

# 앱 개발 입문

성균관대학교 컬처앤테크놀로지융합전공 하계 부트캠프

1일차 - 2023. 06. 19 (월)

## 강의 슬라이드 링크

[https://github.com/kunny/skku-bootcamp-2023-summer/blob/  
main/\\_slides/day1.pdf](https://github.com/kunny/skku-bootcamp-2023-summer/blob/main/_slides/day1.pdf)

# 부트캠프에서 배울 내용 소개

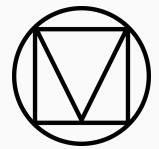




Dart



# Firebase



# MATERIAL DESIGN

11:52

# Sticky Notes

LTE

DEBUG



## About Flutter

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

## (제목 없음)

목요일까지 관리비 납부하기

## (제목 없음)

부트캠프 신청하기

## 만들어야 할 것들

- 파이어베이스 프로젝트
- 애드몹 계정
- 구글 플레이 개발자 계정
- 애플 개발자 계정



11:52

# Sticky Notes

LTE

DEBUG



## About Flutter

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

## (제목 없음)

부트캠프 신청하기

## (제목 없음)

목요일까지 관리비 납부하기

## 만들어야 할 것들

- 파이어베이스 프로젝트
- 애드몹 계정
- 구글 플레이 개발자 계정
- 애플 개발자 계정



11:52

## ← 노트 편집

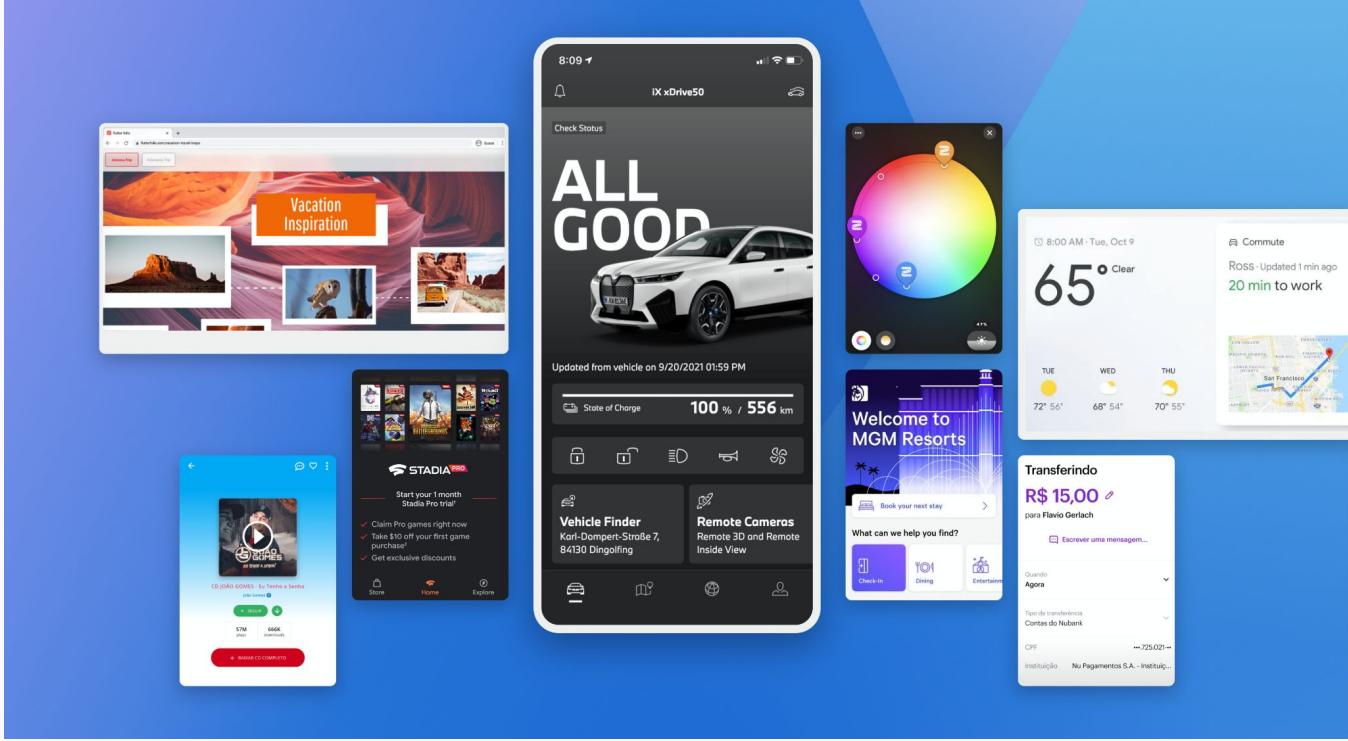
제목 입력

만들어야 할 것들

- 파이어베이스 프로젝트
- 애드몹 계정
- 구글 플레이 개발자 계정
- 애플 개발자 계정



# Flutter 소개



**Flutter** is an **open source** framework by Google for building beautiful, **natively compiled**, **multi-platform** applications from a **single codebase**.

# Open source

Screenshot of the GitHub repository page for [flutter/flutter](https://github.com/flutter/flutter).

The page shows the repository's main interface with sections for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights.

**Code** section:

- Branch: master (selected)
- Branches: 328 branches
- Tags: 504 tags
- Commits: 30,045 commits
- Latest commit: engine-flutter-autoroll Roll Flutter Engine from 8b48451ce023 to 0c84... by ed2f5d3 6 hours ago
