



**iOS Seminar 0**

**한상현**

# 세미나장 소개 - 한상현

---

약 2년차 주니어 iOS 개발자  
아직도 열심히 깨부하는 중... 배울 게 너무 많습니다 TTT

The logo for UPbit, featuring the text "UPbit" in a bold, blue, italicized sans-serif font. The "UP" is in a lighter blue and the "bit" is in a darker blue.

# 오늘 배울 내용

---

- iOS 개발에 대한 개괄
- 개발 공부를 하면서 가지면 좋은 마인드와 방법론
- 과제에 필요한 기본적인 내용
- 과제 설명

# iOS 개발을 해보자

- Xcode (3<sup>rd</sup> party AppCode)
- 아이폰용 os  
: 패드는 iPadOS / 맥은 MacOS / 워치는 WatchOS
- Xcode 하나에서 애플 생태계 하드웨어에 올라가는 모든 소프트웨어 개발 가능
- 애플은 신이고 애플 개발자 사이트는 성경입니다  
(우리 팀장님 왈)



# iOS 개발의 장단점

---

- 장점

1. 환경의 단일화 - 애플이 하라는대로 하기만 하면 된다
2. 앱등이만 하기 때문에 개발자가 귀하다. 어차피 결국 모든 기업은 네이티브 개발자가 필요하기 때문

- 단점

1. 다른 스택을 공부하면 열받음. 다 너무 중구난방이라서
2. 애플이 자꾸 이상한 걸 내놓는데, 못 따라가면 바로 도태됨

# 항상 봐야할 레퍼런스

---

- 애플 개발자 사이트  
<https://developer.apple.com/>  
<https://developer.apple.com/documentation/>
- Swift Document  
<https://docs.swift.org/swift-book/>  
<https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html> -> Swift 안 익숙하시면 정독을 권합니다
- WWDC : 실제 용례에 대한 설명이 많음 + 신기술 / 응용법  
<https://developer.apple.com/wwdc22/>
- Human Interface Guideline  
<https://developer.apple.com/design/human-interface-guidelines/guidelines/overview/>  
: 프론트엔드 UI/UX의 교과서

# 깨부하는 법

---

1. 애플 공식 문서 + 스위프트 공식 문서
2. WWDC 영상
3. StackOverflow + raywenderlich + 각종 블로그  
: 항상 잘 찾아보고 신빙성 있는 저자의 기록인지 확인이 필요

