

# Swift로 CrossPlatform하기

Let'Swift Speaker Jihoon

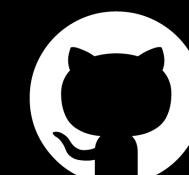
# 프로필

Jihoon



## 안지훈 (Jihoon)

- 다양한 방식의 접근을 좋아하는 개발자
- 마이너한거 좋아함
- Swift 4년째 사용중



[github.com/Jihoonahn](https://github.com/Jihoonahn)



@jihoonahn



Let'Swift 2023  
Deep Dive into the unknown

## Contents

1. Swift로 Cross Platform을 하게된 배경
2. What is Scade
3. Scade는 무엇을 제공해줄까요?
4. Scade 설정 방법
5. Scade 시작해볼까요?
6. Scade는 어떻게 동작해요?
7. 마무리



# Swift로 Cross Platform을 하게된 배경



**Let'Swift 2023**  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경

혼자 프로젝트를 진행하면서



**Let'Swift 2023**  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경

만들던 iOS 앱 프로젝트가  
안드로이드에서도 실행됐으면 좋겠는데..

그런데, 두개 다 만들었다고 해도,  
이걸 내가 두개 다 관리할 시간이 생길까?



# Swift로 Cross Platform을 하게된 배경

## CrossPlatform을 해야하나?



**Let'Swift 2023**  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경

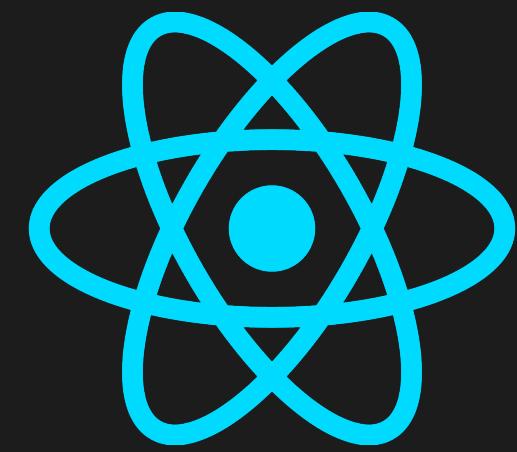


Kotlin Multiplatform

Kotlin

네이티브 코드 사용

네이티브 UI 구성 요소 사용



React Native

Js, Ts

Bridge를 이용한 네이티브 모듈

네이티브 UI 구성 요소 사용



Flutter

Dart

네이티브 코드 사용

자체 UI 렌더링 엔진 사용



Let'Swift 2023  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경

RN이나 Flutter 배우면 기존에 Swift만큼의 숙련도를 낼 수 있을까..

다른 언어는 언제 배워..



새로운 IDE 익숙해지는거  
시간이 걸릴텐데..



Let'Swift 2023  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경

iOS, Android 동시 개발을 통한 생산성 향상



기존 언어를 이용해서 학습에 필요한 리소스 최소화



Let'Swift 2023  
Deep Dive into the unknown

# Swift로 Cross Platform을 하게된 배경



# What is Scade?



**Let'Swift 2023**  
Deep Dive into the unknown

# What is Scade?



[ S C A D E ]

## Scade

- Swift 5.8 지원
- 동일한 소스 코드를 사용하여 iOS  
Android 동시 개발
- 순수 Swift 및 Swift Foundation과  
Swift Dispatch를 사용하는 모든  
Swift 라이브러리 사용 가능



Let'Swift 2023  
Deep Dive into the unknown

# What is Scade?

Swift 3.1

2017년 5월

Swift 4

2017년 11월

Swift 4.1

2018년 5월

Swift 4.2.1

2019년 2월

Swift 5

2019년 8월

Swift 5.2

2020년 7월



**Let'Swift 2023**  
Deep Dive into the unknown

# What is Scade?

2023. 8 (최신 업데이트: v2.3.1 Beta)

현재 지원

- Swift 5.8 지원
- Xcode 14 지원
- 맞춤형 LSP
- Android용 URLSession에 비동기 기능 지원

...



**Let'Swift 2023**  
Deep Dive into the unknown

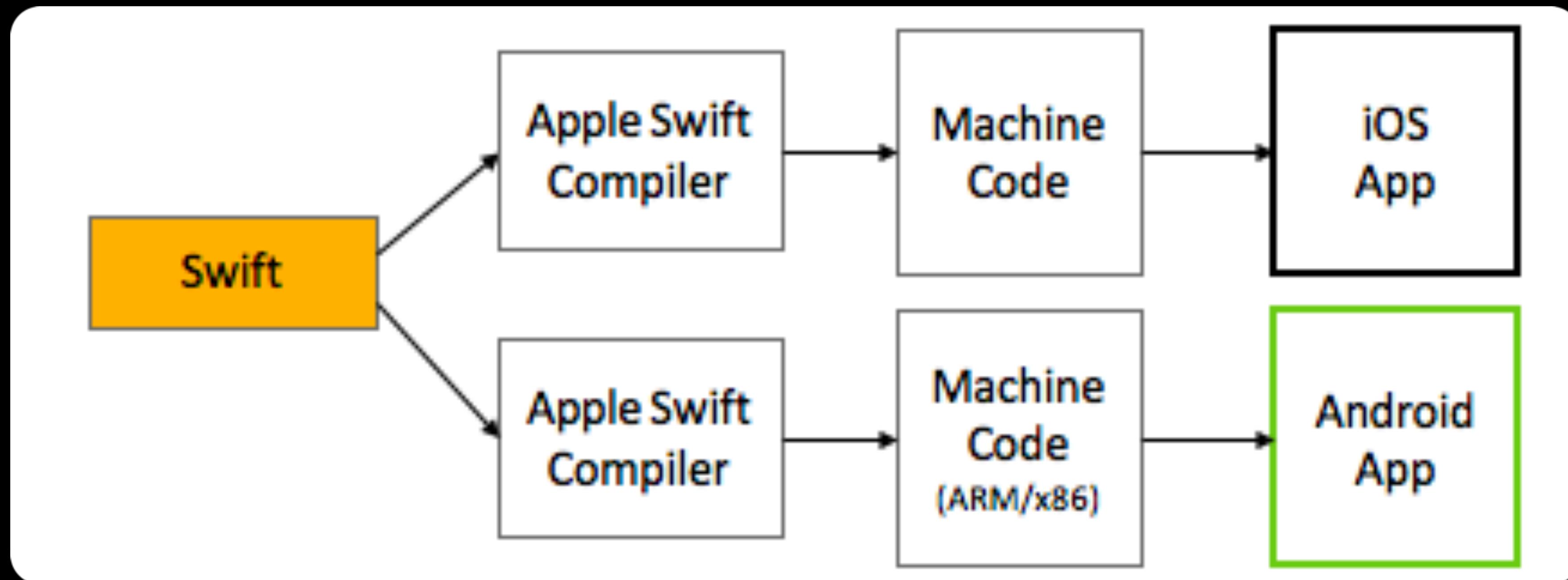
# Scade는 무엇을 제공해줄까요?



**Let'Swift 2023**  
Deep Dive into the unknown

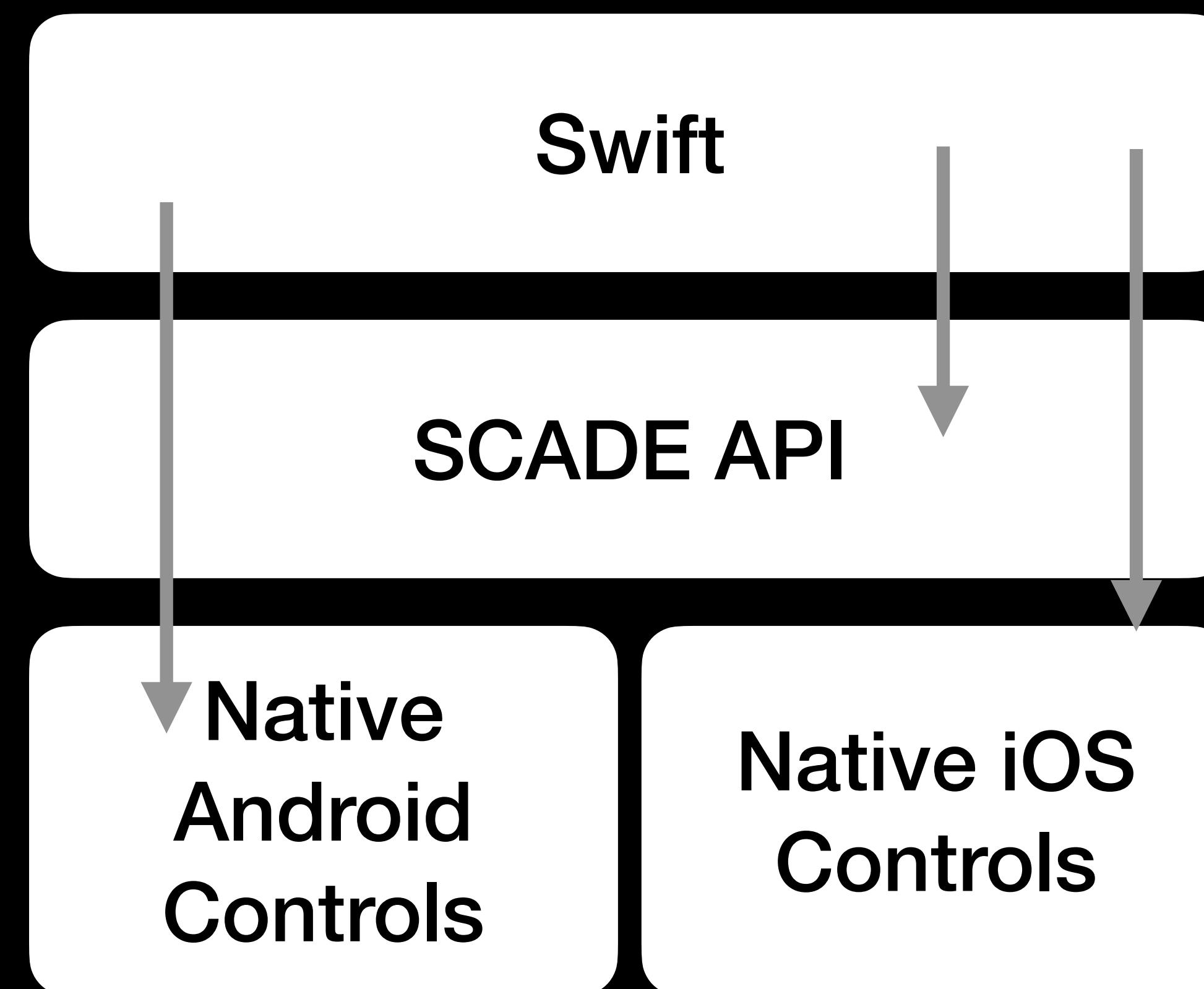
# Scade는 무엇을 제공해줄까요?

Swift Compiler for Android



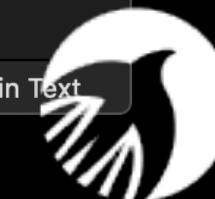
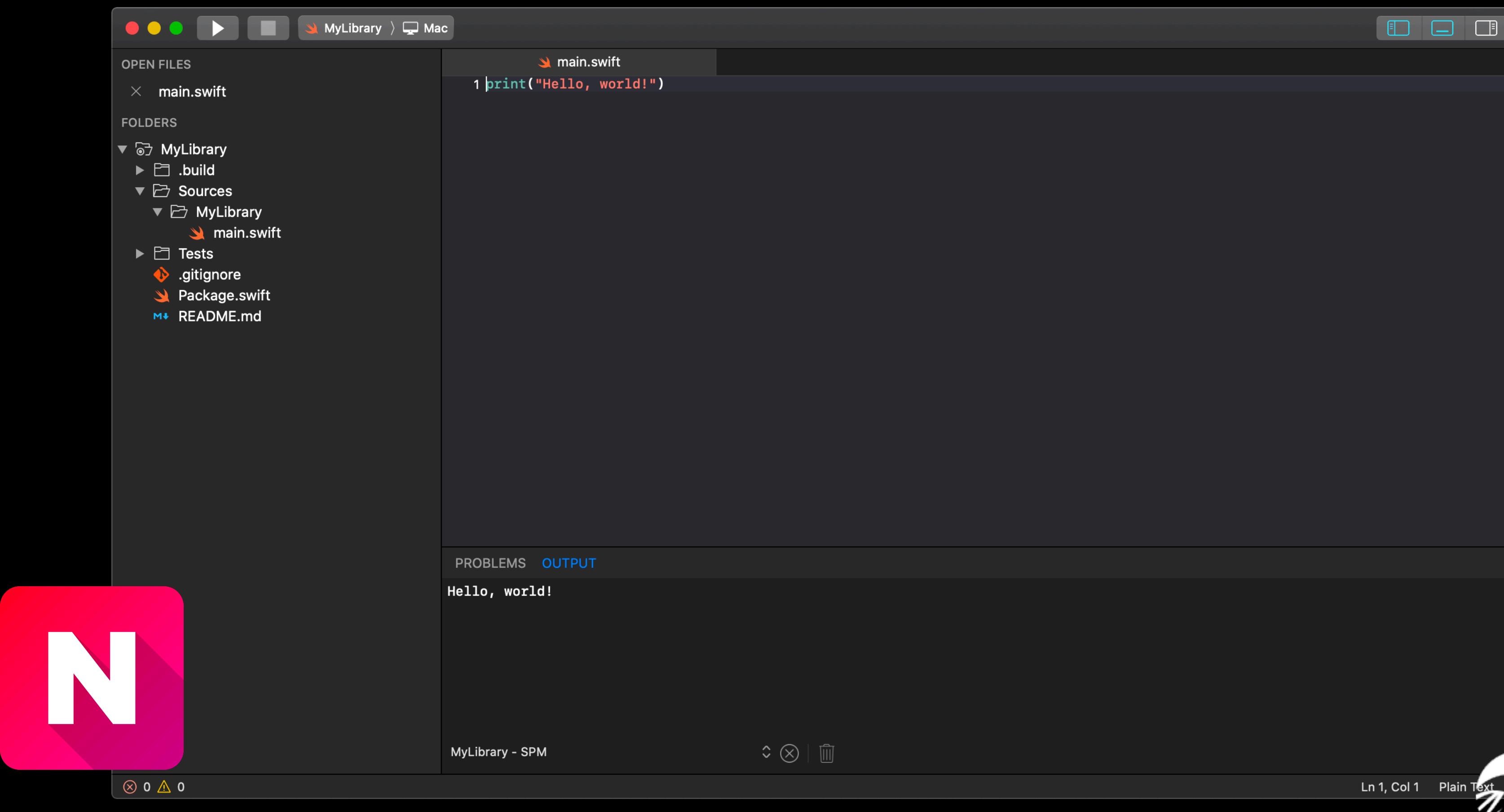
# Scade는 무엇을 제공해줄까요?

Crystal (High Performance Graphics Engine)



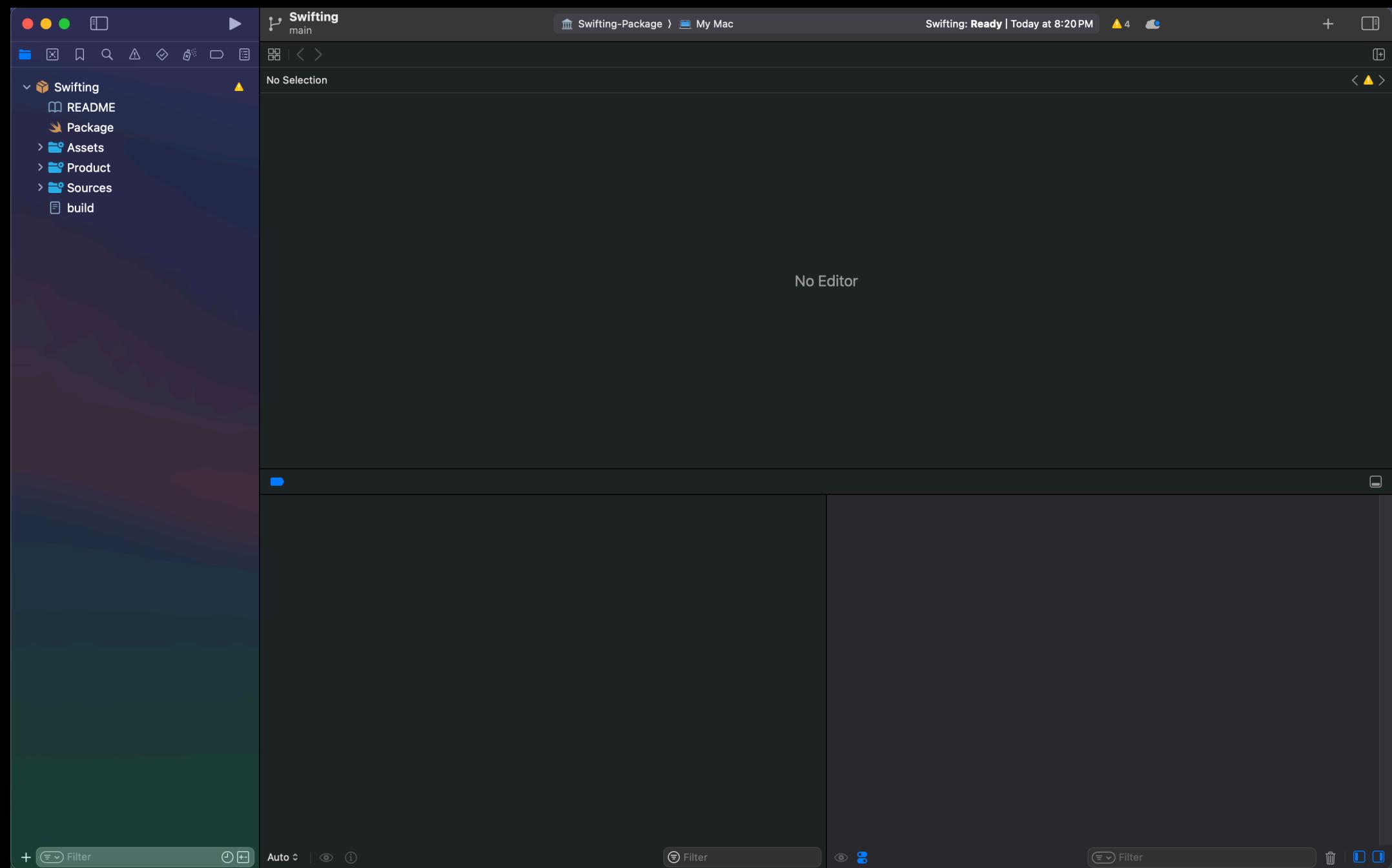
# Scade는 무엇을 제공해줄까요?

