

Week 2-1

WWDC Sessions

P.O.P. 간단 예제 코드

포트폴리오

0. 포트폴리오란

1. 프로젝트의 기술 정보 및 아키텍처에 관한 내용들을 넣습니다
2. 상세 화면마다 스크린샷과 디테일한 기능 설명을 해줍니다
3. 기타 프로젝트 이슈를 적습니다

Week 2-1 P.O.P, Swift Performance

- Swift 언어의 특성을 반영한 프로그래밍
- Protocol Oriented Programming 개념 습득 및 활용
- Swift 언어의 성능향상을 고려한 코드작성
- [아하!모먼트]내가 찾는 그것이 있는 곳 - 공식문서, WWDC (좋은 세션, 문서 소개하기)
- [아하!모먼트]라이브 이력서 검토 III

WWDC Sessions

- 매 년 What's new in xxx
 - 예) What's new in Swift, What's new in UIKit

연도	카
2015	Sw
2016	Sw
2016	De
2016	In n
2016	De
2016	Xc
2016	De
2016	De
2016	Sw
2016	Sw
2016	Sw
2017	De
2017	In n
2017	Ac
2017	Ac
2017	Te
2017	Ul
2017	De
2017	De
2018	Ul
2018	Sw
2018	Sw
2018	De
2018	De

2018	De
2018	Te
2018	Pe
2018	Un
2019	Sw
2019	De
2020	Te
2020	Ac
2020	Sw
2020	Sw
2020	Un
2020	Un
2020	Un
2020	Un
2020	Ac
2020	Un
2020	Sw
2020	Un
2020	Un
2020	Un
2020	De
2020	Te
2020	Sw
2020	Un
2020	Sw
2020	Un
2020	Un

P.O.P. 간단 예제 코드

```
protocol Flyable { func fly() } protocol Runnable { func run() }  
protocol Swimbable { func swim() } extension Runnable { func run() {  
    print("유 배러 런 런 런 런 런") } } extension Swimbable { func swim() {  
    print("나는 배영을 잘함") } } struct Person: Runnable, Swimbable { func  
    run() { print("헉헉") } } class Frog: Runnable, Swimbable { func  
    swim() { print("개굴개굴") } } struct Bird { } Person().swim() // 나는  
배영을 잘함 Frog().swim() // 개굴개굴 Person().run() // 헉헉
```

포트폴리오

0. 포트폴리오란

- 지원하는 분야에 대해 자신이 어떤 노력을 해왔는지, 또 어떤 성과가 있었는지를 기술하는 문서입니다
- 특히 iOS 개발을 위해 다양한 방법으로 공부를 해왔을텐데, 이를 잘 녹여내면 됩니다

- 프로젝트를 몇 개 했는지는 중요하지 않습니다. 무엇을 고민했고 어떻게 얼마나 성장했는지가 나타나야 합니다.
 - 2~3개면 충분합니다
 - 코드에 대한 스스로의 철학이 나타나면 매우 좋습니다
 - 갯수를 채우고 싶다면 서로 다른 서비스를 많이 만들지 않고, 하나의 서비스를 다양한 방법으로 만들어보도록 합니다



하나의 앱을 아래처럼 다양한 방법으로 만들어보기

1. UIKit + Delegation pattern + MVC
2. UIKit + RxSwift + MVVM
3. SwiftUI + Combine + MVVM
4. SwiftUI + RxSwift + MVC

1. 프로젝트의 기술 정보 및 아키텍처에 관한 내용들을 넣습니다

- 항목 별로 어떤 기술을 사용했는지 나열하면 좋습니다
 - 항목을 자세하게 나눌수록 적을 게 많아집니다
 - 보는 사람 입장에서는 더 자세하게 확인할 수 있고, 그만큼 믿음을 줄 수 있습니다



- 화면구현: UIKit, MapKit
- 네트워킹: URLSession, URLSessionProtocol
- 비동기 프로그래밍: DispatchQueue, DispatchSemaphore, DispatchGroup
- 의존성관리도구: CocoaPods, Swift Package Manager
- 컨벤션: SwiftLint
- 테스트: XCTest(Unit Test, UI Test)
- 사용한 디자인패턴: MVVM, Observer, Coordinator, Decorator

2. 상세 화면마다 스크린샷과 디테일한 기능 설명을 해줍니다

- 해당 화면을 구현하기 위해 어떤 기술을 사용했는지(예: UITableView, CoreAnimation 등)를 써주면 좋습니다
- 화면을 구현할 때 무슨 문제가 있었고, 어떻게 해결했는지 기술합니다

3. 기타 프로젝트 이슈를 적습니다

- 고민 및 갈등이 있었다면 적습니다
 - a. 특히 어떤 문제가 있었는지
 - b. 어떻게 해결했는지