- Recent commits include: .github/.github Update cherry-pick template to require PR links (#107856), bin/Roll Flutter Engine from 8b48451ce023 to 0c84d63523d4 (1 revision) 6 hours ago, dev/[web] define \$flutterDriverResult variable early in driver test inita... yesterday, examples/Remove outdated Fuchsia concepts (#107335) yesterday, packages/Add optional flag to determine assertiveness level in aria announcem... yesterday, .ci.yaml/Remove luci scheduling from openpay benchmarks (#108126) 2 days ago, .cirrus.yaml/Update cirrus secret. (#103957) 2 months ago, .gitattributes/Add pre-stable support for create on Windows (#51895) 2 years ago, .gitignore/[macOS] Adds macOS project files to layers example (#102539) 2 months ago, AUTHORS/[flutter\_driver] support send text input action (#106561) 9 days ago, CODEOWNERS/[CODEOWNERS] Remove caseyhiller from ci.yaml (#100642) 4 months ago, CODE\_OF\_CONDUCT.md/Update CODE\_OF\_CONDUCT.md (#94583) 8 months ago, CONTRIBUTING.md/Sync CONTRIBUTING.md with .github copy (#106677) 26 days ago, LICENSE/License update (#45373) 3 years ago, PATENT\_GRANT/Rename patent file (#38686) 3 years ago, README.md/Correct links and grammar in README.md (#108113) 2 days ago, TESTOWNERS/[openpay] Create an OpenPay benchmark (#107838) 2 days ago, analysis\_options.yaml/enable combinator ordering (#107847) 5 days ago

**About** section:

Flutter makes it easy and fast to build beautiful apps for mobile and beyond

**flutter.dev**

Tags: android, windows, macos, dart, ios, mobile, web, material-design, desktop, web-framework, app-framework, skia, linux-desktop, fuchsia, dart-platform