## Nimble (SCADE IDE)

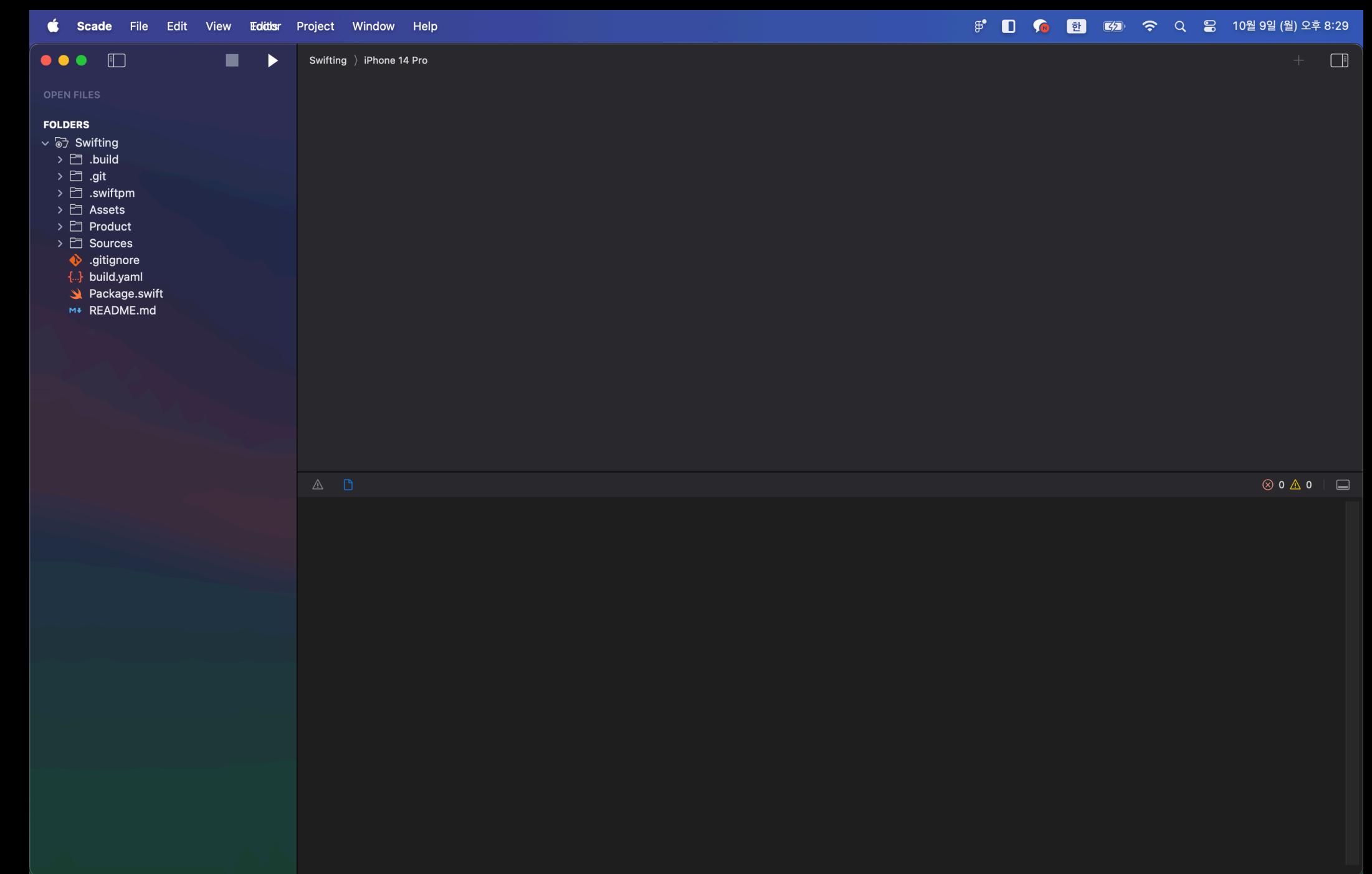


# Scade는 무엇을 제공해줄까요?

## Nimble (SCADE IDE)



Xcode



Nimble



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade는 무엇을 제공해줄까요?

## Nimble (SCADE IDE)

The screenshot shows the Nimble SCADE IDE interface. At the top, there is a code editor window titled '\*main.swift' containing the following Swift code:

```
1 import Foundation
2
3 let k: Int|
```

A code completion dropdown menu is open at the cursor position, showing several options for 'Int':

- S Int **Int**
- S Int8 **Int8**
- S Int16 **Int16**
- S Int32 **Int32**
- S Int64 **Int64**
- T Int **IntegerLiteralType**
- S IntText **IntText**
- T Int **intmax\_t**

Below the code editor, there is a message bar with a red warning icon and the text: "① Cannot find type 'SomeProtocol' in scope".

At the bottom, there is another message bar with a yellow warning icon and the text: "⚠ Initialization of immutable value 'str' was never used; consider replacing with assignment to '\_' or removing it". It also includes a "Replace 'let str' with '\_'" button and a "Fix" button.



# Scade는 무엇을 제공해줄까요?

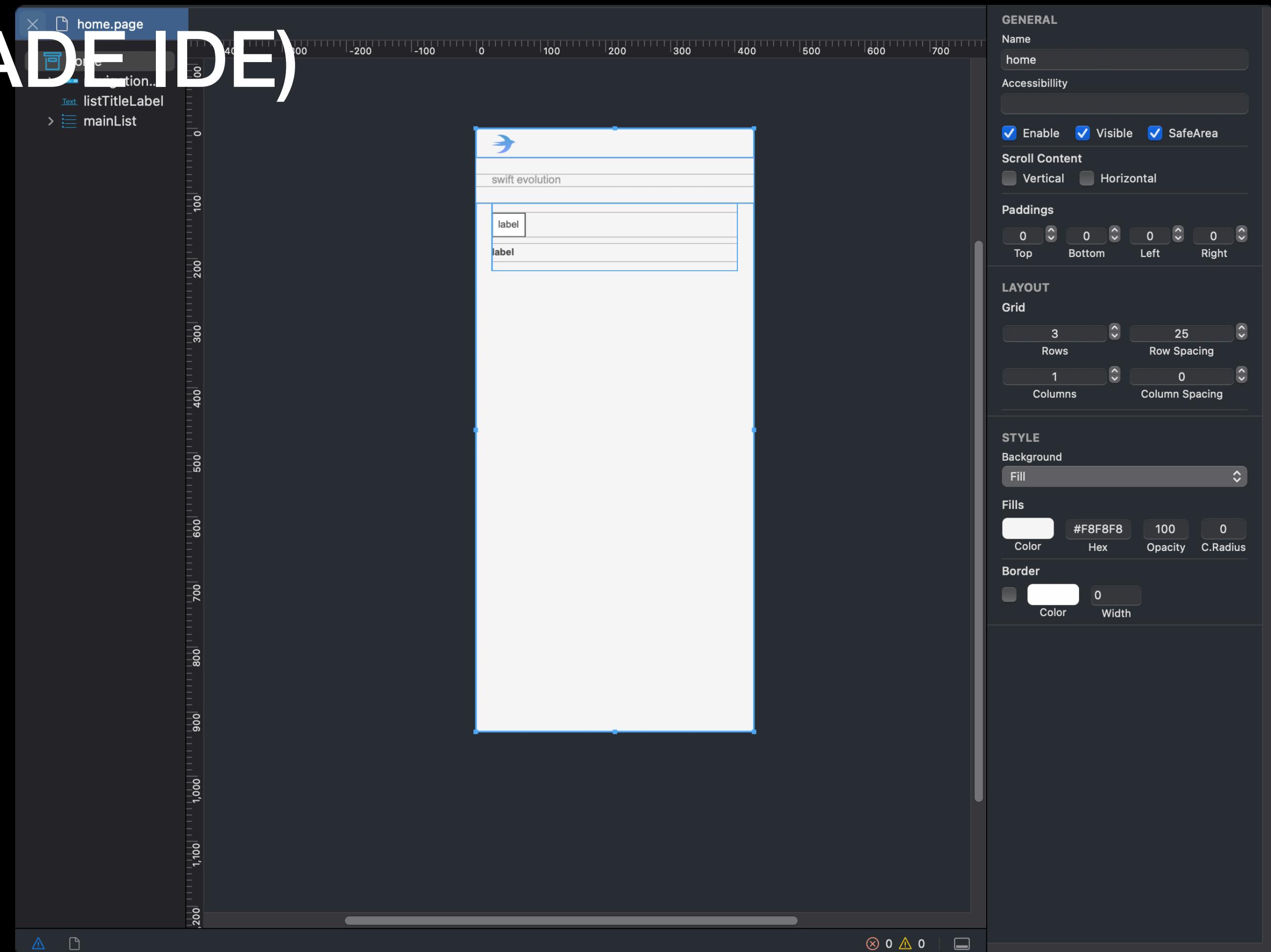
## Nimble (SCADE IDE)



**Let'Swift 2023**  
Deep Dive into the unknown

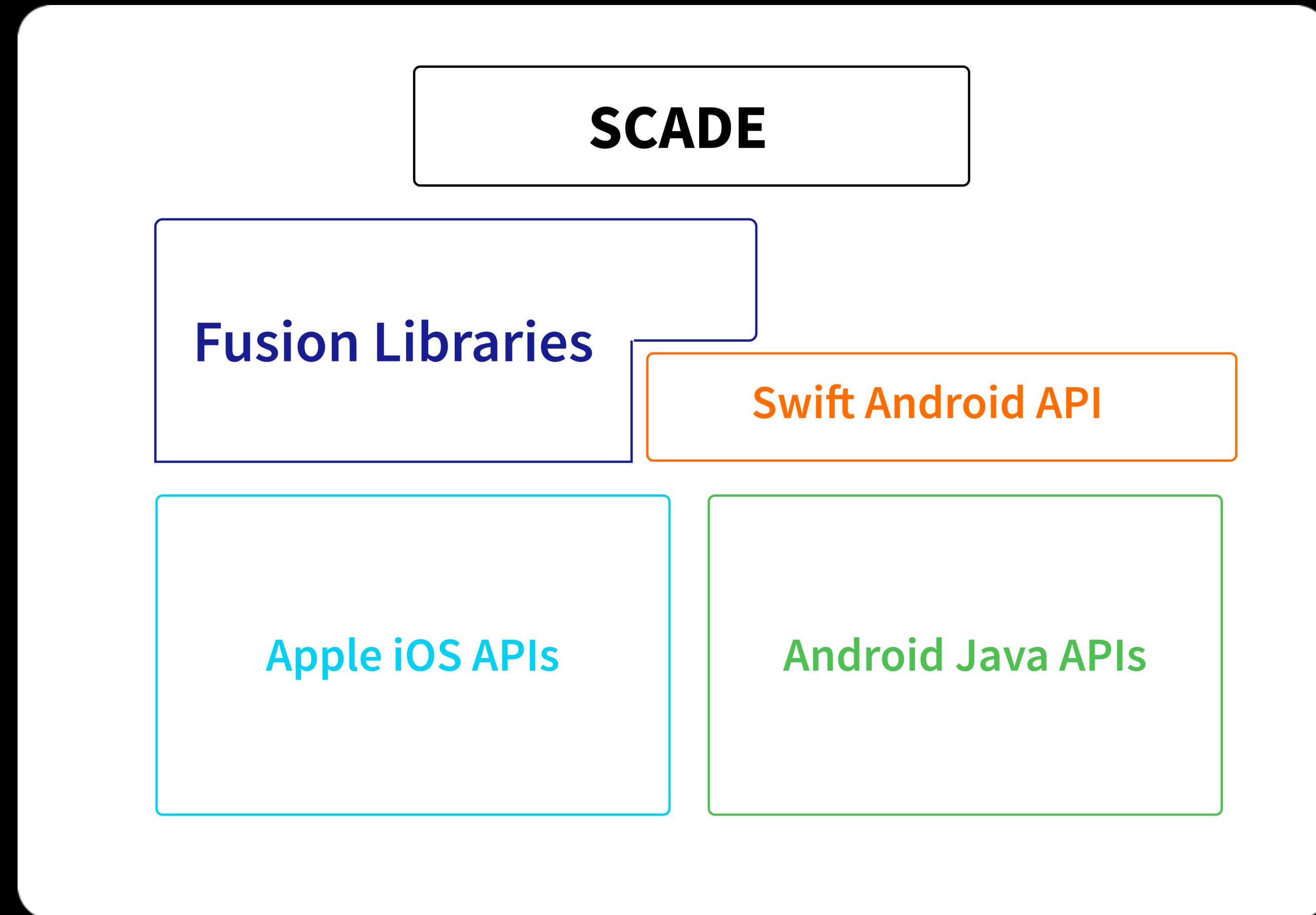
# Scade는 무엇을 제공해줄까요?

## Nimble (SCADE IDE)



# Scade는 무엇을 제공해줄까요?

Fusion



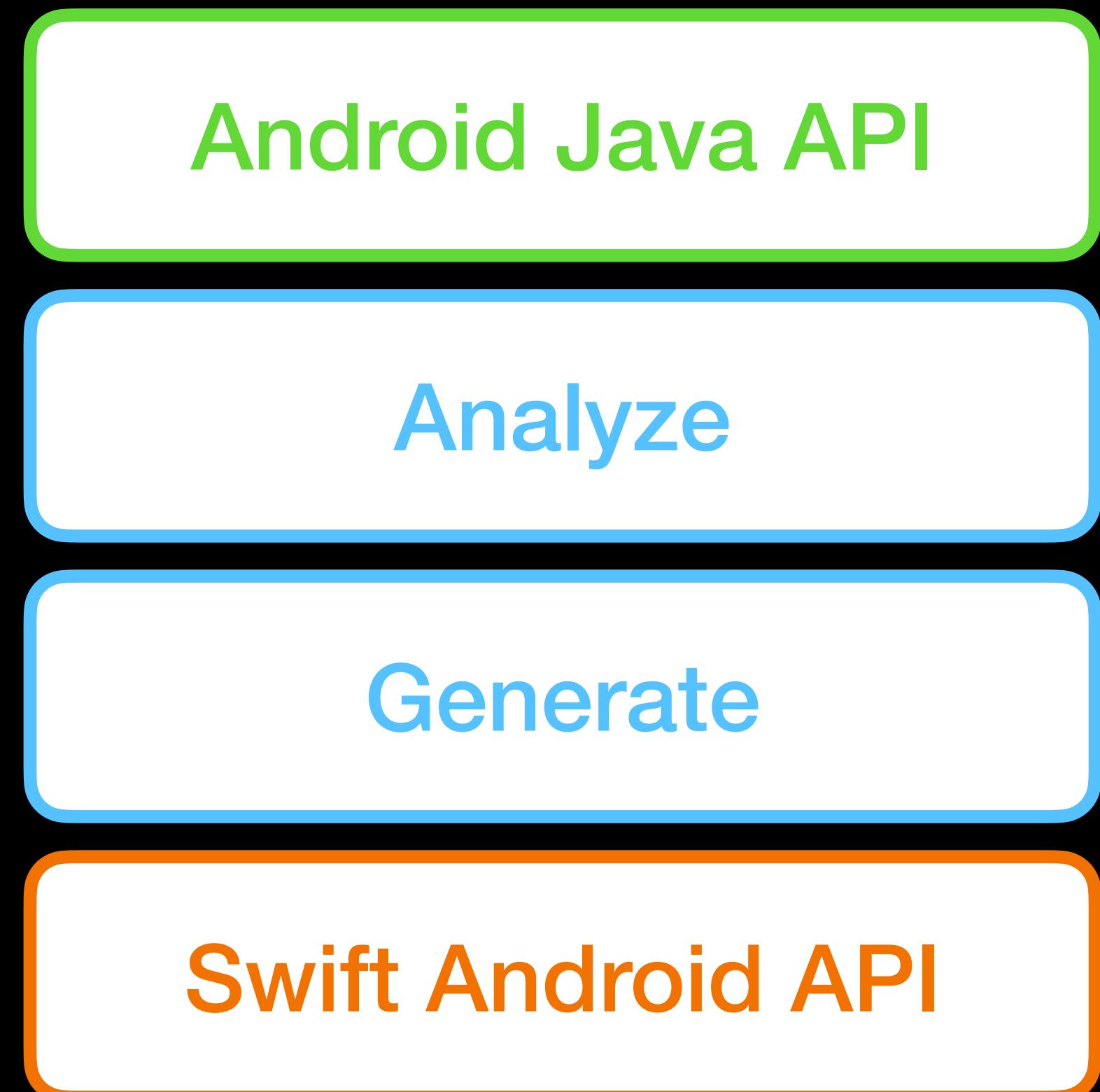
Fusion Architecture



Let'Swift 2023  
Deep Dive into the unknown

# Scade는 무엇을 제공해줄까요?

## Fusion



Fusion Generation Process



Let'Swift 2023  
Deep Dive into the unknown

# Scade는 무엇을 제공해줄까요?

## Fusion

S.No	Library Name	Repository	Description
1	FusionLocation	<a href="https://github.com/scade-platform/FusionLocation">https://github.com/scade-platform/FusionLocation</a>	Android & iOS Geolocation impl
2	FusionNFC	<a href="https://github.com/scade-platform/FusionNFC">https://github.com/scade-platform/FusionNFC</a>	NFC read/write impl for iOS and Android
3	FusionMedia	<a href="https://github.com/scade-platform/FusionMedia">https://github.com/scade-platform/FusionMedia</a>	Media player for the iOS and Android platform
4	FusionBluetooth	<a href="https://github.com/scade-platform/FusionBluetooth">https://github.com/scade-platform/FusionBluetooth</a>	Bluetooth impl for iOS and Android
5	FusionLocalAuth	<a href="https://github.com/scade-platform/FusionLocalAuth">https://github.com/scade-platform/FusionLocalAuth</a>	Local authentication using biometrics for iOS and Android
6	FusionQR	<a href="https://github.com/scade-platform/FusionCamera">https://github.com/scade-platform/FusionCamera</a>	QR code detection using Camera for iOS and Android



[ S C A D E ]



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 설정 방법



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 설정 방법



AndroidStudio



Xcode

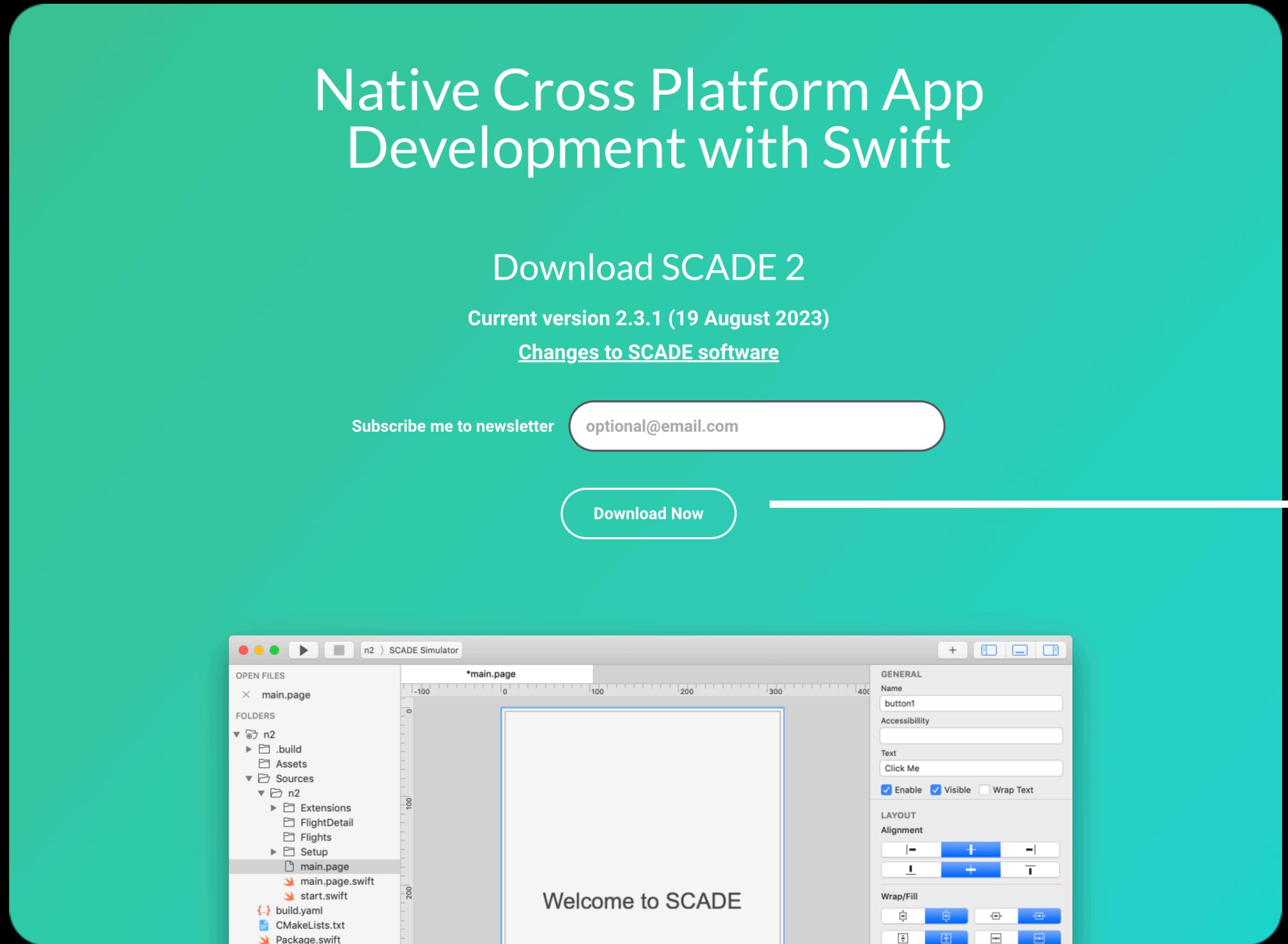


Scade



Let'Swift 2023  
Deep Dive into the unknown

# Scade 설정 방법



<https://www.scade.io/download/>

# Scade 설정 방법

iOS Setting



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 설정 방법

## Version 2.0.55 Beta - November 2022

- Xcode 14 support
- Copying content of SVG Images from clipboard and pasting them to the all widgets in the SCADE IDE
- Improved debug view
- Improved tabs
- Improved SCADE stability
- Increased iOS min version to 13
- Fixes and enhancements



# Scade 설정 방법



## Xcode 14.3.1

June 1, 2023

Hide Details ^

Xcode 14.3.1 includes everything you need to create amazing apps for all Apple platforms. It includes SDKs for iOS 16.4, iPadOS 16.4 tvOS 16.4, watchOS 9.4, and macOS Ventura 13.3.