+a : 책을 사서 보는 것도 좋음 -> 시간이 많이 드는게 문제인데 확실히 한 권 읽고 나면 정리가 잘 되는게 있음

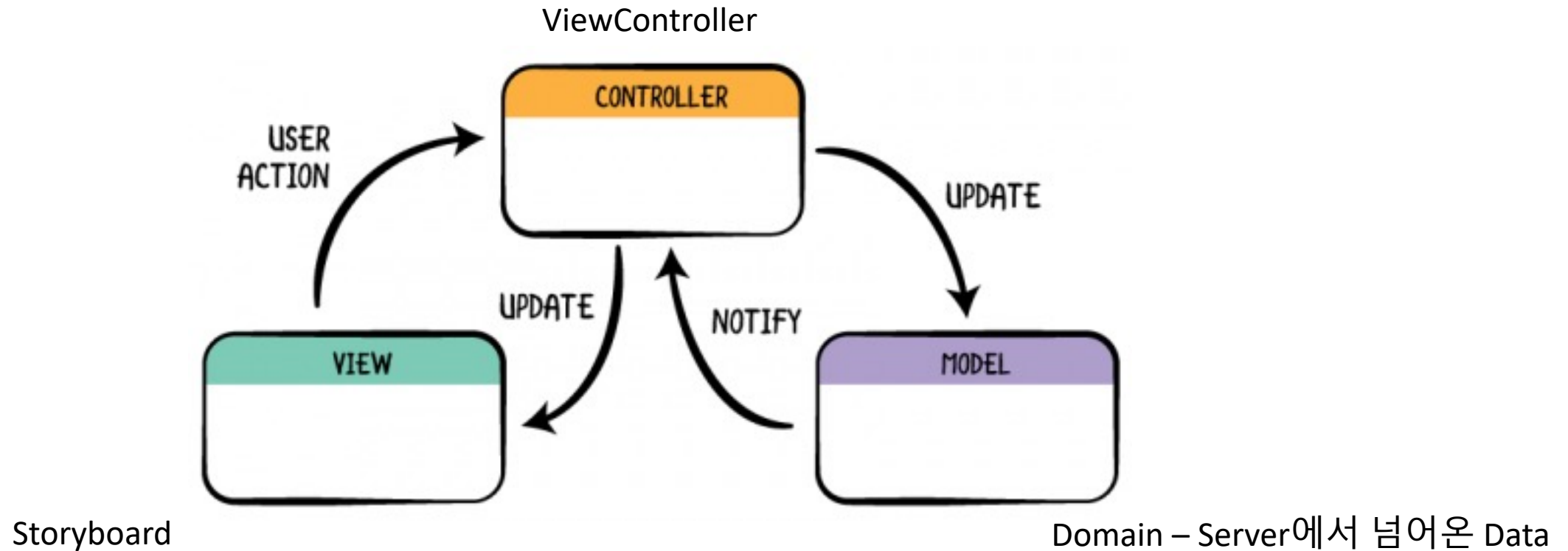
# 기본적인 구조 - 옛날 MVC, 지금 MVVM

---

- MVC : Model (데이터 구조) - View (레이아웃) - Controller (데이터 처리 + 레이아웃)
- MVVM : Model (데이터 구조) - View (ViewController - 레이아웃) - ViewModel (데이터 처리)



# MVC - 우린 이거 안 쓸거예요



# MVVM - 이걸 씁니다

Model : UseCase, Manager 등이 포함된 도메인 단 로직 + Repository로 API 통신



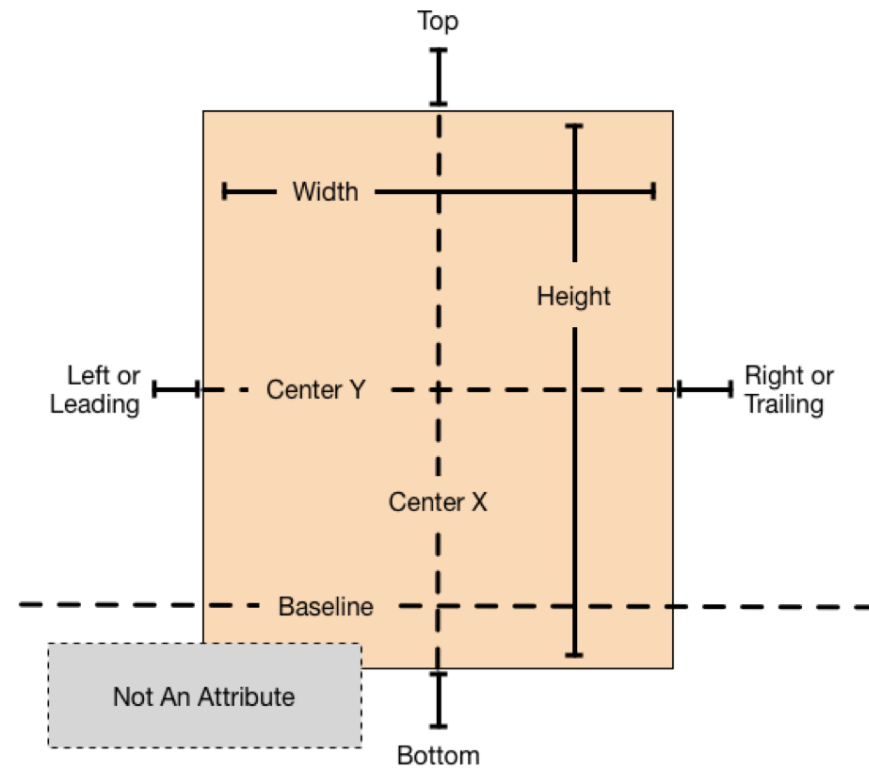
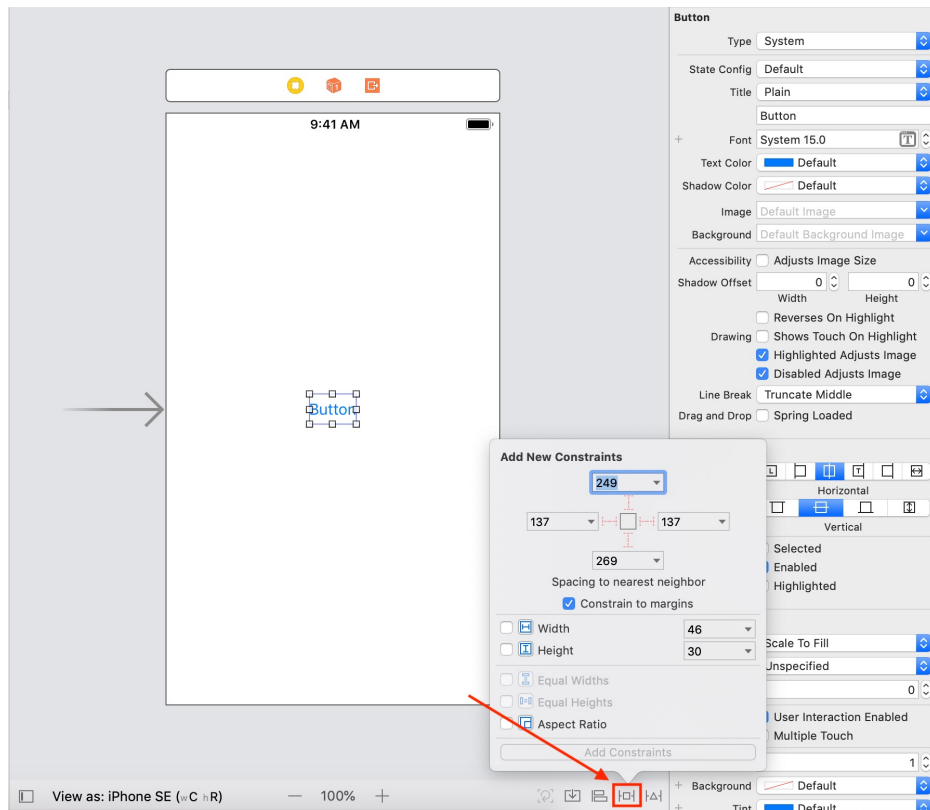
# 환경 세팅