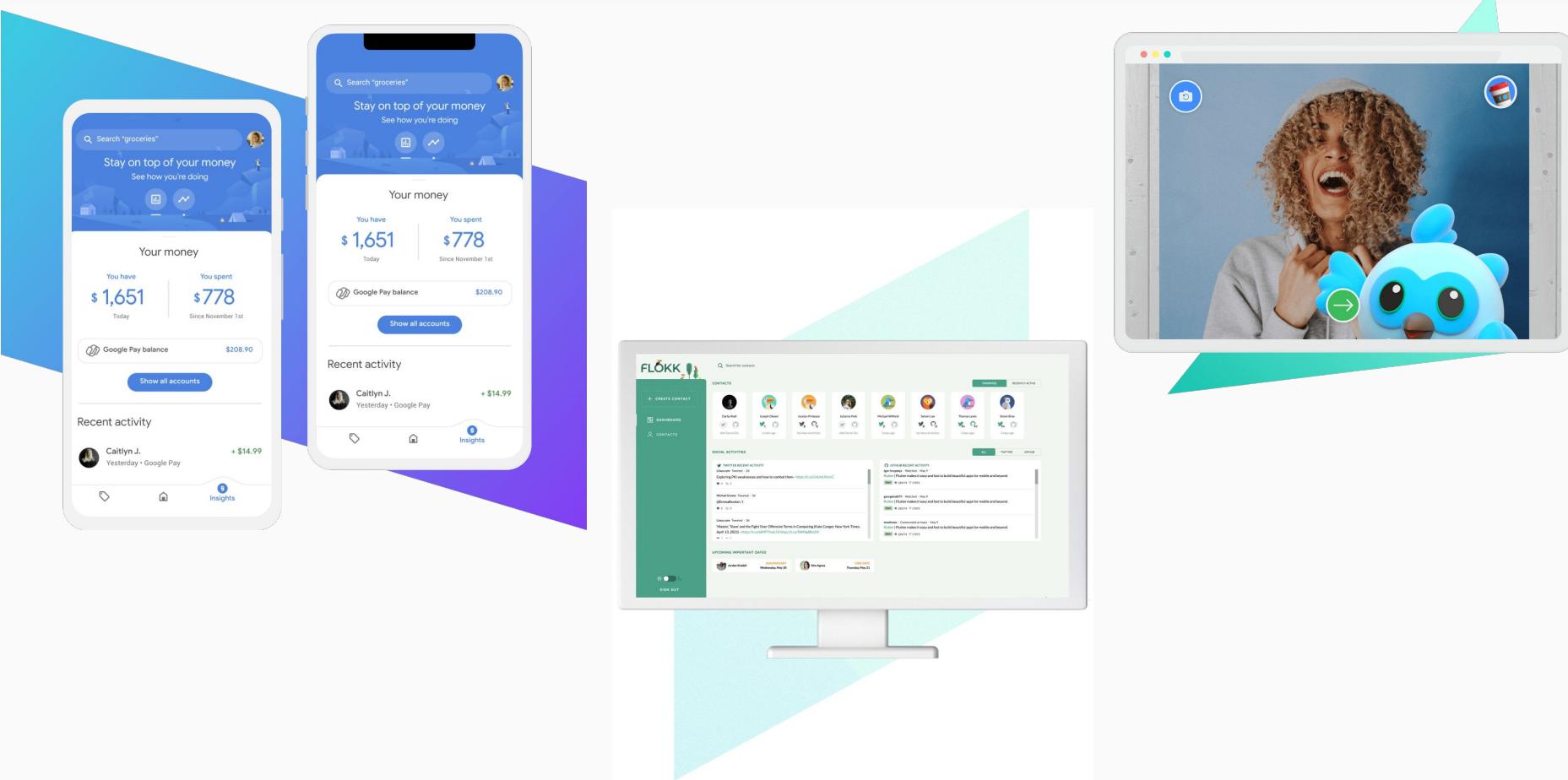
Links: Readme, BSD-3-Clause license, Code of conduct, 143k stars, 3.6k watching, 22.8k forks

**Used by** 8 projects

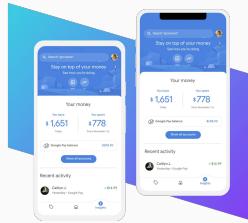
**Contributors** 1,049 contributors

<https://github.com/flutter/flutter>

# Multi-platform, single codebase



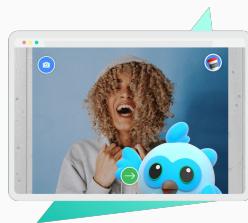
# Multi-platform, single codebase



- Android (Java / Kotlin)
- iOS (Objective-C / Swift)

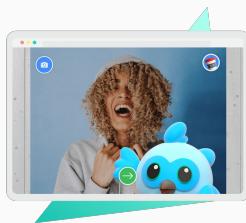
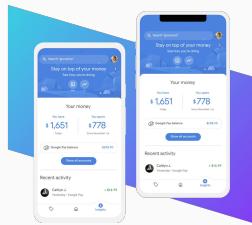


- Windows
- macOS
- Linux



- Angular
- React
- Vue.js

# Multi-platform, single codebase



# Flutter



# Dart

## Multi-platform frameworks: Why Flutter?

- 비 게임 기준으로 Flutter는 [46% 가량](#)의 시장 점유율을 보이고 있으며, 이는 React Native (29%)를 뛰어넘는 수준입니다.<sup>1</sup>
- 구글에서 제공하는 다양한 서비스 (Firebase, Google Play, AdMob)를 손쉽게 사용할 수 있으며, Material Design을 지원하므로 최소한의 노력으로 미려한 디자인의 앱을 만들 수 있습니다.
- 타 플랫폼 대비 활발히 개발되고 있으며, 공식 버전 출시 이후 세번의 판올림(3.x)을 진행했습니다. (참고: React Native의 최신 버전은 아직 0.71입니다)



```
print('Hello, World!')
```



```
void main() {  
    print('Hello, World!');  
}
```



```
class Person:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age  
  
p1 = Person("John", 36)
```



```
class Person {  
    final String name;  
    final int age;  
    Person(this.name, this.age);  
}  
  
Person p1 = Person("John", 36);
```