[Xcode 14.3.1.xip](#) ⓘ

7.01 GB

<https://developer.apple.com/download/all/?q=xcode%2014>



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 설정 방법

## Terminal

```
> sudo xcode-select -s /Applications/Xcode_14.3.1.app/Contents/Developer
```



**Let'Swift 2023**  
Deep Dive into the unknown

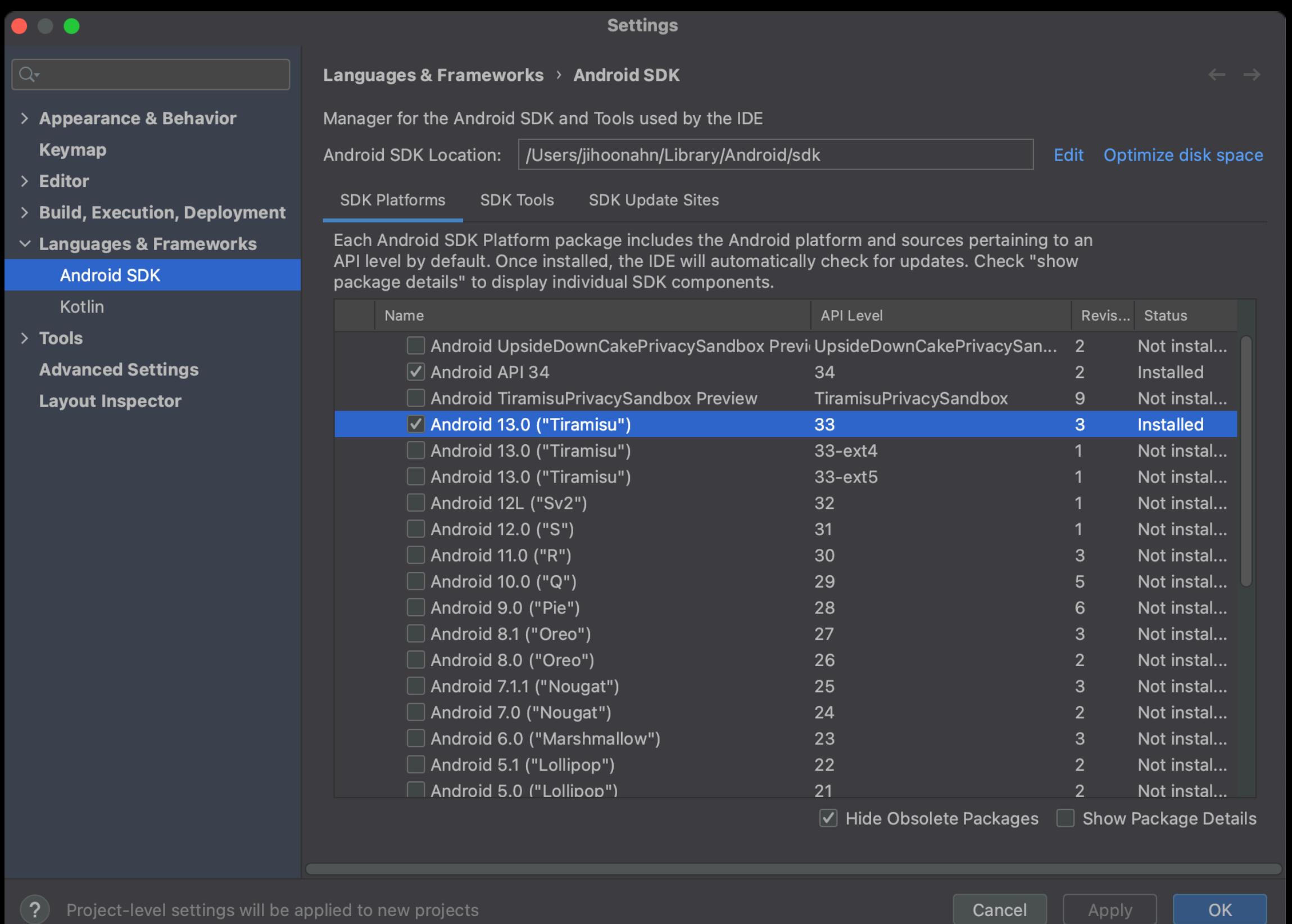
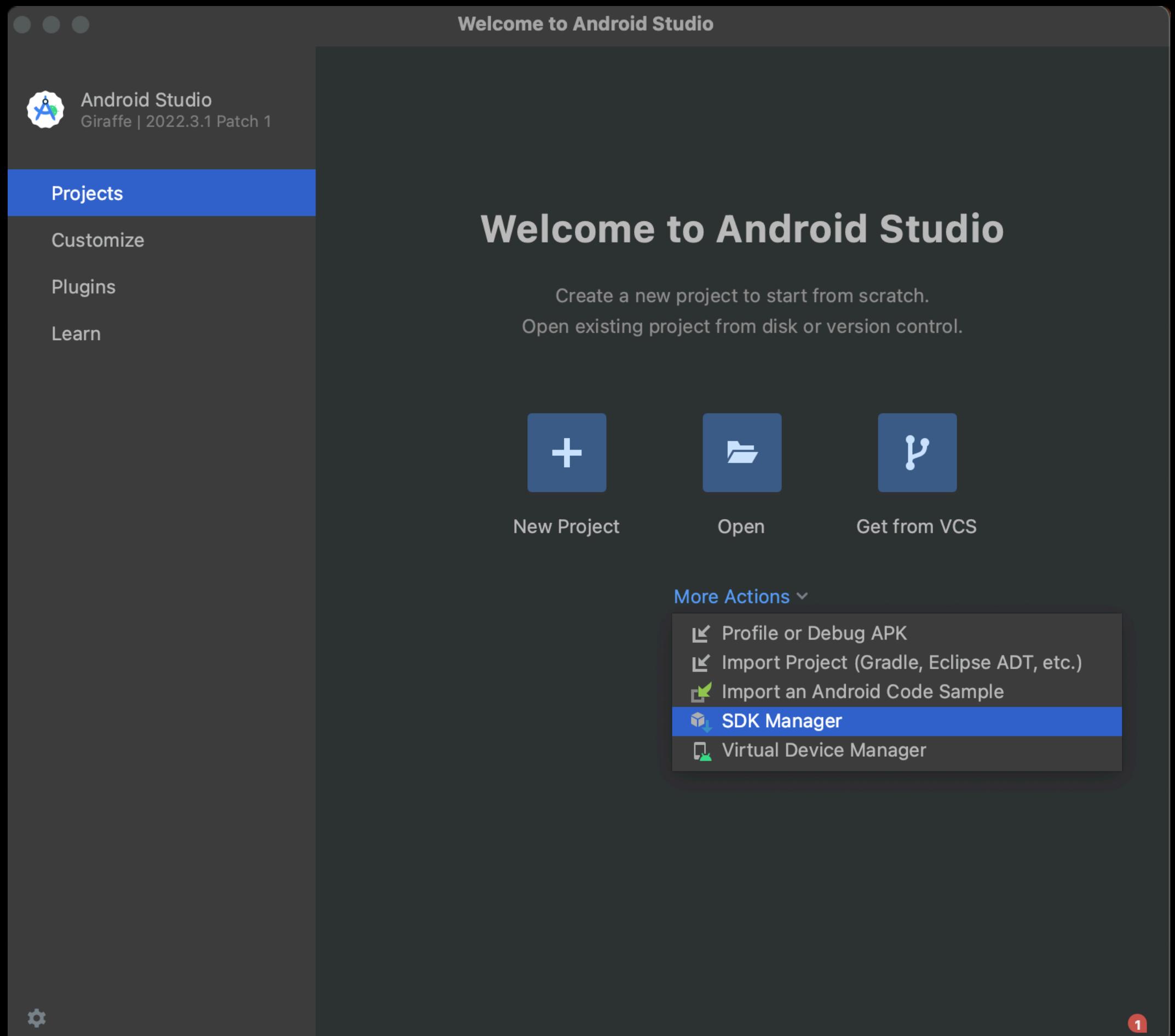
# Scade 설정 방법

Android Setting



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 설정 방법



# Scade 설정 방법

Settings  
Languages & Frameworks > Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: /Users/jihoonahn/Library/Android/sdk [Edit](#) [Optimize disk space](#)

SDK Platforms [SDK Tools](#) [SDK Update Sites](#)

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

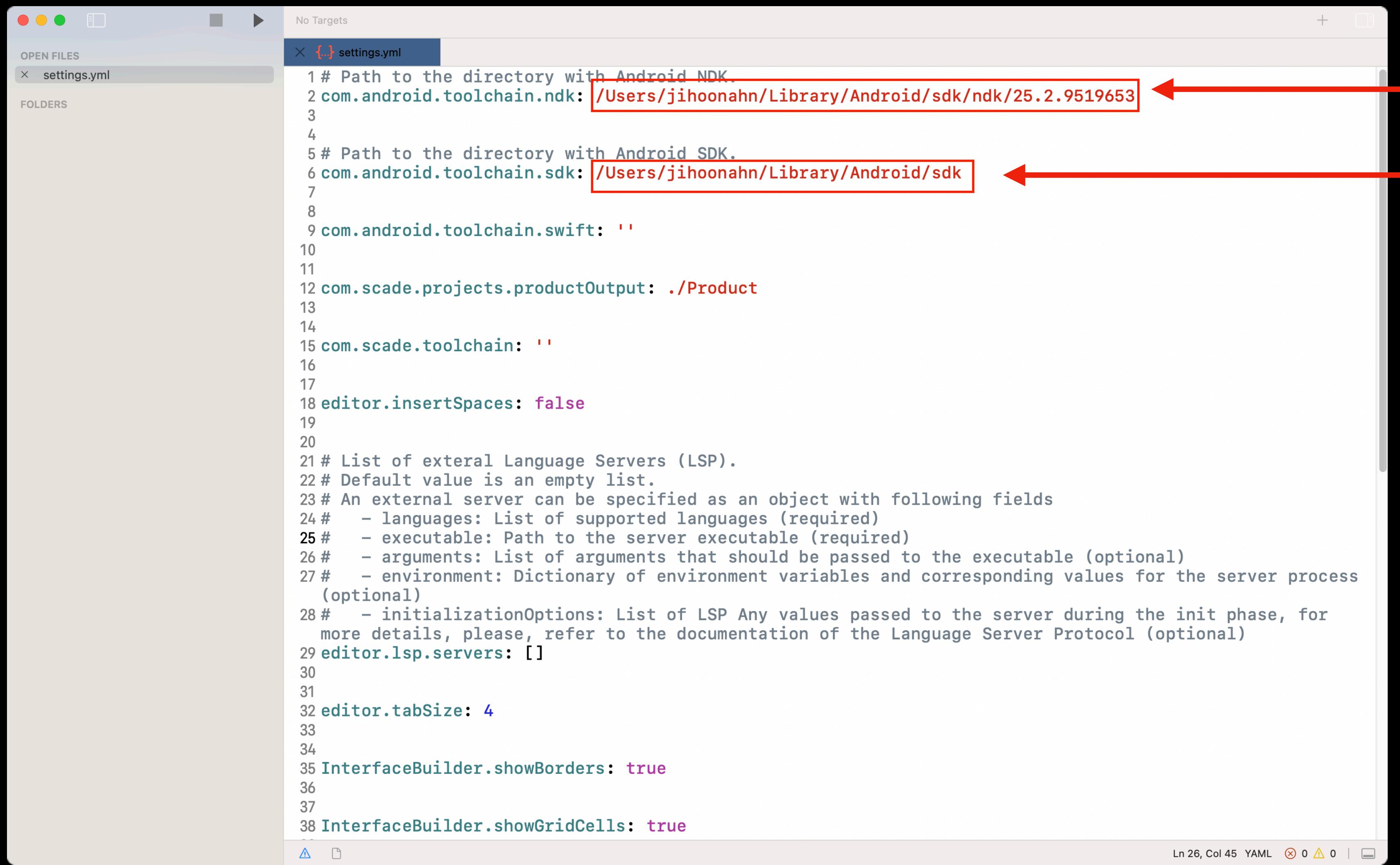
Name	Version	Status
<input checked="" type="checkbox"/> Android SDK Build-Tools 34		Installed
<input type="checkbox"/> NDK (Side by side)		Update Available: 26.0.10792818
<input checked="" type="checkbox"/> Android SDK Command-line Tools (latest)		Installed
<input checked="" type="checkbox"/> CMake		Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit Emulator	2.0	Not installed
<input checked="" type="checkbox"/> Android Emulator	32.1.15	Installed
<input checked="" type="checkbox"/> Android SDK Platform-Tools	34.0.4	Installed
<input type="checkbox"/> Google Play APK Expansion library	1	Not installed
<input type="checkbox"/> Google Play Instant Development SDK	1.9.0	Not installed
<input checked="" type="checkbox"/> Google Play Licensing Library	1	Installed
<input checked="" type="checkbox"/> Google Play services	49	Installed
<input type="checkbox"/> Google Web Driver	2	Not installed
<input type="checkbox"/> Layout Inspector image server for API 29-30	6	Not installed
<input type="checkbox"/> Layout Inspector image server for API 31-34	3	Not installed
<input type="checkbox"/> Layout Inspector image server for API S	3	Not installed

Hide Obsolete Packages  Show Package Details

? Project-level settings will be applied to new projects [Cancel](#) [Apply](#) [OK](#)