---

- **Xcode 13.4.1**로 진행
  - : 아마 우리 세미나 하면서 14가 배포되긴 할텐데, concurrency 관련해서 변경점이 무척 많아서 컴파일 문제가 생길 가능성이 높음.
  - : 앱스토어 말고 웹에서 별도로 다운 받아서 설치하시는걸 추천드립니다
- **Swift 5**
  - : 같은 이유
- **UIKit + MVVM (RxSwift)**
  - : 현재 가장 널리 쓰이는 조합
  - : 앞으로 SwiftUI + Combine + Concurrency 조합이 대세가 될 듯 하기는 하나 일단...

# AutoLayout

<https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/AnatomyofaConstraint.html>



# 다른 뷰를 띄우려면?

---

- `parentVC.present`  
`childVC.dismiss`
- VC가 UINavigationController에 포함되어 있을 경우  
: `navigationController.pushViewController(childVC)`

# Storyboard 없이 앱을 시작하려면...?

---

- AppDelegate / SceneDelegate에 대한 이해가 필요
  - > iOS 13이전에는 모든걸 AppDelegate가 관리했으나 iOS 13부터는 화면 관련 로직은 SceneDelegate가 관리
  - > 시작 뷰는 SceneDelegate에서 Window를 생성하며 시작 (구글링!)

# Xcode를 봅시다

---



# 앞으로의 과제 / 출석 관련 공지

- 출석은 대면 / 비대면 무관하게 제 시간에 확인만 되시면 인정해드립니다
- 지각 2회 -> 결석 1회로 잡겠습니다 (지각은 10분 이후부터)
- 결석 2회 시 세미나 과제 제출 여부와 무관하게 탈락시킬 예정입니다 (최소한의 참여율)
- 과제는 돌아가기만 하면 만점
  - : 뭔가 잘못 짜여서 퍼포먼스가 안 좋거나, 주요 기능은 되지만 버그가 조금 있는 정도는 괜찮아요
  - : grace day는 전 세미나 합쳐서 3일 (사용 시 깃헙 PR 코멘트에 적어주세요)
  - : 매 과제 추가 과제가 있을 예정이고, 구현 시 grace day 1일 드립니다
  - : 추가 과제는 수행하지 않아도 괜찮습니다



# Assignment 0

---

- 로그인 / 로그아웃 화면 만들어보기

## 1. 홈 화면

- ID / Password 입력 필드, 하단 로그인 버튼
- 로그인 버튼 탭 시 다음 화면 (로그인 화면) 으로 넘어감

## 2. 로그인 화면

- ID / Password 노출, 하단 로그아웃 버튼
- 로그아웃 버튼 탭 시 홈 화면으로 돌아감

# Assignment 0

---

- 추가 과제 -> 안 해도 무방, 하면 grace day 1일  
: 앱을 껐다가 키더라도 로그인 중이라면 로그인 상태를 유지  
-> 바로 로그인 화면 노출해야 함  
: UserDefaults (구글링을 잘 해보세요)
- 참고하면 좋은 키워드  
: UITextField, UIButton, UINavigationController, present, dismiss, UILabel

# 혼자 공부해보면 좋은 내용

---

- 공식 문서 :  
[https://developer.apple.com/documentation/uikit/views\\_and\\_controls/table\\_views](https://developer.apple.com/documentation/uikit/views_and_controls/table_views)
- RxDataSource (나중에 가르쳐 드릴 예정) :  
<https://github.com/RxSwiftCommunity/RxDataSources>
- Rx로 TableView 만들기 :  
<https://eunjin3786.tistory.com/29>
- TableView 개념 익히기 : <https://zeddios.tistory.com/55?category=682195>
- Codable 심화 :  
<https://minsone.github.io/programming/swift-codable-and-exceptions-extension>