# 개발 환경 설정 절차

1. 앱 코드를 편집할 때 사용하는 [안드로이드 스튜디오](#) 설치 및 초기 설정
2. 안드로이드 스튜디오에 Flutter 앱 개발 기능([Flutter 플러그인](#)) 추가하기
3. [안드로이드 SDK 커맨드라인 도구](#) 설치 (안드로이드 앱 개발시 필요)
4. Flutter 앱 개발에 필요한 도구가 포함된 [Flutter SDK](#) 설치
5. Flutter SDK 동작에 필요한 도구인 [Git](#) 설치 (Windows만)
6. Flutter SDK [구성 완료하기](#)
7. [안드로이드 에뮬레이터](#) 설정하기

# 안드로이드 스튜디오 설치

# 안드로이드 스튜디오 설치 (Windows)

developers  Platform Android Studio Google Play Jetpack Kotlin Docs Games Search Language  로그인

ANDROID STUDIO

Download What's new User guide Preview



Android Studio provides the fastest tools for building apps on every type of Android device.

[Download Android Studio](#)

Android Studio Bumblebee | 2021.1.1 Patch 1 for Windows 64-bit (872 MB)

[Download options](#) [Release notes](#)

[d.android.com/studio](https://d.android.com/studio)

# 안드로이드 스튜디오 설치 (Windows)

tees) arising out of or accruing from (a) your use of the SDK, (b) any application you develop on the SDK that infringes any copyright, trademark, trade secret, trade dress, patent or other intellectual property right of any person or defames any person or violates their rights of publicity or privacy, and (c) any non-compliance by you with the License Agreement.

## 13. Changes to the License Agreement

13.1 Google may make changes to the License Agreement as it distributes new versions of the SDK. When these changes are made, Google will make a new version of the License Agreement available on the website where the SDK is made available.

## 14. General Legal Terms

14.1 The License Agreement constitutes the whole legal agreement between you and Google and governs your use of the SDK (excluding any services which Google may provide to you under a separate written agreement), and completely replaces any prior agreements between you and Google in relation to the SDK. 14.2 You agree that if Google does not exercise or enforce any legal right or remedy which is contained in the License Agreement (or which Google has the benefit of under any applicable law), this will not be taken to be a formal waiver of Google's rights and that those rights or remedies will still be available to Google. 14.3 If any court of law, having the jurisdiction to decide on this matter, rules that any provision of the License Agreement is invalid, then that provision will be removed from the License Agreement without affecting the rest of the License Agreement. The remaining provisions of the License Agreement will continue to be valid and enforceable. 14.4 You acknowledge and agree that each member of the group of companies of which Google is the parent shall be third party beneficiaries to the License Agreement and that such other companies shall be entitled to directly enforce, and rely upon, any provision of the License Agreement that confers a benefit on (or rights in favor of) them. Other than this, no other person or company shall be third party beneficiaries to the License Agreement. 14.5 EXPORT RESTRICTIONS. THE SDK IS SUBJECT TO UNITED STATES EXPORT LAWS AND REGULATIONS. YOU MUST COMPLY WITH ALL DOMESTIC AND INTERNATIONAL EXPORT LAWS AND REGULATIONS THAT APPLY TO THE SDK. THESE LAWS INCLUDE RESTRICTIONS ON DESTINATIONS, END USERS AND END USE.

14.6 The rights granted in the License Agreement may not be assigned or transferred by either you or Google without the prior written approval of the other party. Neither you nor Google shall be permitted to delegate their responsibilities or obligations under the License Agreement without the prior written approval of the other party. 14.7 The License Agreement, and your relationship with Google under the License Agreement, shall be governed by the laws of the State of California without regard to its conflict of laws provisions. You and Google agree to submit to the exclusive jurisdiction of the courts located within the county of Santa Clara, California to resolve any legal matter arising from the License Agreement. Notwithstanding this, you agree that Google shall still be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction. July 27, 2021



I have read and agree with the above terms and conditions

[Download Android Studio Bumblebee 2021.1.1 Patch 1 for Windows](#)

android-studio-2021.1.1.21-windows.exe

# 안드로이드 스튜디오 설치 (Windows)

안드로이드 스튜디오 설치 마법사 실행 후, 기본 옵션 그대로 설치합니다.