# Scade 설정 방법



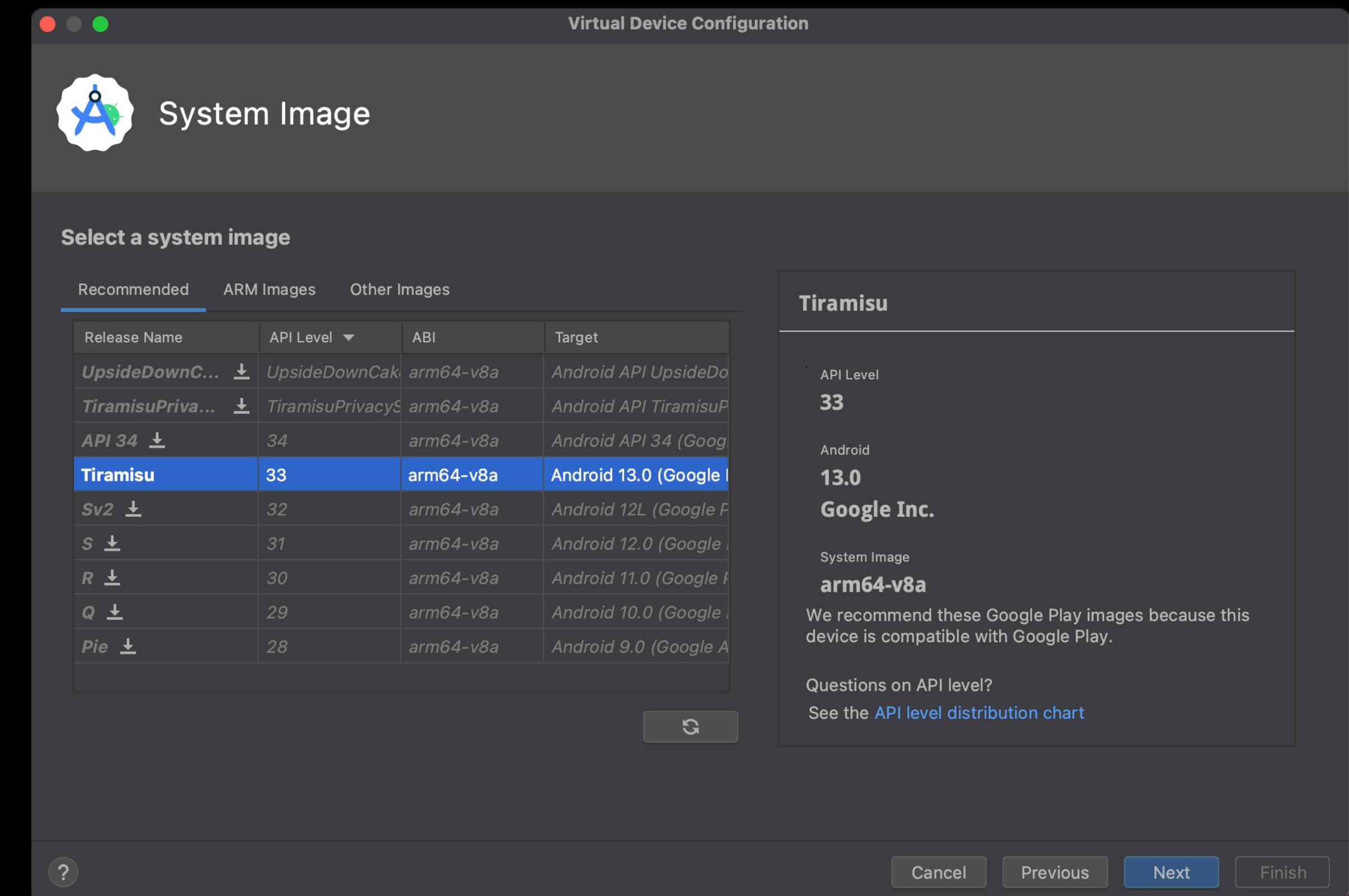
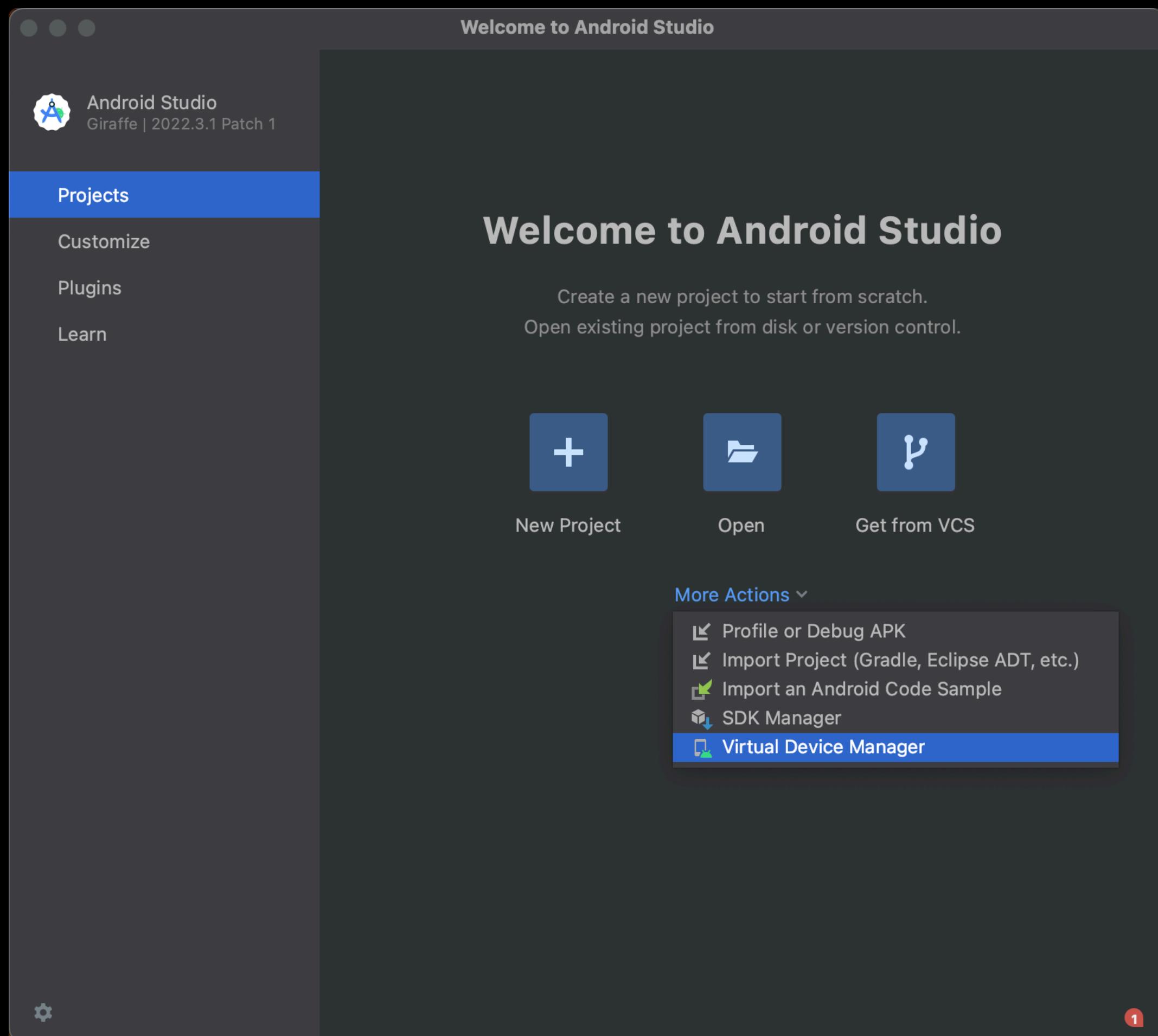
```
No Targets
× {..} settings.yml
OPEN FILES
× settings.yml
FOLDERS

1 # Path to the directory with Android NDK.
2 com.android.toolchain.ndk: /Users/jihoonahn/Library/Android/sdk/ndk/25.2.9519653 ← NDK 위치
3
4
5 # Path to the directory with Android SDK.
6 com.android.toolchain.sdk: /Users/jihoonahn/Library/Android/sdk ← SDK 위치
7
8
9 com.android.toolchain.swift: ''
10
11
12 com.scade.projects.productOutput: ./Product
13
14
15 com.scade.toolchain: ''
16
17
18 editor.insertSpaces: false
19
20
21 # List of external Language Servers (LSP).
22 # Default value is an empty list.
23 # An external server can be specified as an object with following fields
24 #   - languages: List of supported languages (required)
25 #   - executable: Path to the server executable (required)
26 #   - arguments: List of arguments that should be passed to the executable (optional)
27 #   - environment: Dictionary of environment variables and corresponding values for the server process
#     (optional)
28 #   - initializationOptions: List of LSP Any values passed to the server during the init phase, for
#     more details, please, refer to the documentation of the Language Server Protocol (optional)
29 editor.lsp.servers: []
30
31
32 editor.tabSize: 4
33
34
35 InterfaceBuilder.showBorders: true
36
37
38 InterfaceBuilder.showGridCells: true
```

NDK 위치  
SDK 위치



# Scade 설정 방법



# Scade Setting



Scade 사용 준비 끝!



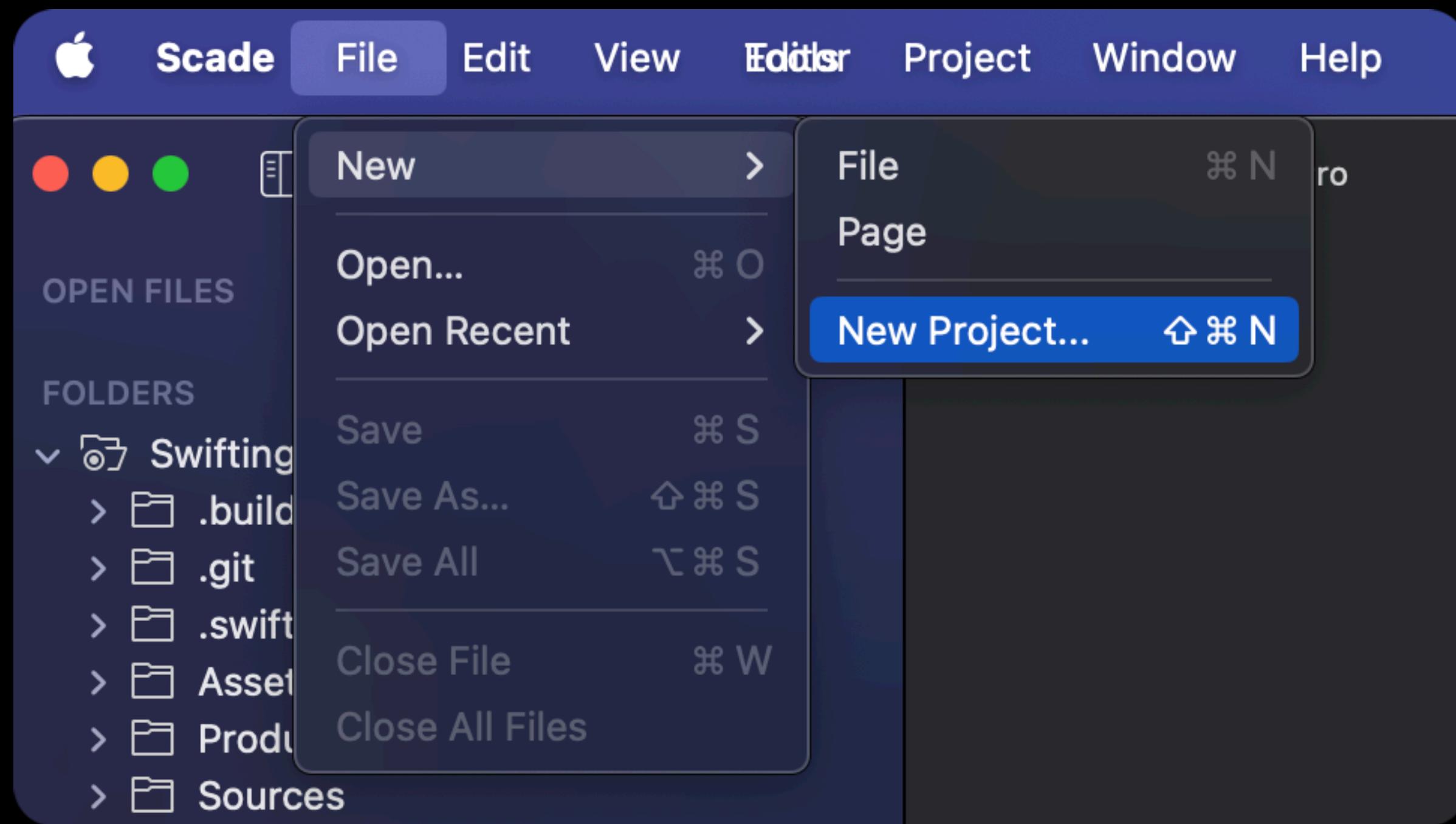
Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?



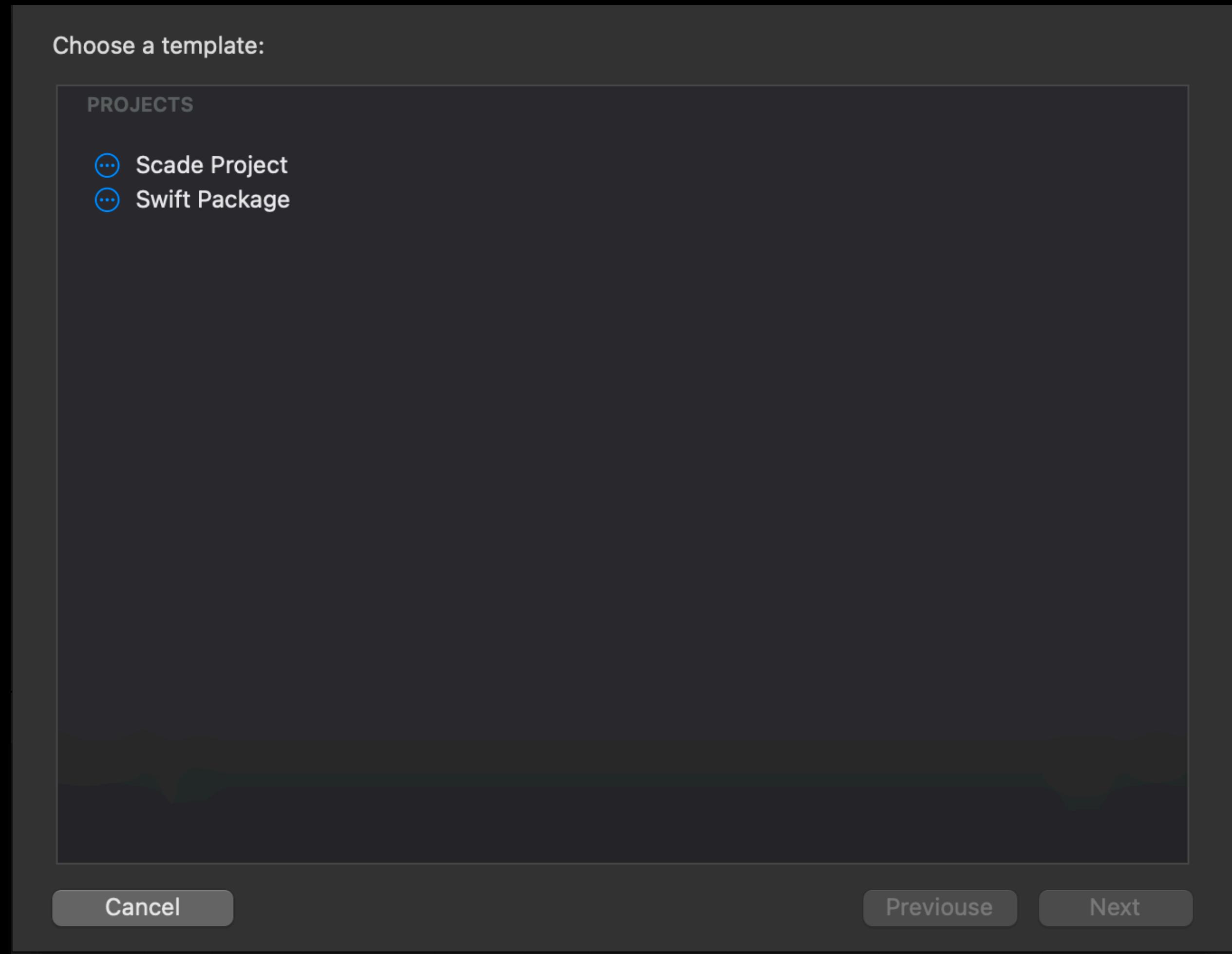
**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

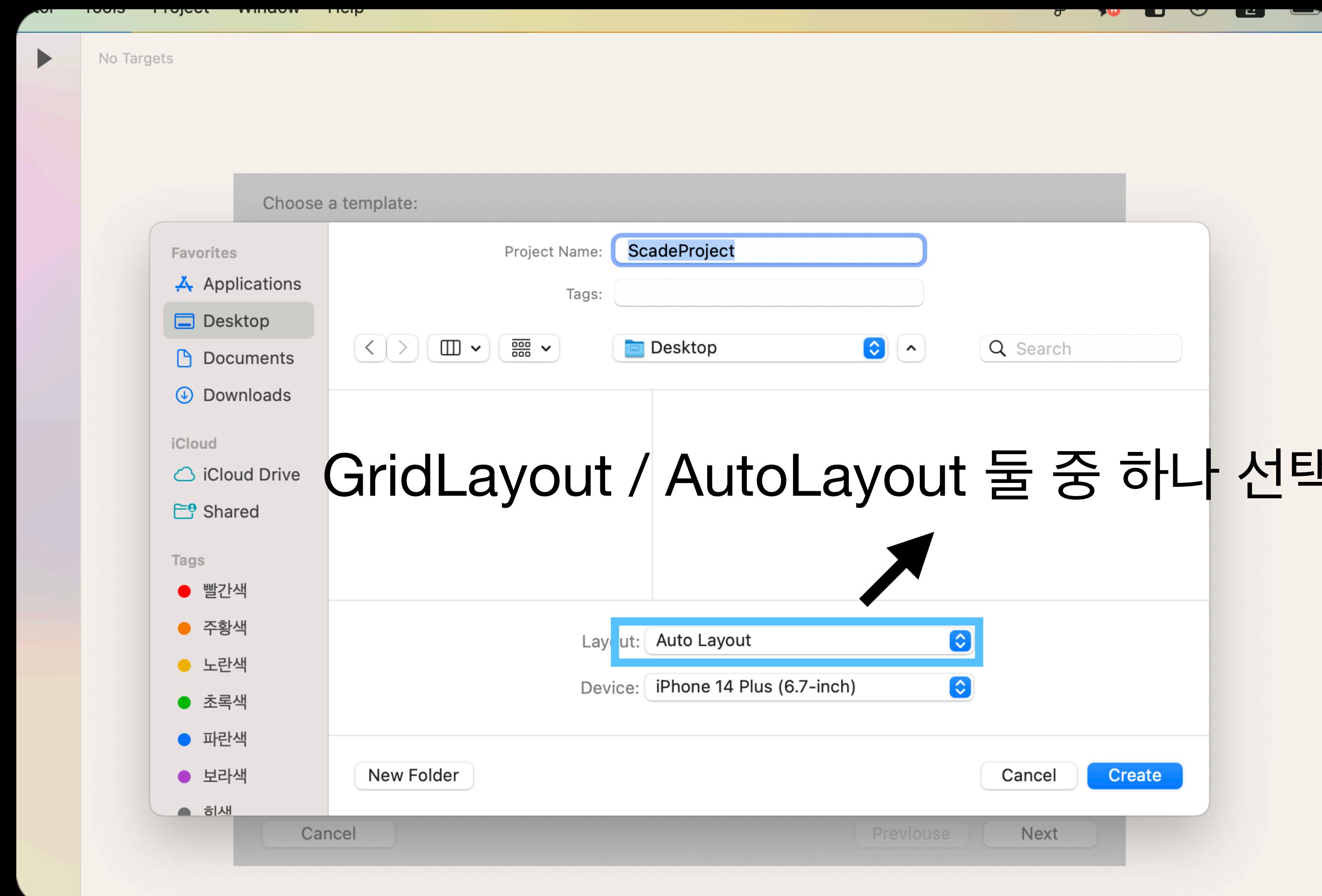


File > New > New Project / Shift + Command + N

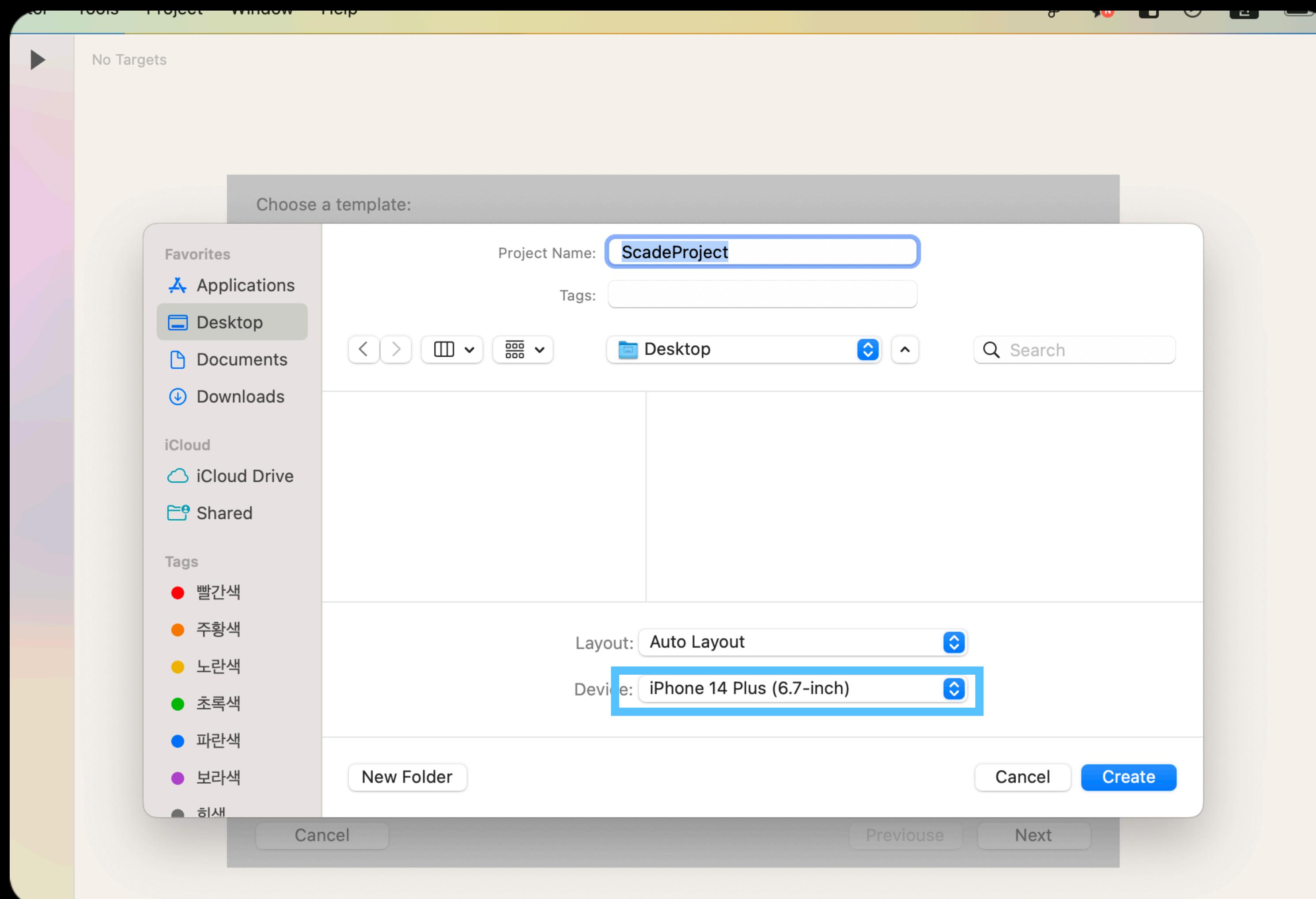
# Scade 시작해볼까요?



