



APP SCHOOL : iOS

앱 개발 기초

Swift와 Objective-C 섞어쓰기

A screenshot of a web browser window showing the Apple Developer documentation for Swift. The URL in the address bar is <https://developer.apple.com/documentation/swift/importing-objective-c-into-swift/>. The page title is "Importing Objective-C into Swift".

The left sidebar contains a navigation menu with the following items:

- Swift
- Importing Objective-C into Swift
- Importing Swift into Objective-C
- Cocoa Frameworks
- Working with Foundation Types
- Working with Core Foundation Types
- Objective-C APIs
 - Using Imported Lightweight Generics in Swift
 - Using Imported Protocol-Qualified Classes in ...
- C APIs
 - Using Imported C Structs and Unions in Swift
 - Using Imported C Functions in Swift
 - Using Imported C Macros in Swift
- Calling Objective-C APIs Asynchronously
- Mixing Swift and C++ in an Xcode project
- Using C++ APIs in Swift and Swift APIs in C++
- Operators
 - func < (((), ()) -> Bool
 - func < <A, B, C>((A, B, C), (A, B, C)) -> Bool
 - func > <A, B, C, D, E>((A, B, C, D, E), (A, B, C, D, E)) -> Bool
 - func < <A, B, C, D, E>((A, B, C, D, E), (A, B, C, D, E)) -> Bool

At the bottom of the sidebar are two input fields: "Filter" and a search bar with a magnifying glass icon.

The main content area shows the "Importing Objective-C into Swift" article. The title is "Importing Objective-C into Swift" and the subtitle is "Access classes and other declarations from your Objective-C code in Swift." Below the title is a large image of a smartphone displaying a code editor with Swift and Objective-C code.

Overview

You can use Objective-C and Swift files together in a single project, no matter which language the project used originally. This makes creating mixed-language app and framework targets as straightforward as creating an app or framework target written in a single language.

The diagram illustrates the import process from Objective-C to Swift. It shows three files: a .h file (Objective-C header), a .h file (Bridging header), and a .swift file (Swift source). An arrow points from the .h file to the .h file with the annotation "#import MyHeader.h". A dashed arrow points from the .h file to the .swift file, labeled "Bridging header".

The process for using your Objective-C declarations from your Swift code within mixed-language targets differs slightly depending on whether you're writing an app or a framework. Both processes are described below.

A screenshot of a web browser window showing the Apple Developer documentation for Swift. The URL in the address bar is <https://developer.apple.com/documentation/swift/importing-swift-into-objective-c>. The page title is "Importing Swift into Objective-C".

The left sidebar contains a navigation menu with the following sections and links:

- Swift**
 - Importing Objective-C into Swift
 - Importing Swift into Objective-C**
- Cocoa Frameworks
 - Working with Foundation Types
 - Working with Core Foundation Types
- Objective-C APIs
 - Using Imported Lightweight Generics in Swift
 - Using Imported Protocol-Qualified Classes in ...
- C APIs
 - Using Imported C Structs and Unions in Swift
 - Using Imported C Functions in Swift
 - Using Imported C Macros in Swift
- Calling Objective-C APIs Asynchronously
- Mixing Swift and C++ in an Xcode project
- Using C++ APIs in Swift and Swift APIs in C++
- Operators
 - func < (((), ()) -> Bool
 - func < <A, B, C>((A, B, C), (A, B, C)) -> Bool
 - func > <A, B, C, D, E>((A, B, C, D, E), (A, B, C, D, E)) -> Bool
 - func < <A, B, C, D, E>((A, B, C, D, E), (A, B, C, D, E)) -> Bool

At the bottom of the sidebar are two buttons: "Filter" and a search bar with a magnifying glass icon.

The main content area displays the article "Importing Swift into Objective-C" with the subtitle "Access Swift types and declarations from within your Objective-C codebase." Below the article is a section titled "Overview" with a sub-section icon. The text in the "Overview" section explains how to work with types declared in Swift from Objective-C by importing an Xcode-generated header file. It also describes the generation of the header's name based on the product module name and provides details about non-alphanumeric character replacement.

On the right side of the main content area, there is a diagram illustrating the import process. It shows a Swift file icon (labeled ".swift") connected by a line to an Objective-C header file icon (labeled ".h"). A callout bubble above the connection line contains the text "@class MySwiftClass", indicating that the Swift class is being imported into the Objective-C header.

Objective-C로 만든 클래스와 함수를 Swift로 작성한 프로젝트에서 호출하는 방법

UIKit 프로젝트에서

- 먼저, Objective-C 클래스와 함수를 Swift에서 사용할 수 있도록 브릿징 헤더 파일을 만들어야 합니다.
- 브릿징 헤더 파일을 만들었다면, 해당 파일을 프로젝트 설정에서 Objective-C Bridging Header 항목에 추가해야 합니다.
- 이제 오브젝티브씨 클래스와 함수를 스위프트에서 사용할 수 있습니다.
- 참고로, 브릿징 헤더 파일을 만드는 방법은 다음과 같습니다.
 - Xcode에서 File > New > File... 메뉴를 선택합니다.
 - iOS 탭에서 Header File을 선택하고 Next 버튼을 클릭합니다.
 - 파일 이름을 입력하고 Create 버튼을 클릭합니다.
 - 생성된 헤더 파일에 오브젝티브씨 클래스와 함수를 선언합니다

함께
해요

함께 해봅시다

함께
해요

Swift와 Objective-C 섞어쓰기

함께 해봅시다

- 다음의 앱을 만들어봅시다
- Swift로 작성한 ViewController가 있는 UIKit 앱 프로젝트를 만듭시다
- 이 앱에서는 숫자 두 개를 받아서 각종 계산을 합니다
 - 덧셈, 뺄셈, 곱셈, 나눗셈
- 실제 계산을 위한 코드는 Objective-C 클래스를 만들어 처리합시다
 - `(int)plus:(int)number1 with:(int)number2;`
 - `(int)minus:(int)number1 with:(int)number2;`
 - `(int)multiply:(int)number1 with:(int)number2;`
 - `(int)divide:(int)number1 with:(int)number2;`



감사합니다