# 안드로이드 스튜디오 설치 (macOS)

developers  Platform Android Studio Google Play Jetpack 더보기 ▾

검색

한국어

ANDROID STUDIO

Download What's new User guide Preview

# android studio



Android Studio provides the fastest tools for building apps on every type of Android device.

Download Android Studio

Android Studio Chipmunk | 2021.2.1 Patch 1 for Mac (~1017 MiB)

Download options Release notes

d.android.com/studio

# 안드로이드 스튜디오 설치 (macOS)

14.1 The License Agreement constitutes the whole legal agreement between you and Google and governs your use of the SDK (excluding any services which Google may provide to you under a separate written agreement), and completely replaces any prior agreements between you and Google in relation to the SDK. 14.2 You agree that if Google does not exercise or enforce any legal right or remedy which is contained in the License Agreement (or which Google has the benefit of under any applicable law), this will not be taken to be a formal waiver of Google's rights and that those rights or remedies will still be available to Google. 14.3 If any court of law, having the jurisdiction to decide on this matter, rules that any provision of the License Agreement is invalid, then that provision will be removed from the License Agreement without affecting the rest of the License Agreement. The remaining provisions of the License Agreement will continue to be valid and enforceable. 14.4 You acknowledge and agree that each member of the group of companies of which Google is the parent shall be third party beneficiaries to the License Agreement and that such other companies shall be entitled to directly enforce, and rely upon, any provision of the License Agreement that confers a benefit on (or rights in favor of) them. Other than this, no other person or company shall be third party beneficiaries to the License Agreement. 14.5 EXPORT RESTRICTIONS. THE SDK IS SUBJECT TO UNITED STATES EXPORT LAWS AND REGULATIONS. YOU MUST COMPLY WITH ALL DOMESTIC AND INTERNATIONAL EXPORT LAWS AND REGULATIONS THAT APPLY TO THE SDK. THESE LAWS INCLUDE RESTRICTIONS ON DESTINATIONS, END USERS AND END USE. 14.6 The rights granted in the License Agreement may not be assigned or transferred by either you or Google without the prior written approval of the other party. Neither you nor Google shall be permitted to delegate their responsibilities or obligations under the License Agreement without the prior written approval of the other party. 14.7 The License Agreement, and your relationship with Google under the License Agreement, shall be governed by the laws of the State of California without regard to its conflict of laws provisions. You and Google agree to submit to the exclusive jurisdiction of the courts located within the county of Santa Clara, California to resolve any legal matter arising from the License Agreement. Notwithstanding this, you agree that Google shall still be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction. July 27, 2021



I have read and agree with the above terms and conditions

## Select the version of Android Studio that's right for your Mac:

Android Studio Chipmunk 2021.2.1 Patch 1

Mac with Intel chip

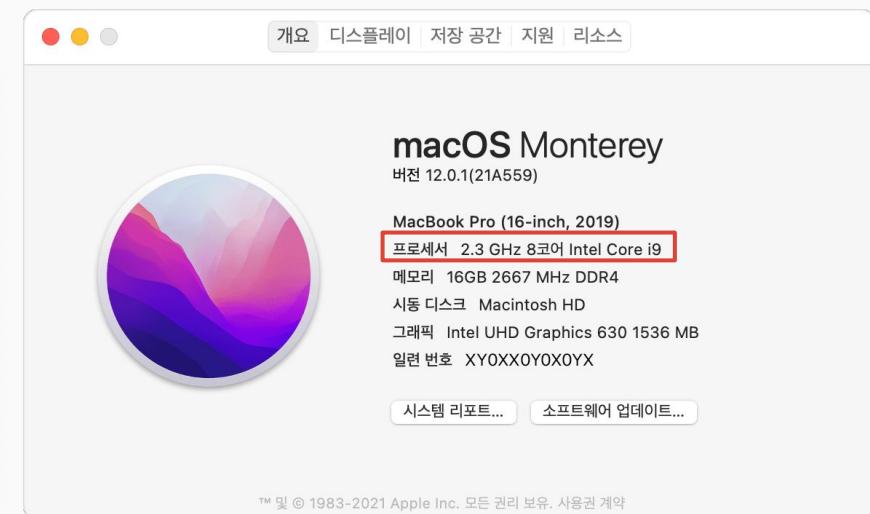
Mac with Apple chip

[android-studio-2021.2.1.15-mac.dmg](#)