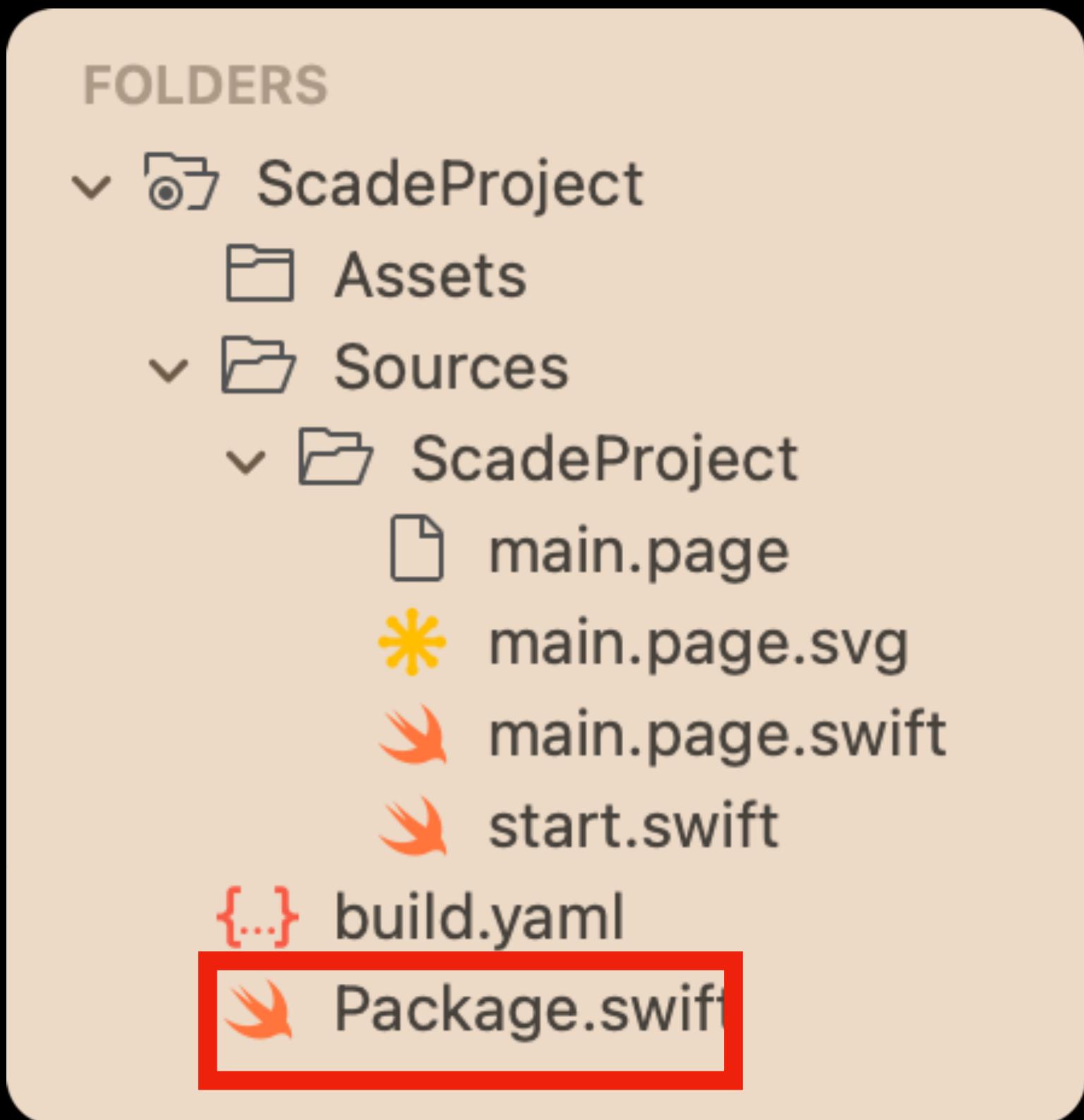
# Scade 시작해볼까요?



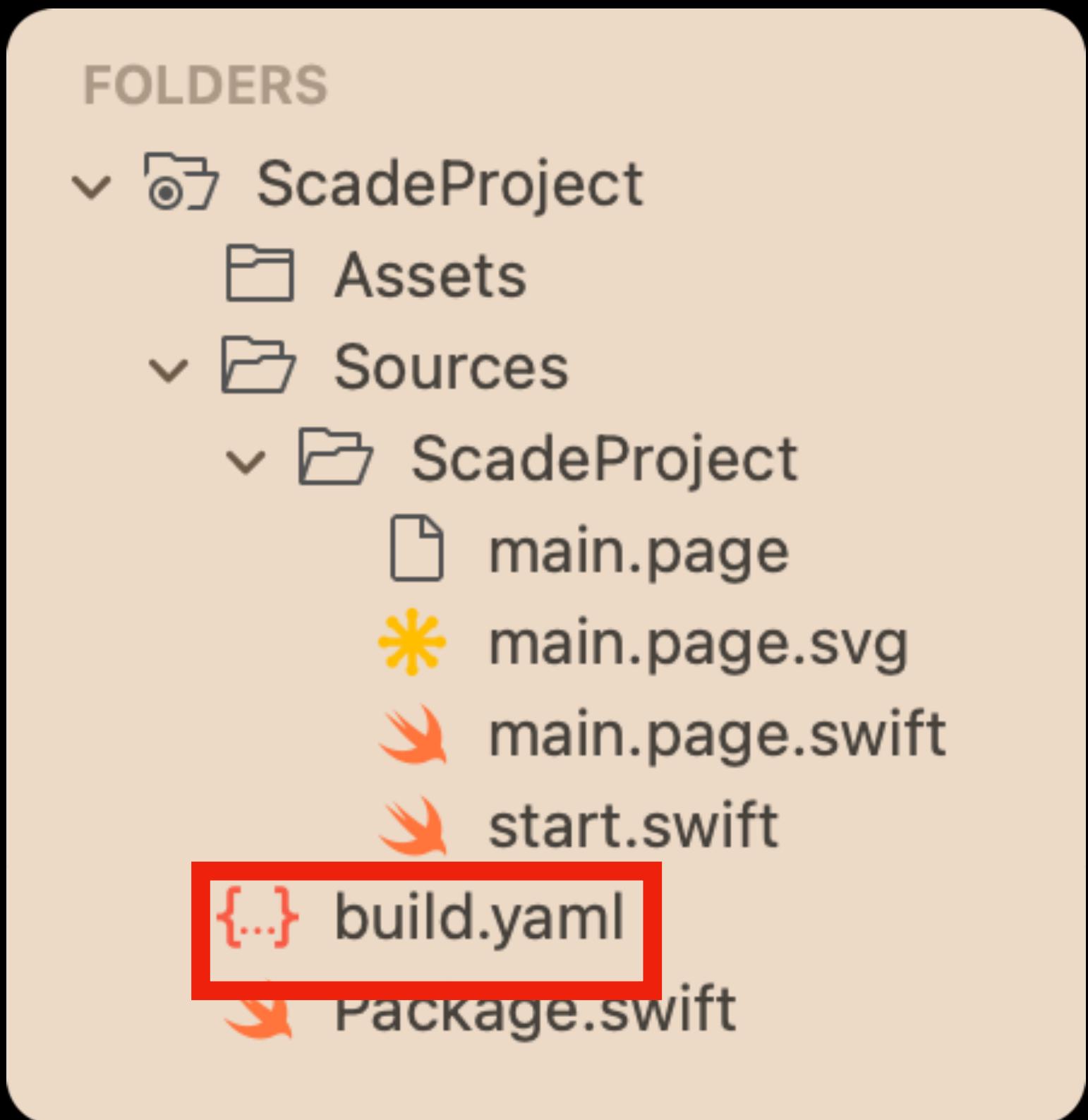
# Scade 시작해볼까요?



# Scade 시작해볼까요?



# Scade 시작해볼까요?



# Scade 시작해볼까요?

```
import ScadeKit

class ScadeProject: SCDApplication {

    let window = SCDLatticeWindow()
    let mainAdapter = MainPageAdapter()

    override func onFinishLaunching() {
        mainAdapter.load("main.page")
        mainAdapter.show(view: window)
    }
}
```

start.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

```
import ScadeKit

class ScadeProject: SCDApplication {

    let window = SCDLatticeWindow()
    let mainAdapter = MainPageAdapter()

    override func onFinishLaunching() {
        mainAdapter.load("main.page")
        mainAdapter.show(view: window)
    }
}
```



AppDelegate

start.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

```
import ScadeKit

class ScadeProject: SCDAplication {

    let window = SCDLatticeWindow()
    let mainAdapter = MainPageAdapter()

    override func onFinishLaunching() {
        mainAdapter.load("main.page")
        mainAdapter.show(view: window)
    }
}
```

start.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

```
import ScadeKit

class ScadeProject: SCDAplication {

    let window = SCDLatticeWindow()
    let mainAdapter = MainPageAdapter()

    override func onFinishLaunching() {
        mainAdapter.load("main_page")
        mainAdapter.show(view: window)
    }
}
```

start.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

```
import ScadeKit

class MainPageAdapter: SCDLatticePageAdapter {

    // page adapter initialization
    override func load(_ path: String) {
        super.load(path)
    }
}
```

main.page.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

```
import ScadeKit

class MainPageAdapter: SCDLatticePageAdapter {

    // page adapter initialization
    override func load(_ path: String) {
        super.load(path)
    }
}
```



ViewController

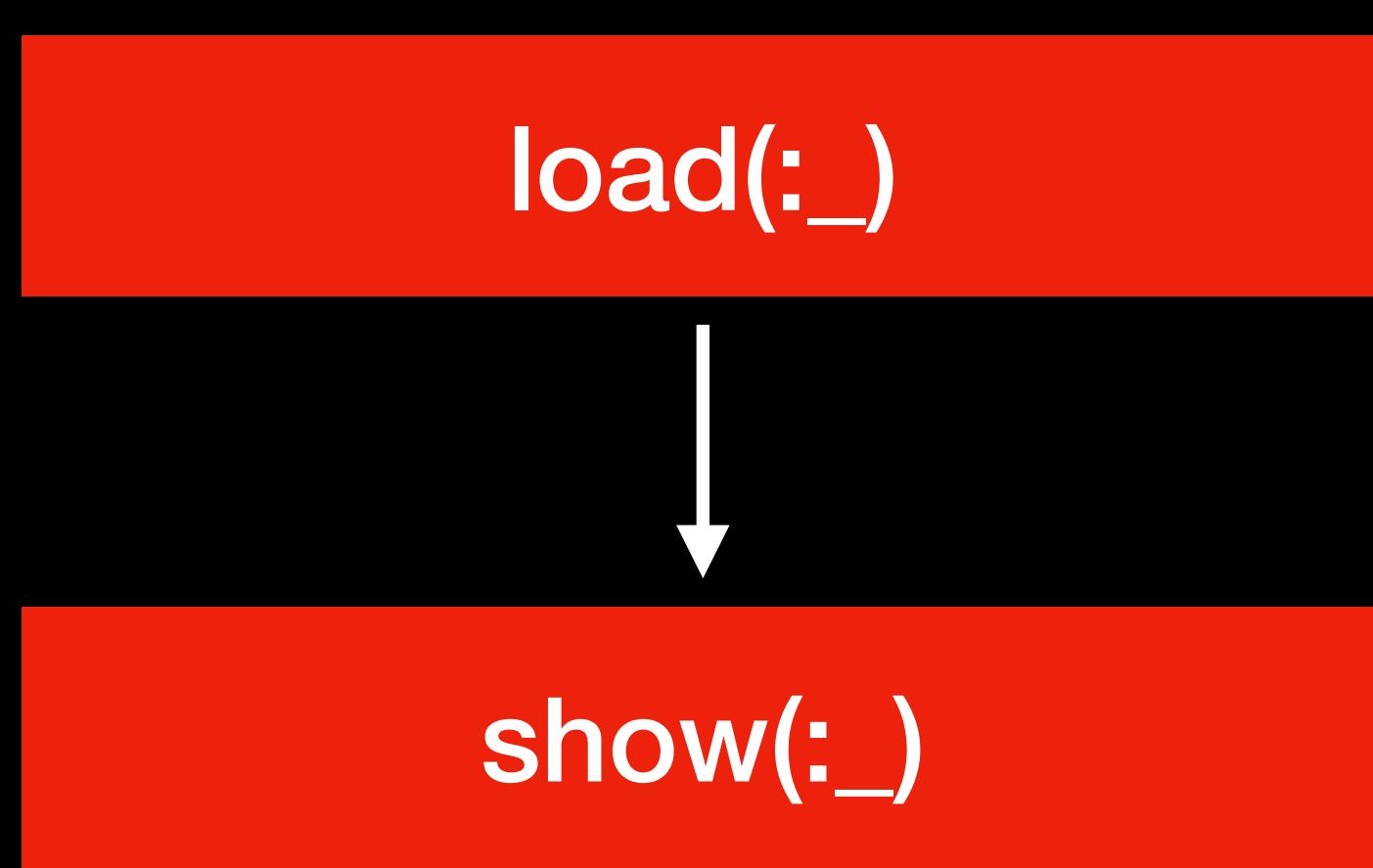
main.page.swift



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?

## Page LifeCycle



# Scade 시작해볼까요?

load(:\_)

- 데이터 구조 설정 및 초기화
- UI 객체에 대한 참조
- 설정 이벤트
- 데이터 로딩 로직을 구현하고 싶을때 사용



# Scade 시작해볼까요?

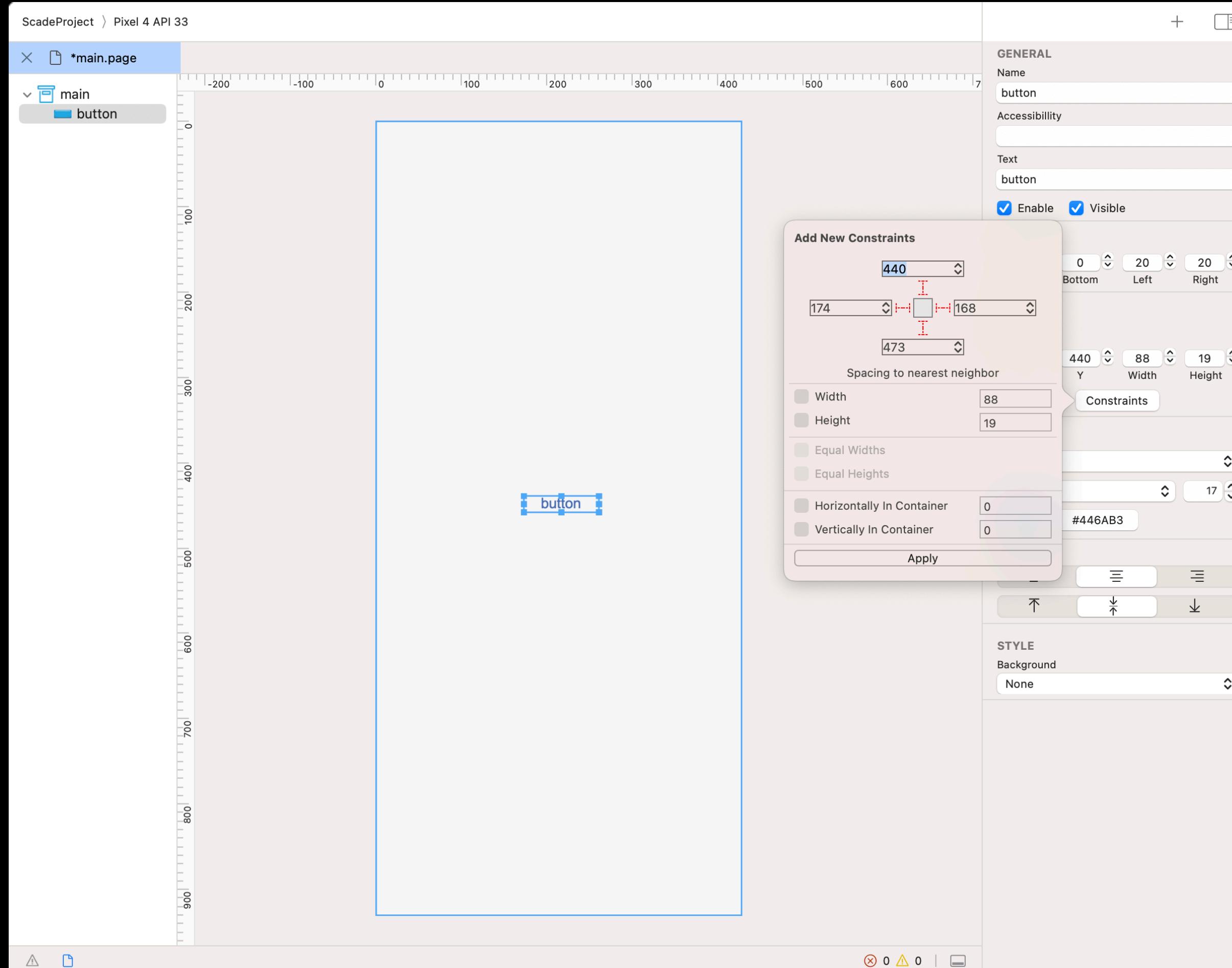
show(:\_)

