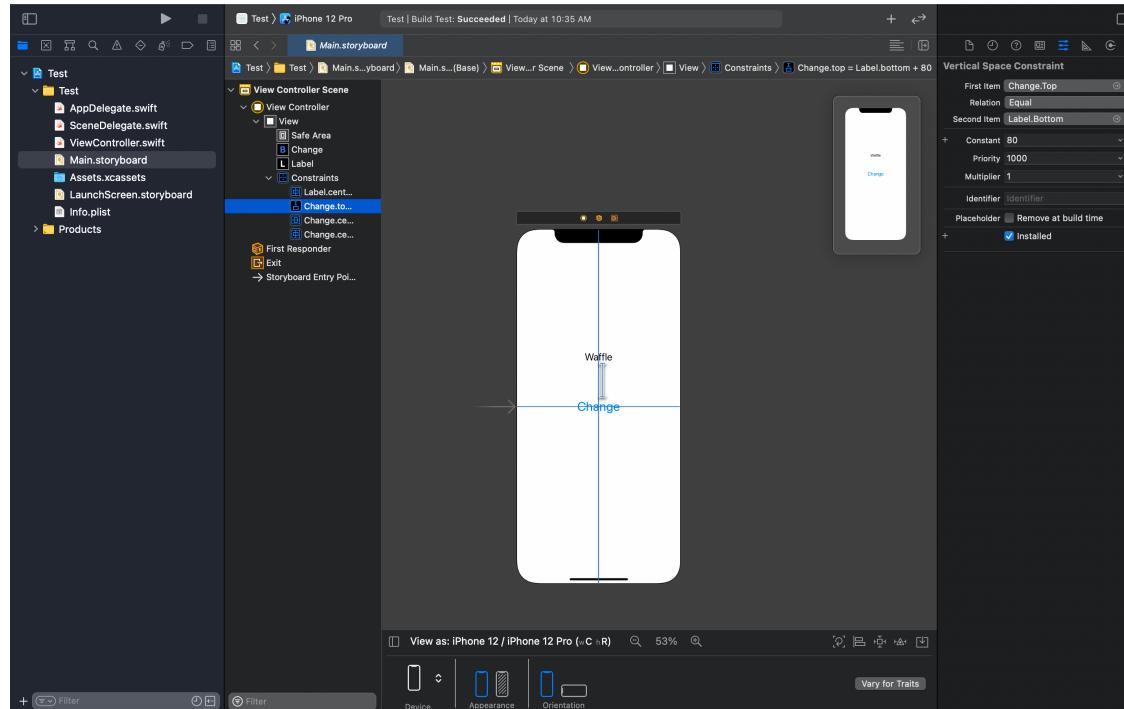
## UIButton

: UIControl을 상속

(UIControl은 UIView를 상속)

모든 UIKit 클래스에 대한 정의를  
애플 개발자 페이지에서 확인 가능!

# iOS 프로젝트 실습



## Auto Layout

: UIKit 앱은 뷰의 레이아웃을 정하기 위해 constraint를 이용

Auto Layout은 정의된 constraint에 따라 View의 위치와 크기를 계산

# iOS 프로젝트 실습

```
28
29 @IBAction func moveToNext(_ sender: UIButton) {
30     let identifier = "SecondViewController"
31
32     guard let controller =
33         storyboard?.instantiateViewController(identifier: identifier) as?
34         SecondViewController else { return }
35
36     navigationController?.pushViewController(controller, animated: true)
37 }
```

## Navigation Controller

: 뷰의 계층 관계를 만들어 줌

# Assignment 0

---

## 로그인 뷰 만들기

- 목표 : Xcode와 Swift, iOS 기본 UI 객체와 친해지자!
- 구현 내용
  - 메인 뷰에서 username, email, password를 입력 받고 로그인 버튼을 누르면 navigation controller를 통해 다음 뷰로 이동해 username과 email 보여주기.
  - MVC 패턴을 이용해야 함
  - 모든 View는 Auto layout이 적용되어 있어야 함
  - username이 2자 이하이면 “username은 2자 이상입니다” Alert View 띄워주기
  - \*User 모델 구현 시 password는 property로 넣지 않음