[android-studio-2021.2.1.15-mac\\_arm.dmg](#)

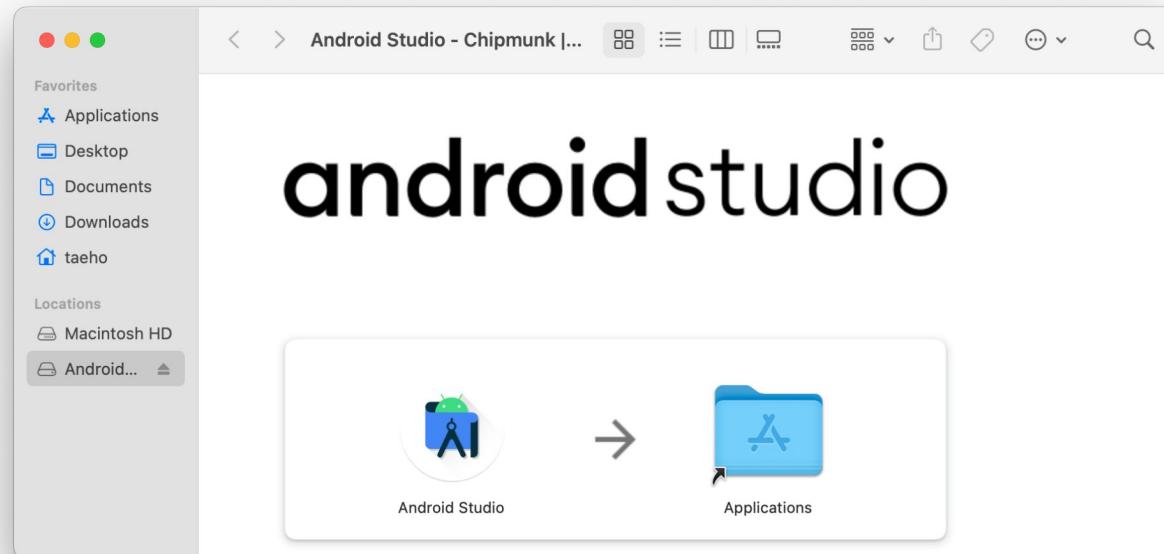
# 안드로이드 스튜디오 설치 (macOS)

컴퓨터가 사용하는 칩 종류 확인 (Apple 로고 > 이 Mac에 관하여)



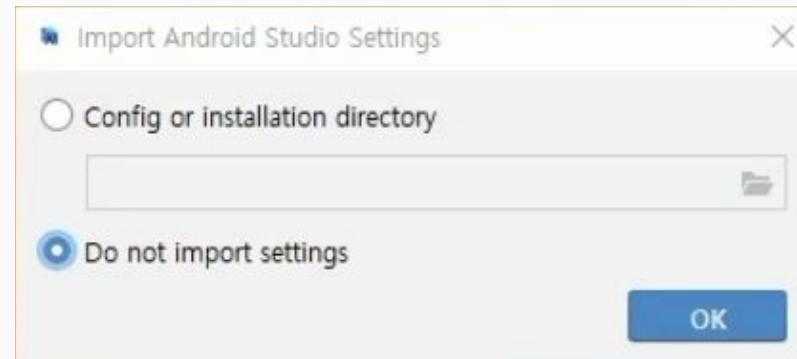
# 안드로이드 스튜디오 설치 (macOS)

다운로드한 \*.dmg 파일을 연 후, 안드로이드 스튜디오를 애플리케이션 폴더에 복사합니다. (Drag & Drop)