- 화면이 나타날 때마다 작업을 수행

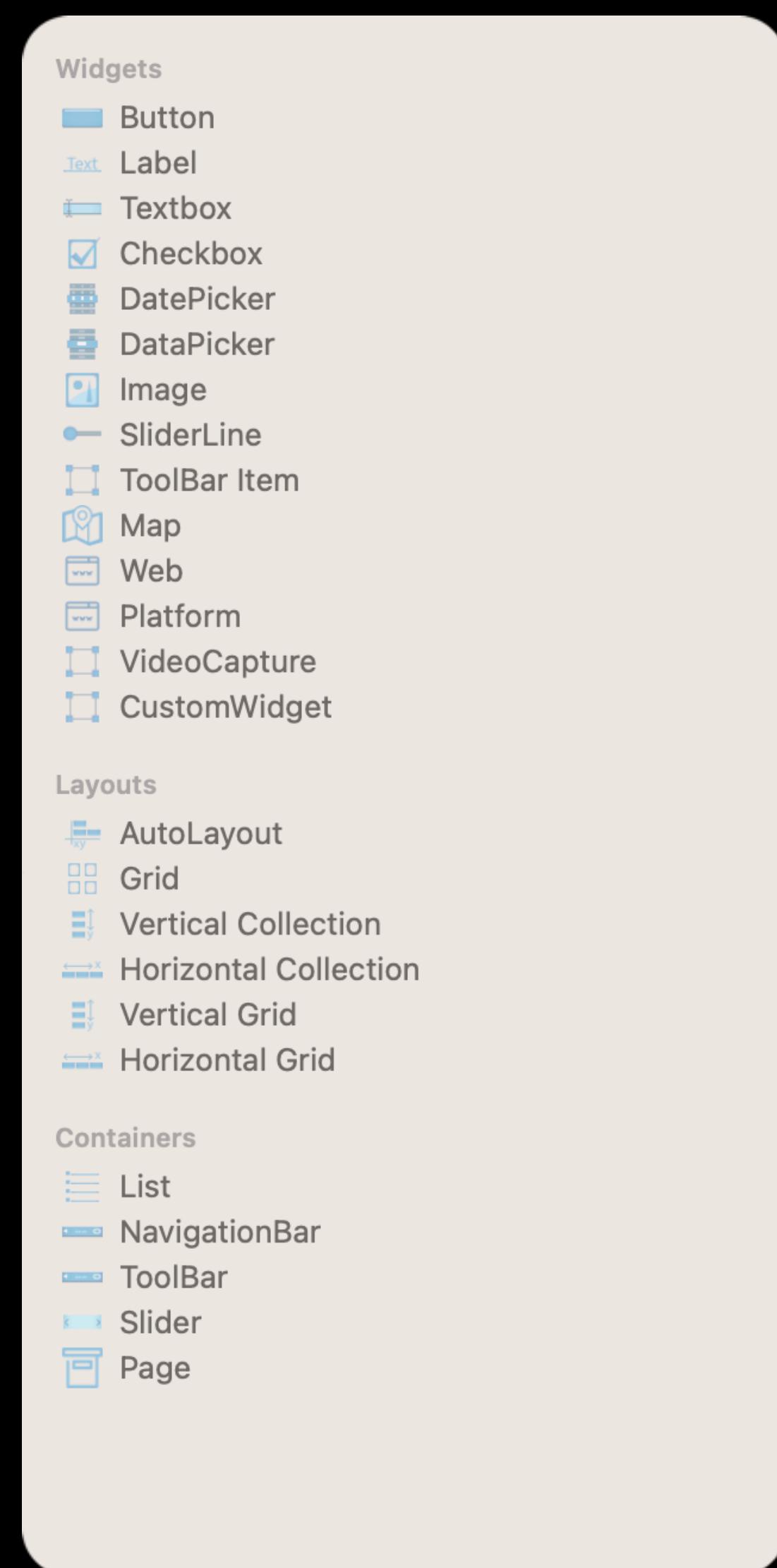


**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

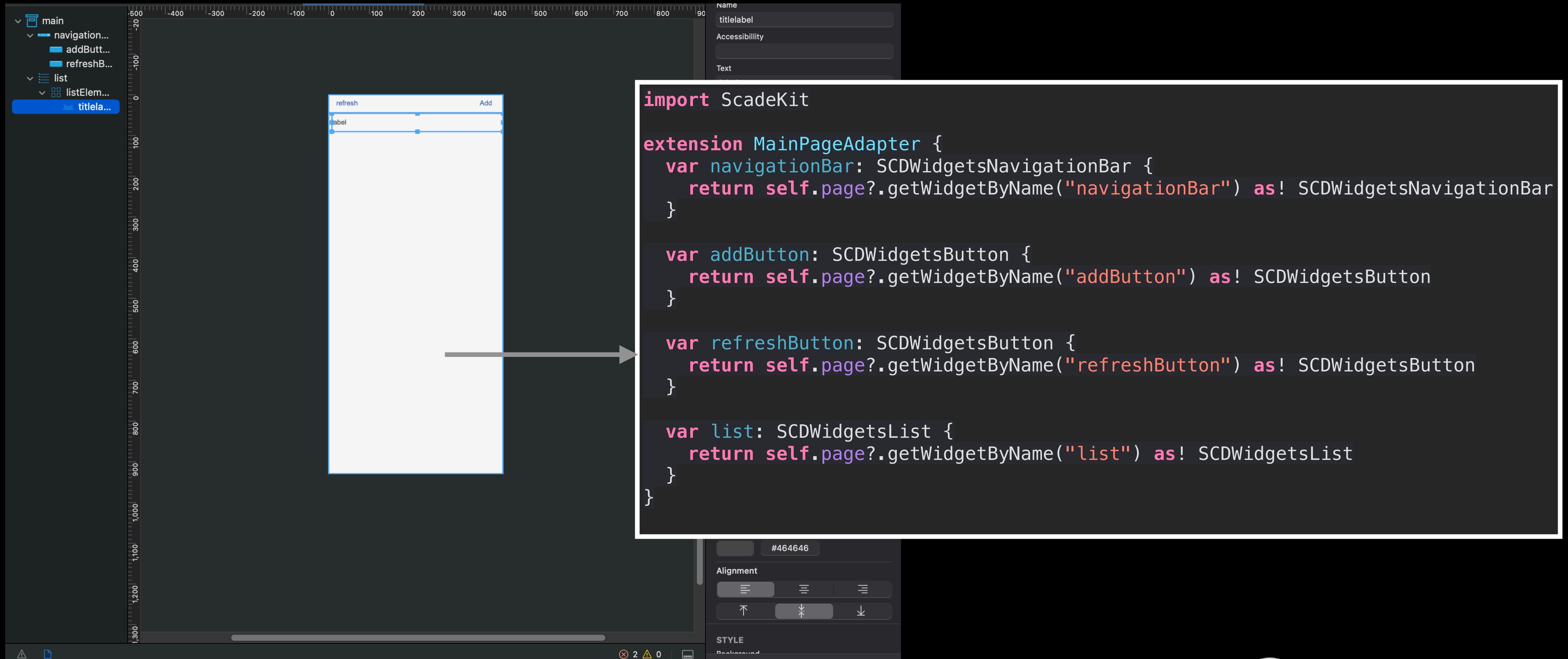


main.page



Let'Swift 2023  
Deep Dive into the unknown

# Scade 시작해볼까요?



main.page

# Scade 시작해볼까요?

```
import ScadeKit

final class MainPageAdapter: SCDLatticePageAdapter {

    var info: [String] = []
    // page adapter initialization
    override func load(_ path: String) {
        super.load(path)
        bind()
    }

    private func bind() {
        list.elementProvider = SCDWidgetsElementProvider { (item: String, template) in
            (template.getWidgetByName("titlelabel") as? SCDWidgetsLabel)?.text = item
        }
        closeButton.onClick.append(
            SCDWidgetsEventHandler { _ in
                self.list.items = SQLServices.shared.fetchTodo()
            }
        )
        addButton.onClick.append(
            SCDWidgetsEventHandler { _ in
                self.navigation?.go(page: "addtodo.page", transition: .fromRight)
            }
        )
        list.items = SQLServices.shared.fetchTodo()
    }
}
```



# Scade 시작해볼까요?

## 간단한 예시 코드



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



Swift



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

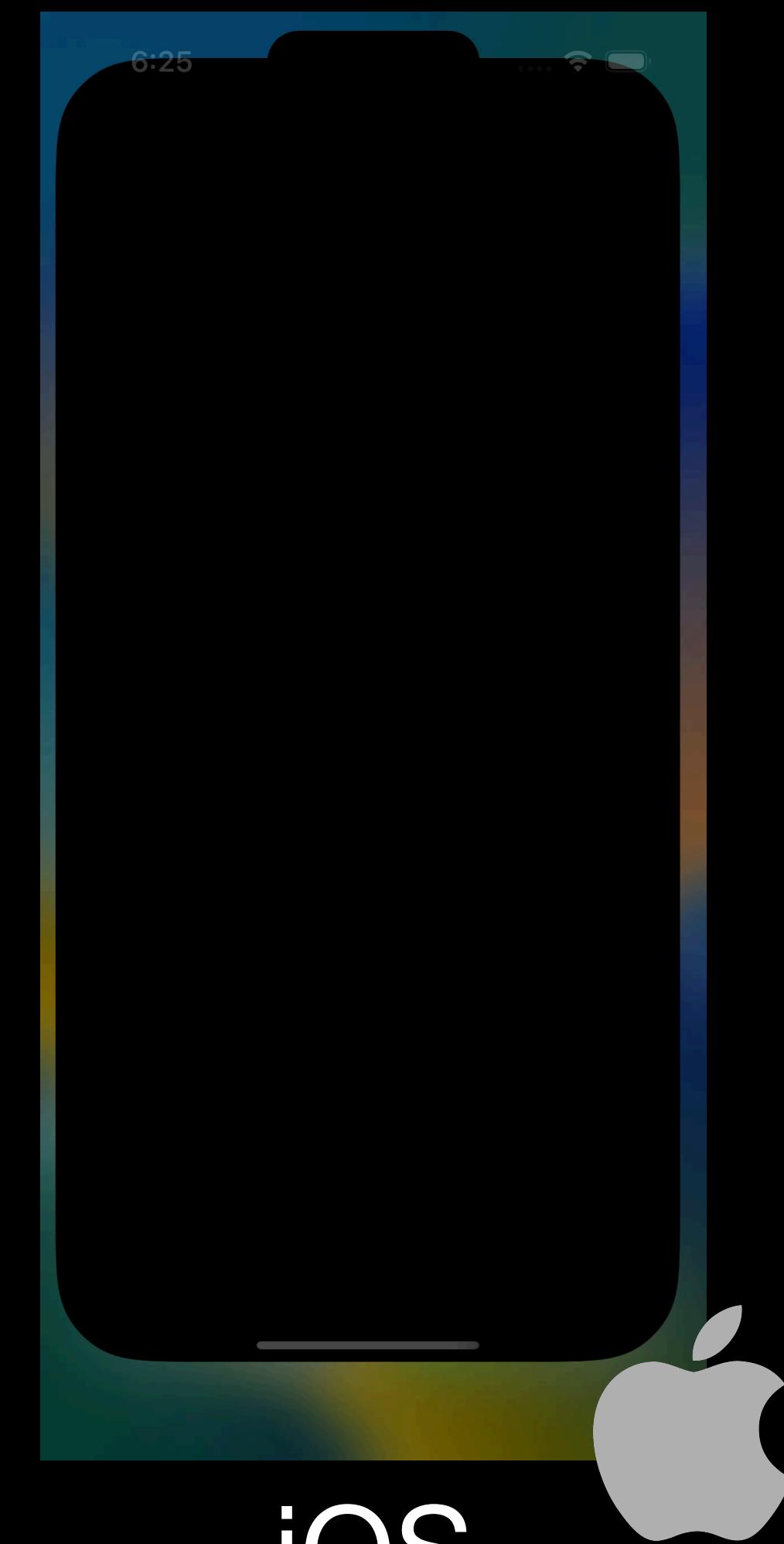
The screenshot shows the Swift Evolution website. The left sidebar has a navigation menu with links like About Swift, Blog, Getting Started, Install, Download, Platform Support, Documentation, Overview, Swift Compiler, Standard Library, Package Manager, Core Libraries, REPL, Debugger & Playgrounds, Swift on Server, Swift Evolution (which is currently selected), Source Code, Continuous Integration, and Source Compatibility. The main content area has a title "Swift Evolution". It contains text about the evolution process, mentioning a public forum and release goals. Below this is a section titled "409 proposals" with two entries:

- SE-0409 Access-level modifiers on import declarations**  
Author: Alexis Laferrière  
Review Manager: Frederick Kellison-Linn  
Status: Active Review  
Scheduled: September 14–26
- SE-0220 count(where:)**  
Author: Soroush Khanlou  
Review Manager: Ben Cohen  
Implementation: swift#16099, swift#22289  
Status: Active Review  
Scheduled: July 18–28



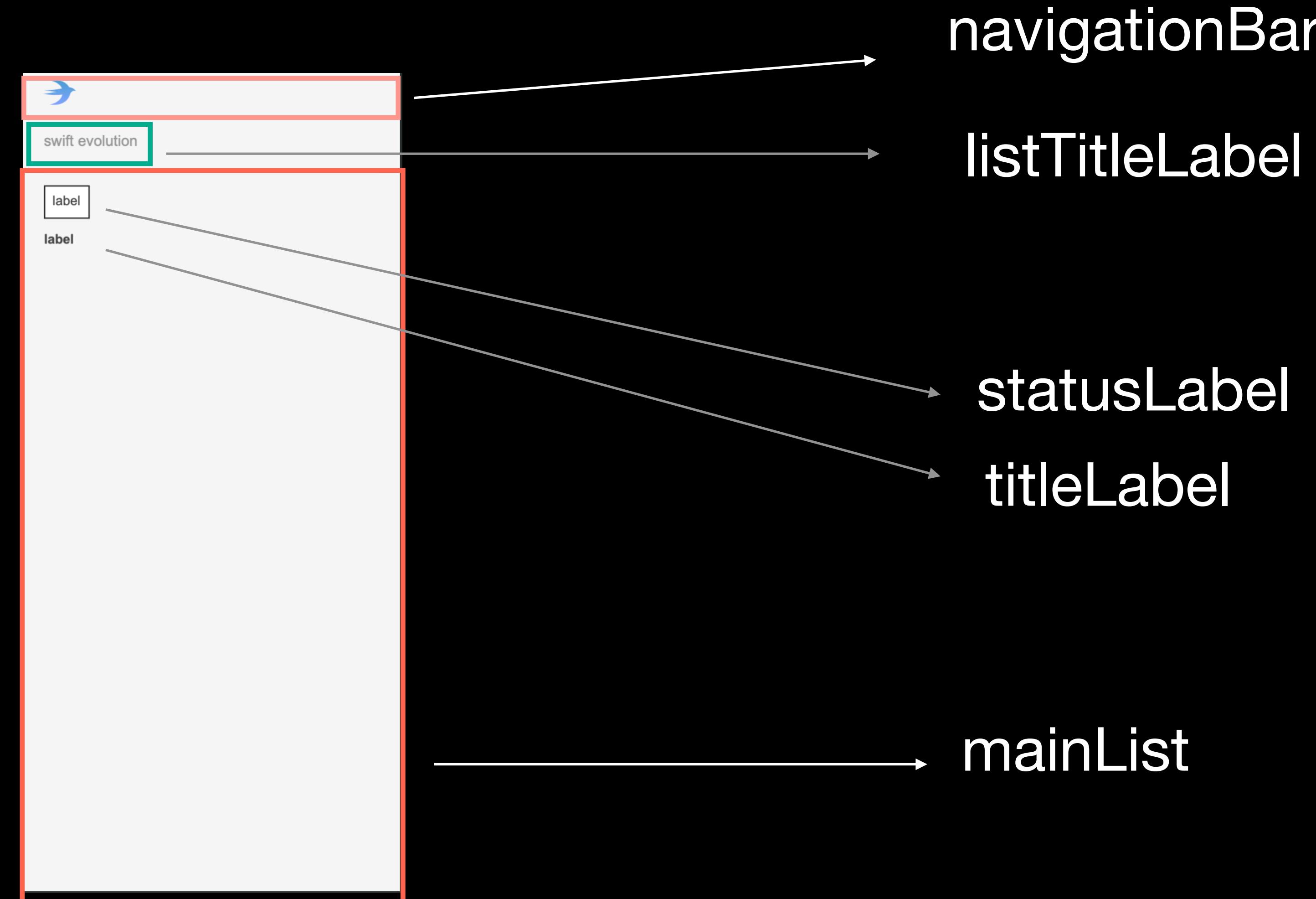
<https://www.swift.org/swift-evolution/>

# Scade 시작해볼까요?



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



# Scade 시작해볼까요?

```
mainList.elementProvider = SCDWidgetsElementProvider { (item: SwiftEvolutionEntity, template) in
    (template.getWidgetByName("statusLabel") as? SCDWidgetsLabel)?.text = item.status.transformedState
    (template.getWidgetByName("titleLabel") as? SCDWidgetsLabel)?.text = item.title
}
```

```
viewModel.networkingAction {
    self.mainList.items = self.viewModel.swiftEvolution
}
```

# Scade 시작해볼까요?

```
public func performRequest(context api: API) async throws -> Data {
    do {
        let request = try api.toRequest()
        let (data, response) = try await requestable.data(for: request)
        guard let httpResponse = response as? HTTPURLResponse else {
            throw NetworkingError(reason: .noResponse)
        }
        guard api.validationCode ~= httpResponse.statusCode else {
            throw StatusError(reason: StatusErrorReason(statusCode: httpResponse.statusCode))
        }
        return data
    } catch {
        guard let errorDict = api.errorDict,
              let statusError = error as? StatusError,
              let apiError = errorDict[statusError.reason.statusCode] else {
            throw error
        }
        throw apiError
    }
}
```



# Scade 시작해볼까요?

```
struct SwiftEvolutionResponseDTO: Decodable {  
    let authors: [Authors]  
    let id: String  
    let link: String  
    let reviewManager: ReviewManager?  
    let sha: String  
    let status: Status  
    let summary: String  
    let title: String  
    let trackingBugs: [TrackingBugs]?  
    let warnings: [Warnings]?  
}
```



# Scade 시작해볼까요?

```
import Foundation
import ScadeKit

public final class SwiftEvolutionEntity: EObject {
    public let authors: [AuthorsEntity]
    public let id: String
    public let link: String
    public let sha: String
    public let status: StatusEntity
    public let summary: String
    public let title: String
}
```



# Scade 시작해볼까요?

그래서 기존 iOS App을 Scade로  
マイグ레이션 할 수 있어요?



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

네 가능합니다!



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

기능 지원 여부 확인 필요



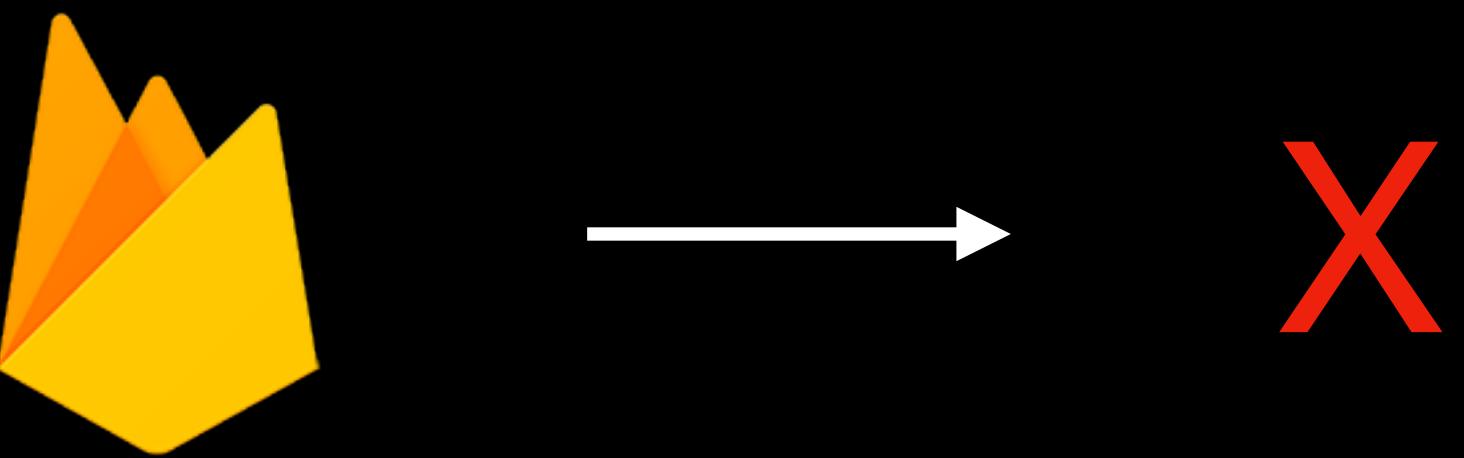
**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



```
fatal error: 'Foundation/Foundation.h' file not found
```



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?

기존 iOS UI 코드 -> Scade UI 코드

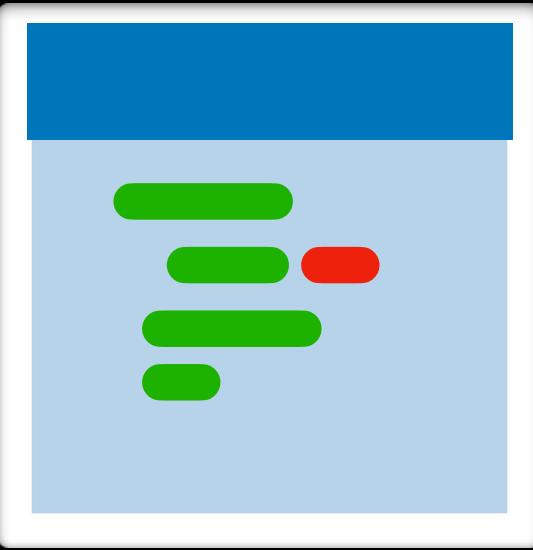


**Let'Swift 2023**  
Deep Dive into the unknown

# Scade 시작해볼까요?



+



Scade의 UI

기존 기능 관련 코드



Let'Swift 2023  
Deep Dive into the unknown

# Scade는 어떻게 동작해요?



**Let'Swift 2023**  
Deep Dive into the unknown

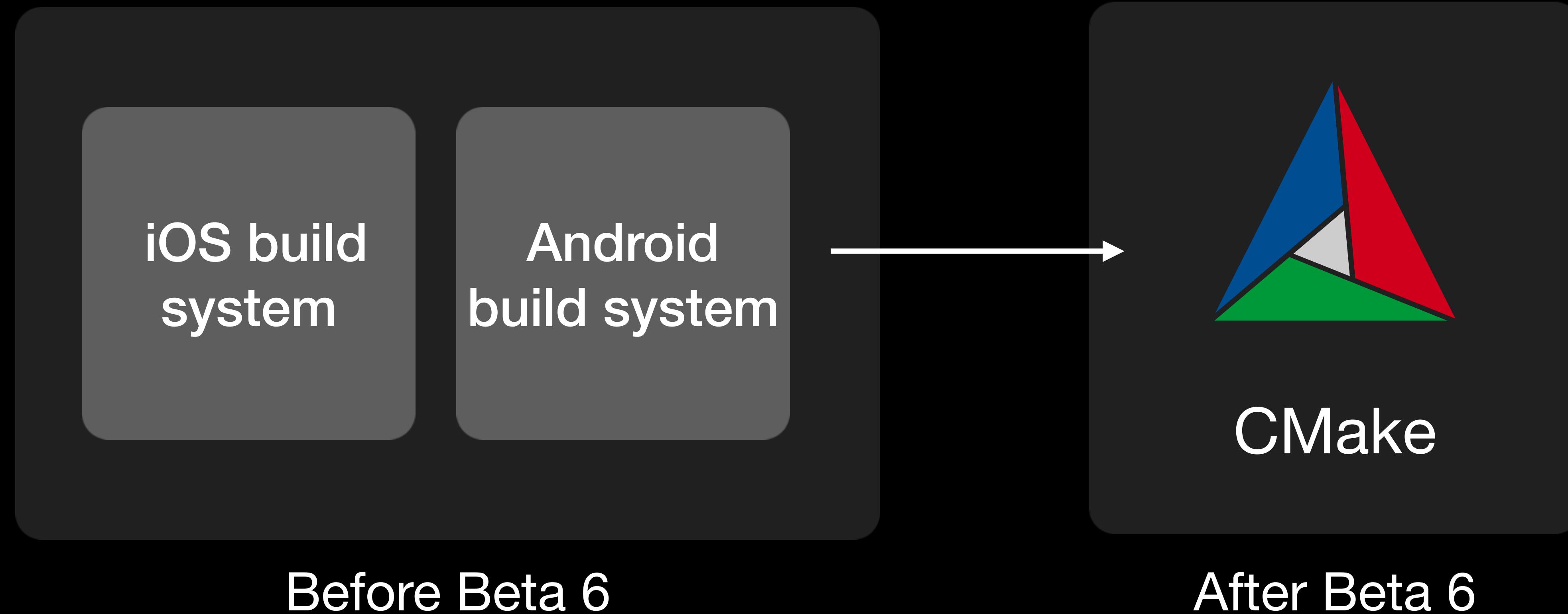
# Scade는 어떻게 동작해요?

## Scade Build System



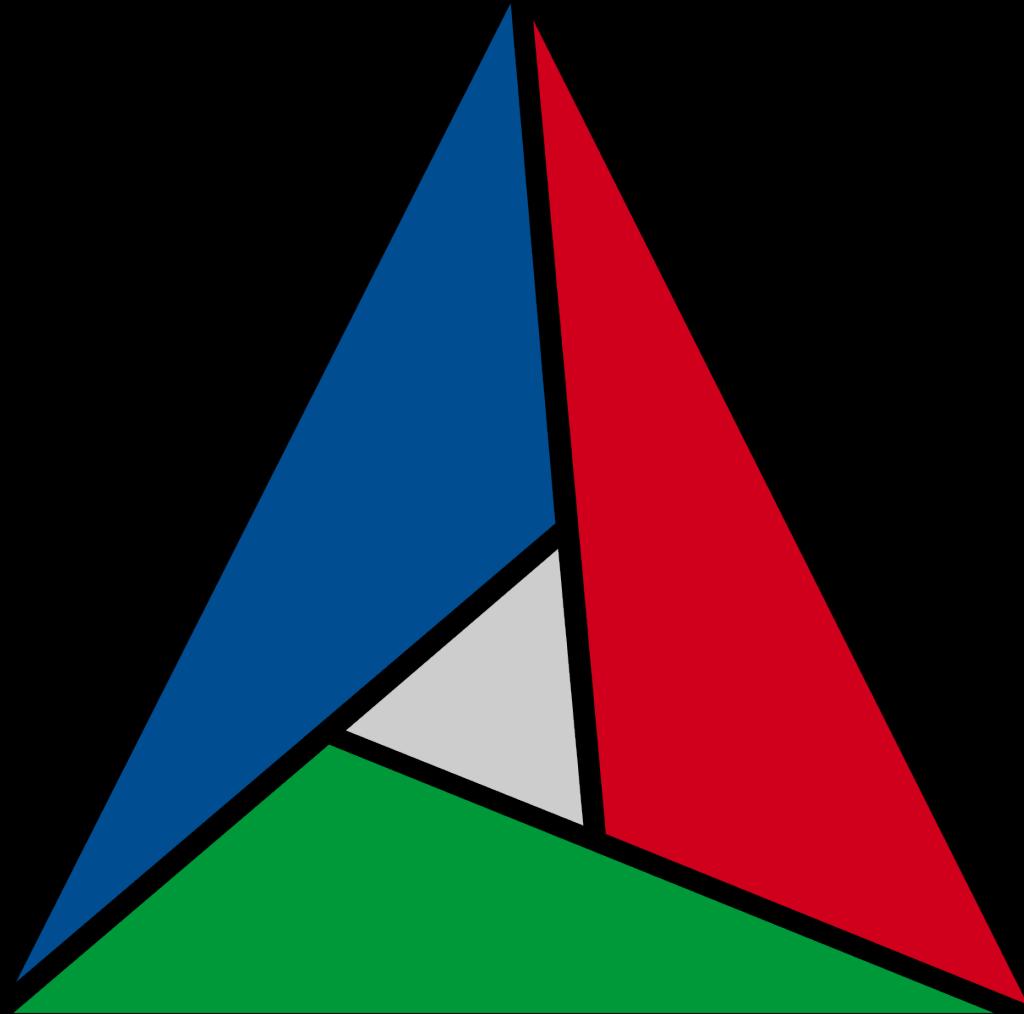
**Let'Swift 2023**  
Deep Dive into the unknown

# Scade는 어떻게 동작해요?



# Scade는 어떻게 동작해요?

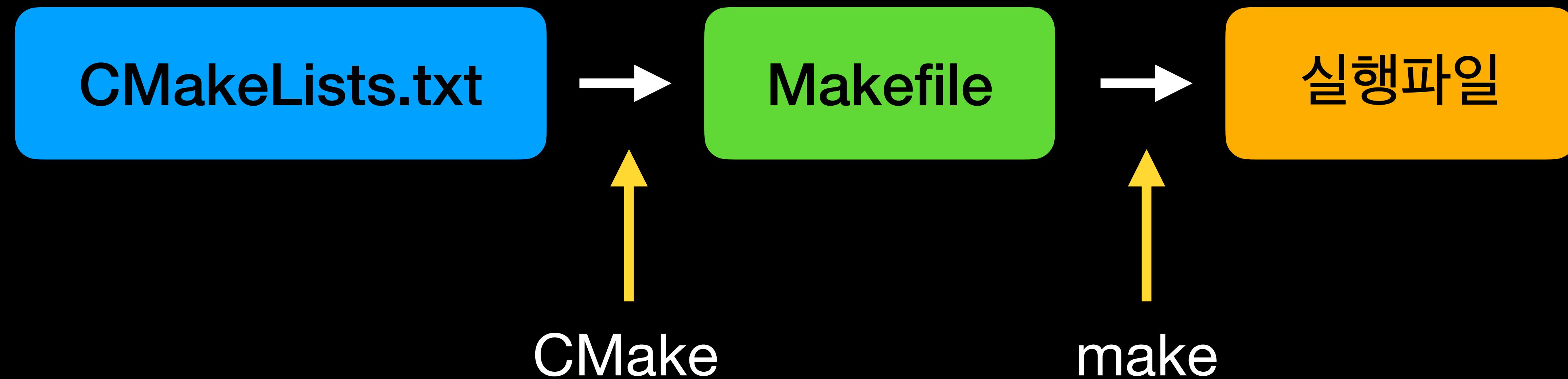
## CMake



- 크로스 플랫폼 빌드 자동화 도구
- Kitware라는 회사의 오픈소스 프로젝트



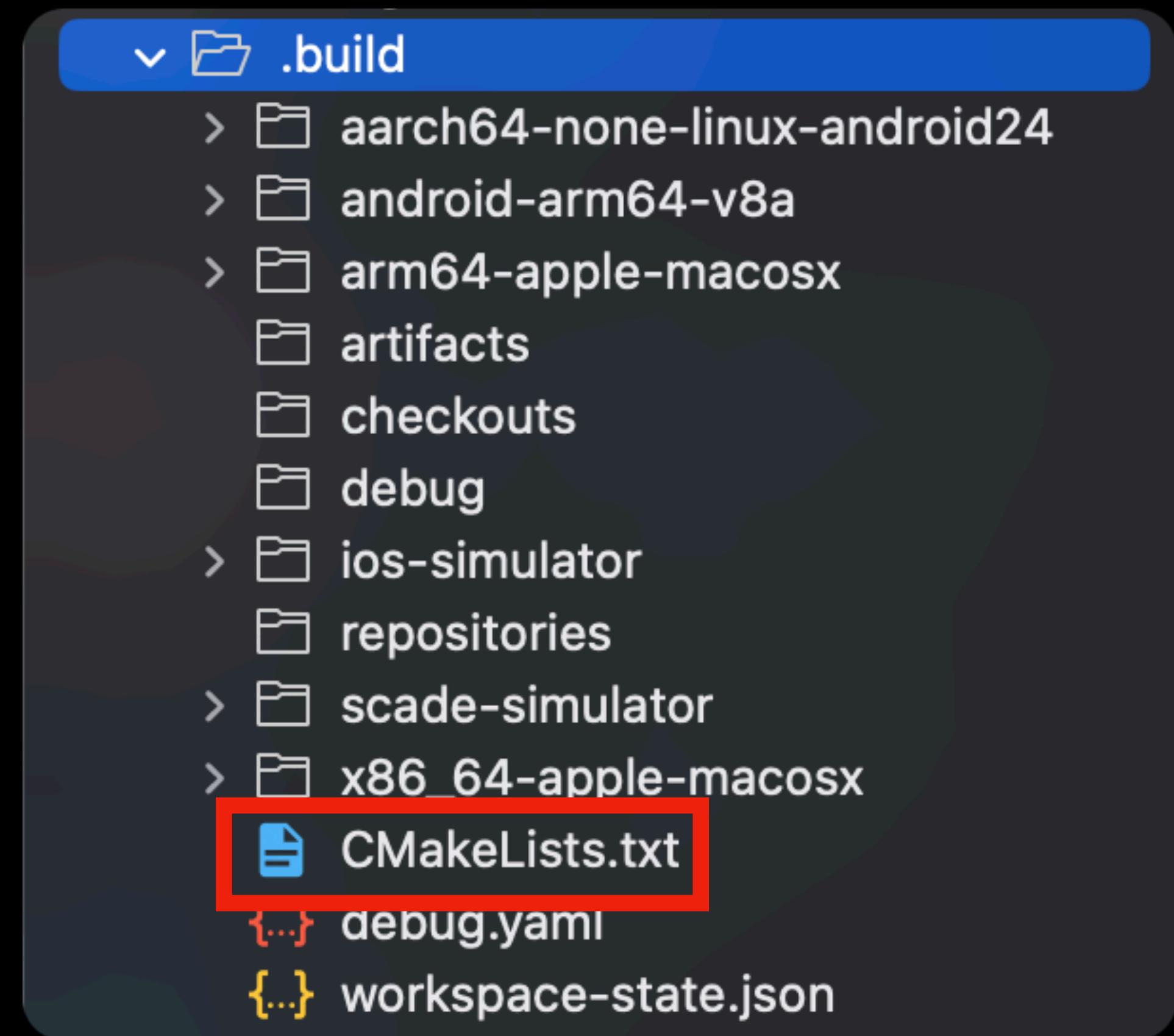
# Scade는 어떻게 동작해요?



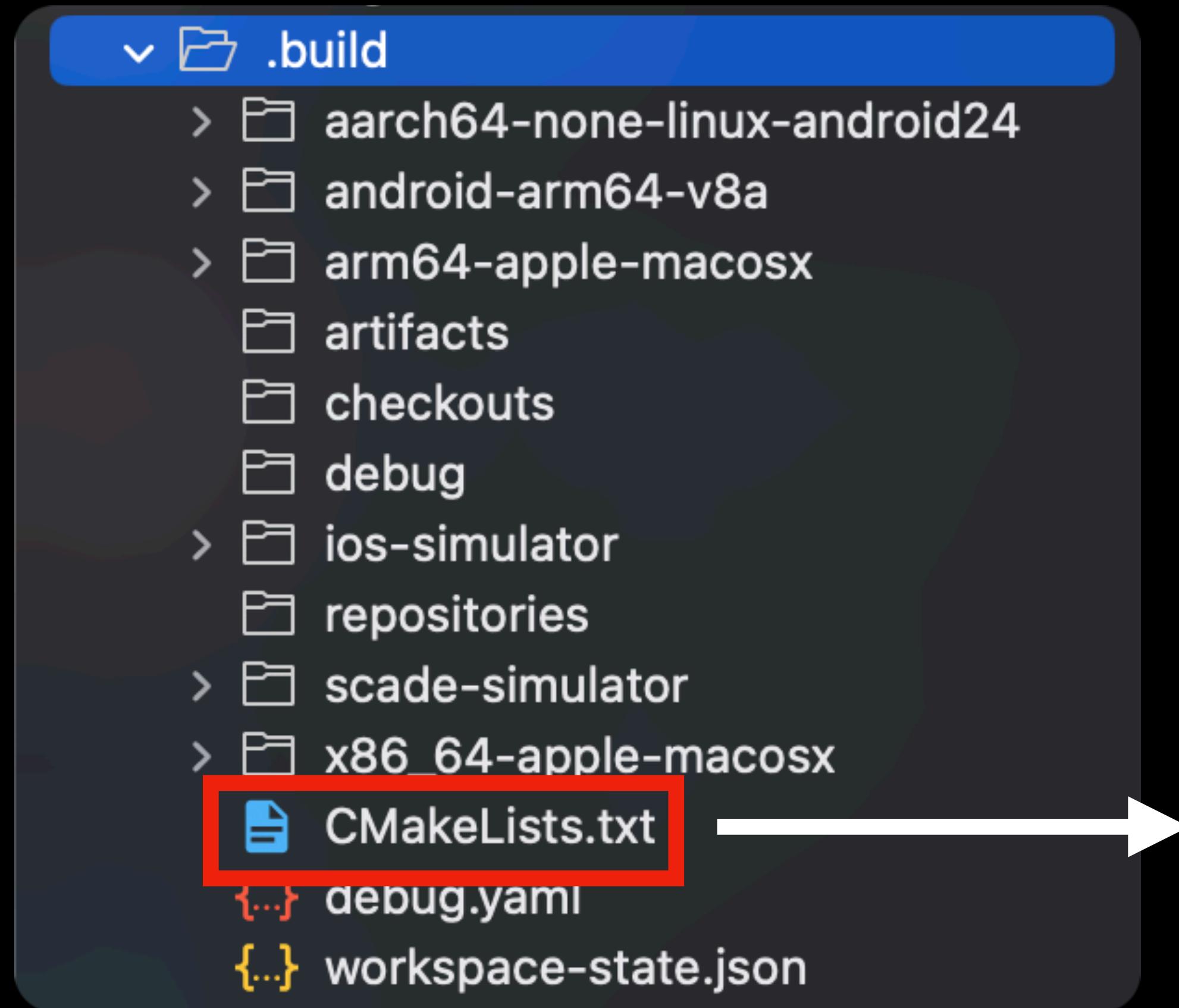
# Scade는 어떻게 동작해요?



# Scade는 어떻게 동작해요?



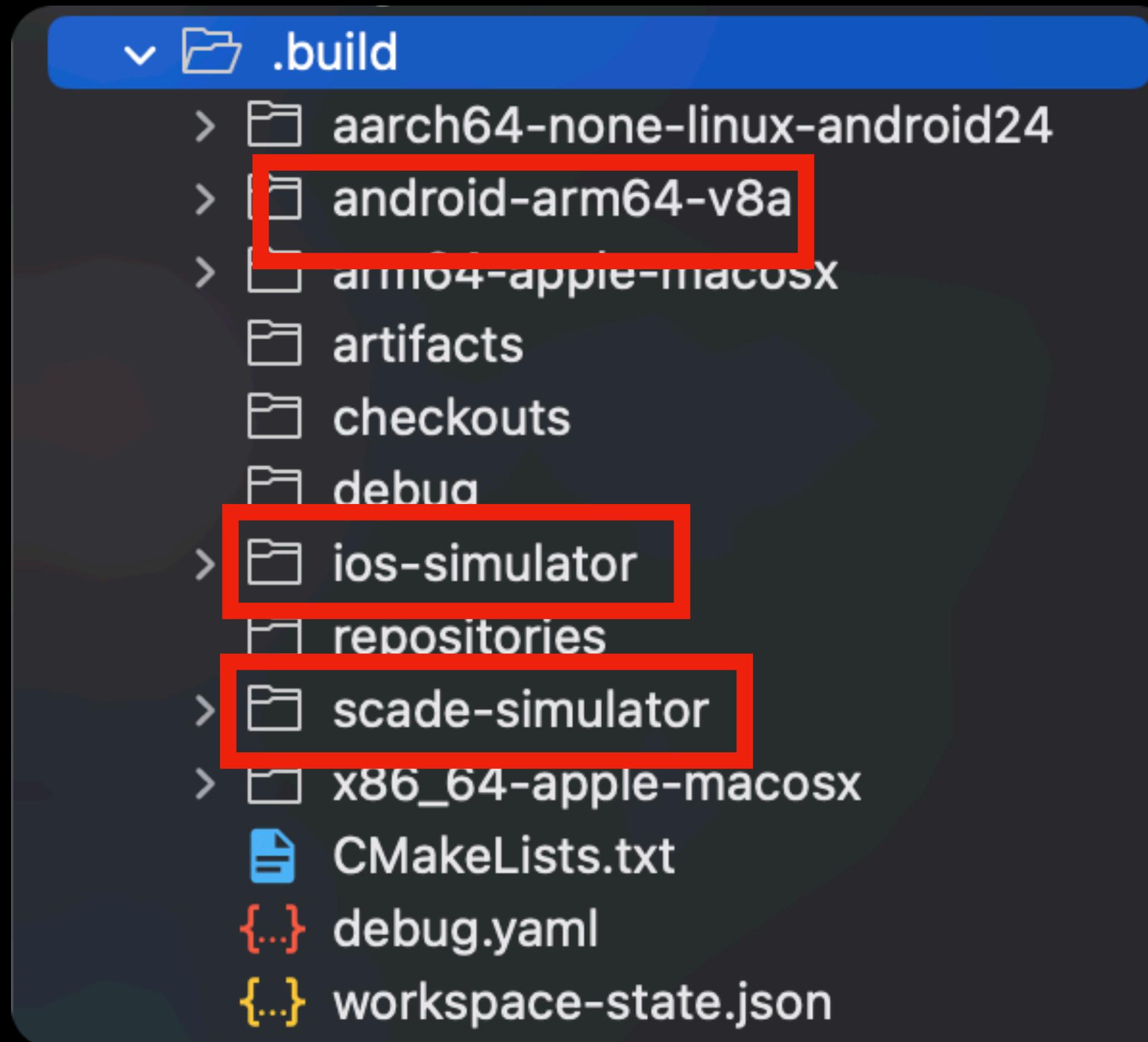
# Scade는 어떻게 동작해요?



```
cmake_minimum_required(VERSION 3.0)
project("Swifting")
include(ScadeSDK)
add_scade_application("Swifting" "com.jihoonahn.swifting"
    "${CMAKE_CURRENT_SOURCE_DIR}.."
    USE_SPM
    APP_NAME "Swifting"
    APP_SPM_NAME "Swifting"
    APPLE_DISPLAY_NAME "Swifting"
    ANDROID_DISPLAY_NAME "Swifting"
    SOURCES
    "Sources/Core/Networking/NetworkRequestable.swift"
    "Sources/Core/Networking/NetworkDependency.swift"
    "Sources/Core/Networking/Network.swift"
    "Sources/Core/Networking/HTTPMethod.swift"
    "Sources/Core/Networking/API.swift"
    "Sources/Core/Networking/StatusErrorReason.swift"
    "Sources/Core/Networking/NetworkError.swift"
    "Sources/Core/Networking/NetworkingErrorReason.swift"
    "Sources/Core/Utility/URL/BundleURL.swift"
    "Sources/Core/Utility/Logging/Logging.swift"
    "Sources/Swifting/App/AppDependency.swift"
    "Sources/Swifting/App/start.swift"
    "Sources/Swifting/PageAdapter/Home/home.page.dependency.swift"
    "Sources/Swifting/PageAdapter/Home/home.page.viewModel.swift"
    "Sources/Swifting/PageAdapter/Home/home.page.swift"
    "Sources/Swifting/PageAdapter/Splash/splash.page.swift"
```



# Scade는 어떻게 동작해요?



# Scade는 어떻게 동작해요?

## JVM 상호작용



**Let'Swift 2023**  
Deep Dive into the unknown

# Scade는 어떻게 동작해요?



+

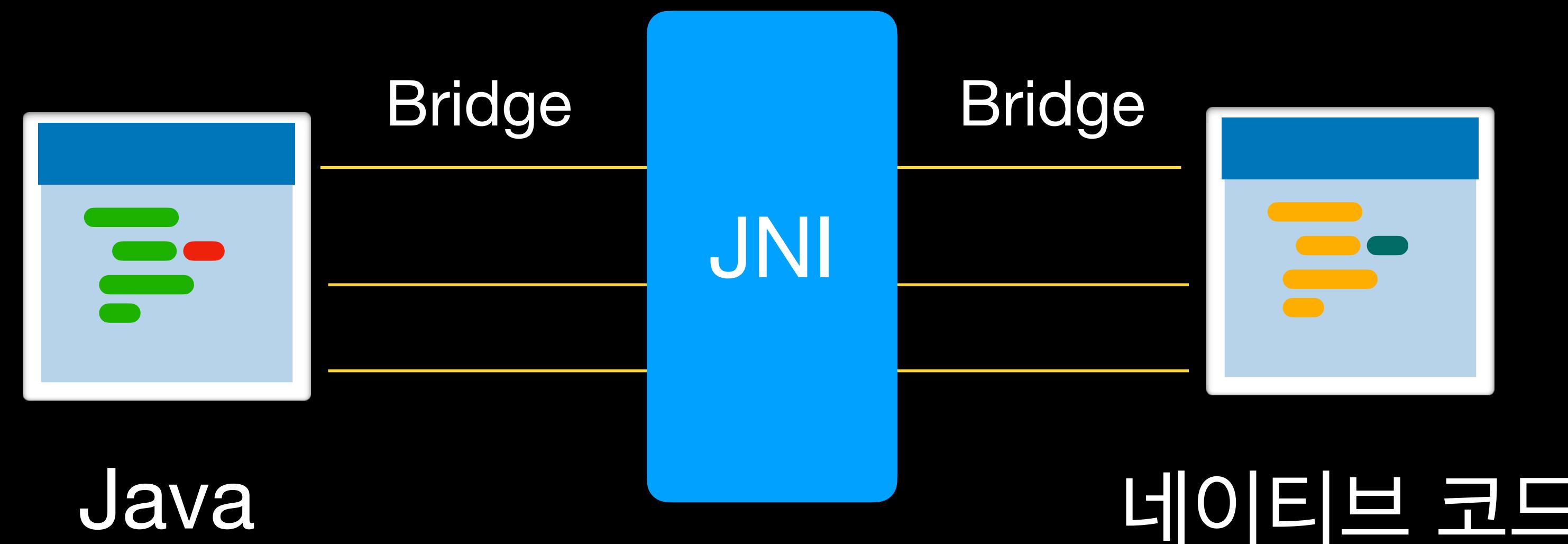


Android NDK

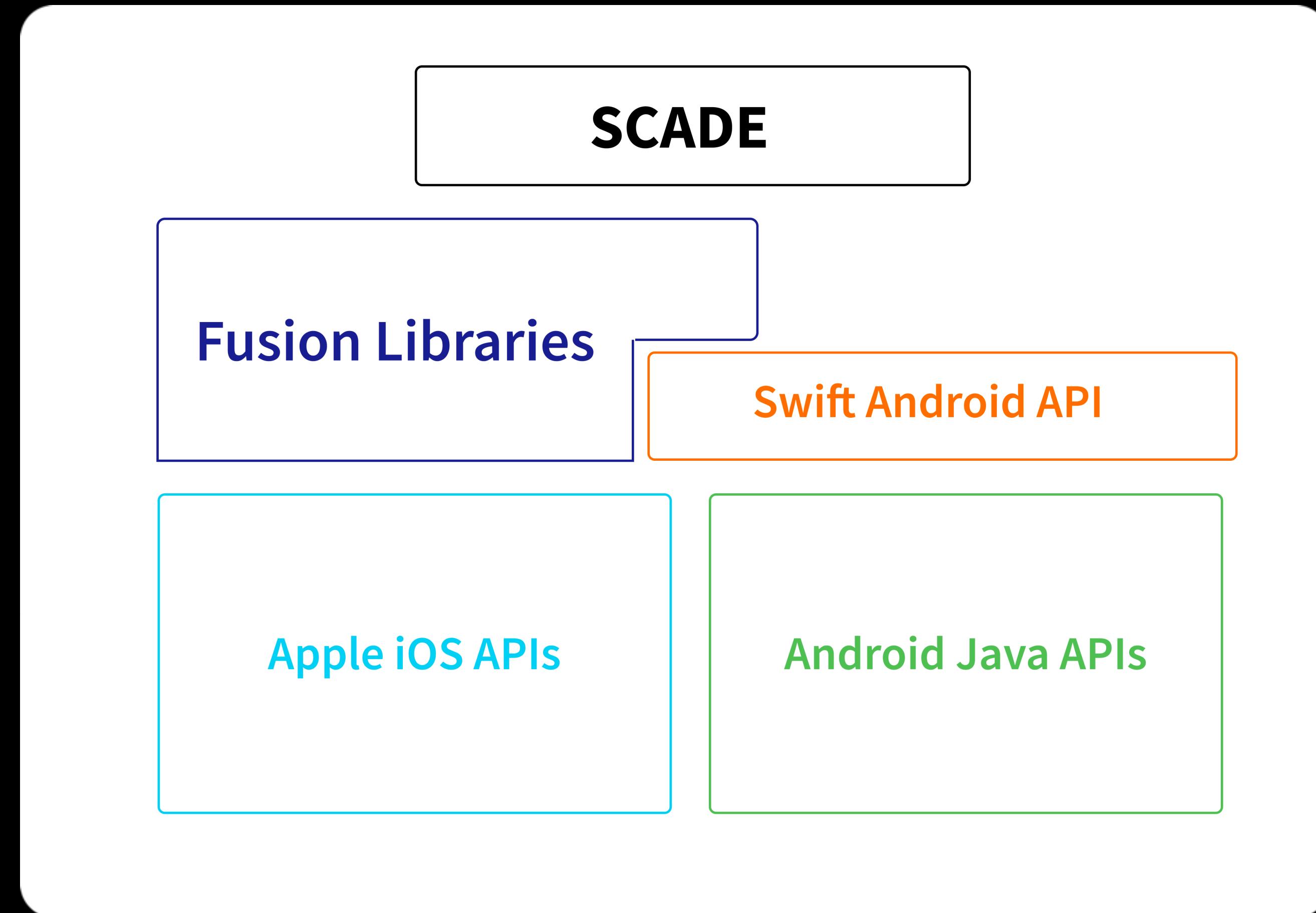


**Let'Swift 2023**  
Deep Dive into the unknown

# Scade는 어떻게 동작해요?



# Scade는 어떻게 동작해요?



# Scade는 어떻게 동작해요?

Android Java API

JNI

Swift Android API



**Let'Swift 2023**  
Deep Dive into the unknown

마무리

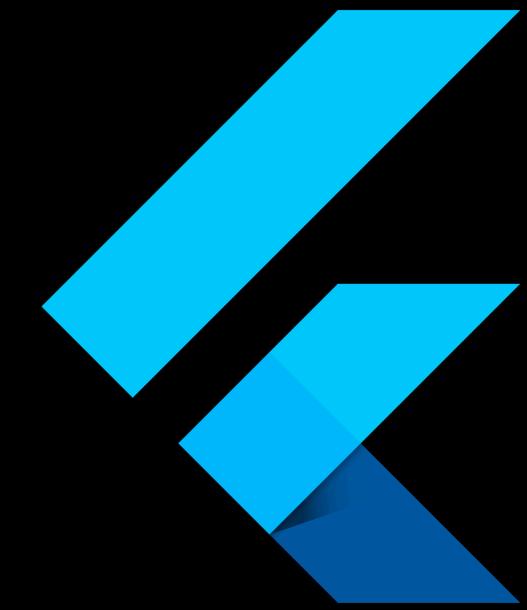


Let'Swift 2023  
Deep Dive into the unknown

# 마무리



# 마무리



오픈소스

9

Star

7

버전

Beta v2.3.1

오픈소스

7499

Star

157k

버전

Beta v3.14



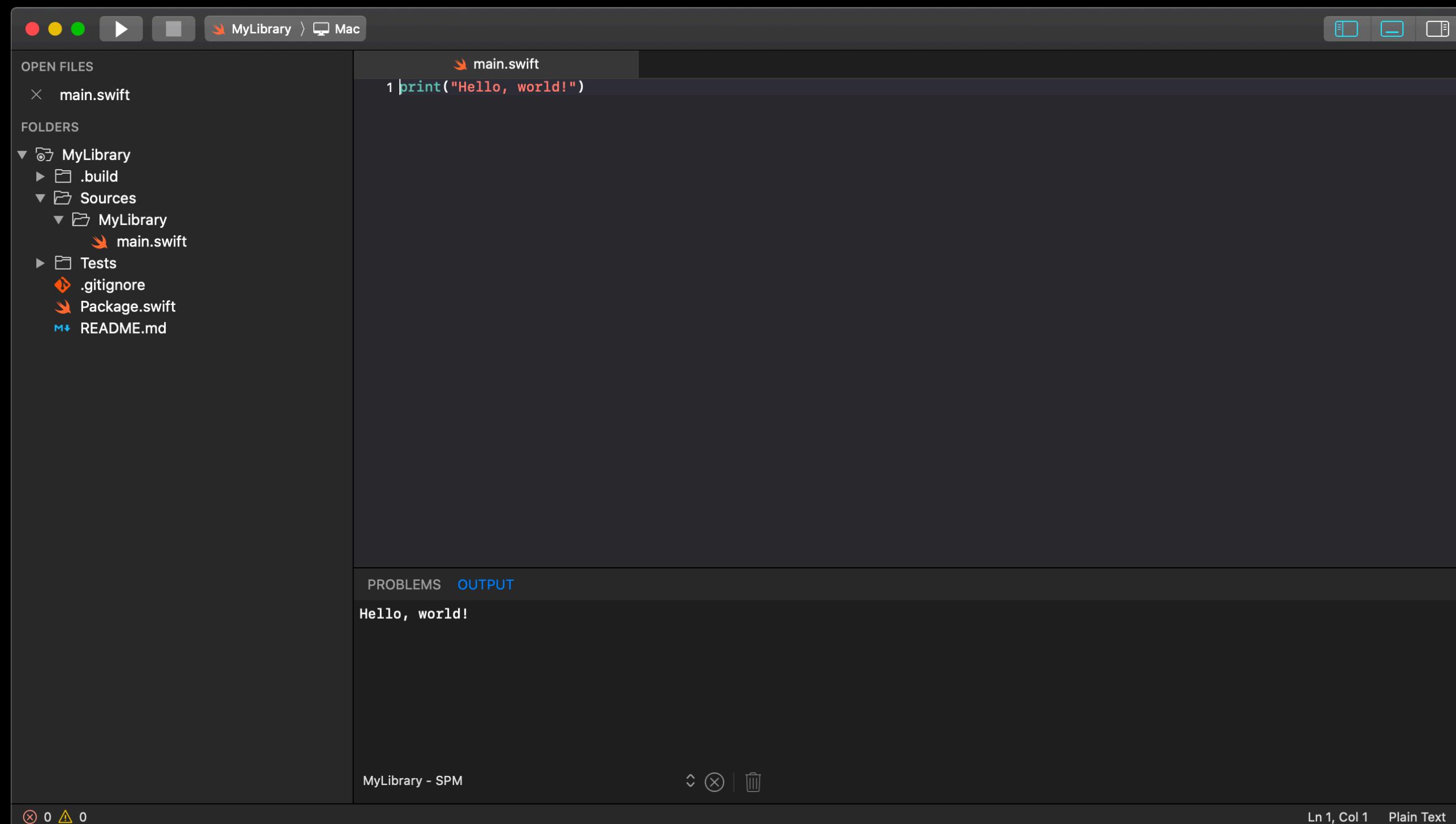
Let'Swift 2023  
Deep Dive into the unknown

# 마무리

Component	Support
Swift	Yes. Swift 5.x on both Android and iOS
Swift Foundation	Yes. Works on both Android and iOS
Swift Dispatch	Yes. Works on both Android and iOS
iOS frameworks - kit support	Yes. Each and every kit works on iOS. Your SCADE iOS app is never constrained and can use the full iOS SDK, ie. UIKit, PassKit ... without limitations.
Swift Libraries	Yes. All your favourite libraries that use pure Swift and Swift Foundation and Switch Dispatch work <b>on iOS and Android</b> . Most of your favourite libraries just work, i.e. SwiftData, SwiftMoment, SQLite, FileKit,....
SwiftUI	SwiftUI is a central concept and very important part of the future of SCADE. Currently, you can use the optional command directive to use SwiftUI in the iOS part of your SCADE app. For using SwiftUI to Android and iOS as a cross platform solution, we are working on an elegant solution which will be part of a later major release.
SPM	Yes. Works nicely.



# 마무리



A screenshot of the Xcode IDE. The project navigation sidebar on the left shows a folder structure for 'MyLibrary' containing '.build', 'Sources' (with 'MyLibrary' and 'main.swift'), 'Tests', '.gitignore', 'Package.swift', and 'README.md'. The main editor window on the right displays the 'main.swift' file with the single line of code: `1 print("Hello, world!")`. Below the editor is the 'PROBLEMS' and 'OUTPUT' tab, which shows the output 'Hello, world!'. At the bottom of the interface, there are SPM-related buttons and status indicators.



## Languages



<https://github.com/scade-platform/Nimble>

발표자료에 대해 피드백 해주신 리이오(Leeo)님과 오거나이저 분들 감사합니다.

# Thank You



**Let'Swift 2023**  
Deep Dive into the unknown



# Let'Swift 2023

Deep Dive into the unknown