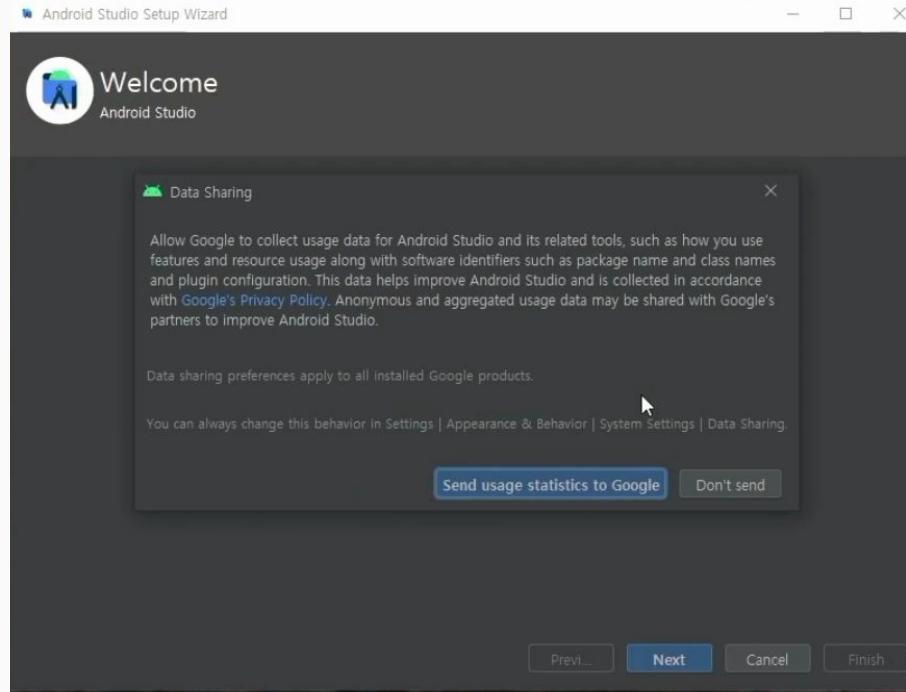
# 안드로이드 스튜디오 초기 설정

Import Android Studio Settings 파일로그에서 **Do not import settings**을 선택합니다.



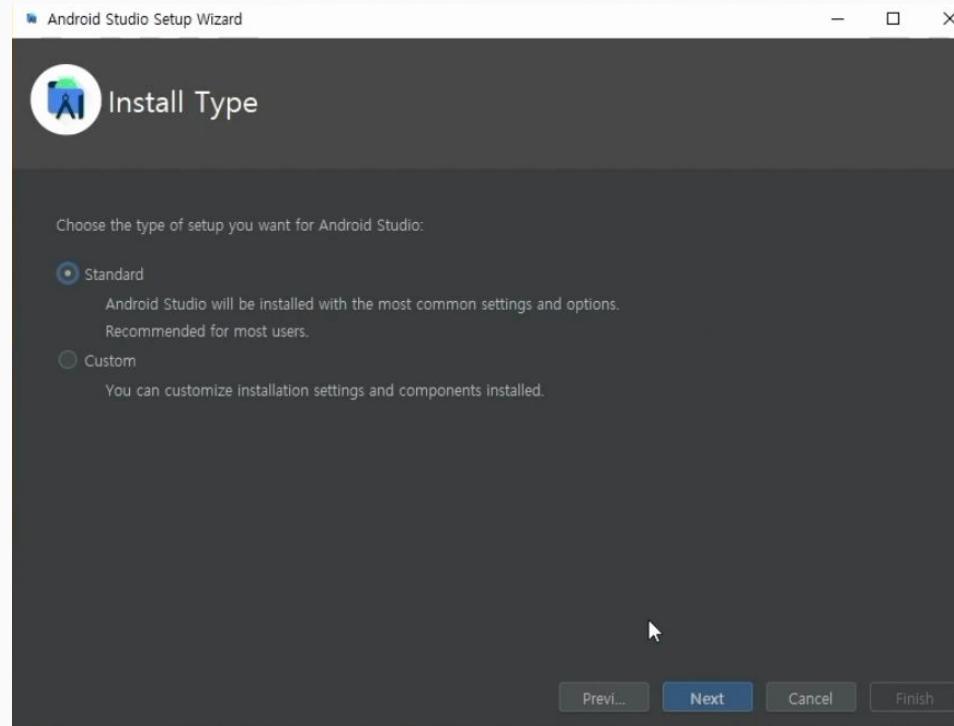
# 안드로이드 스튜디오 초기 설정

Data Sharing 설정에서 **원하는 옵션**을 선택합니다. (개발 툴 사용 통계 수집 동의 여부)



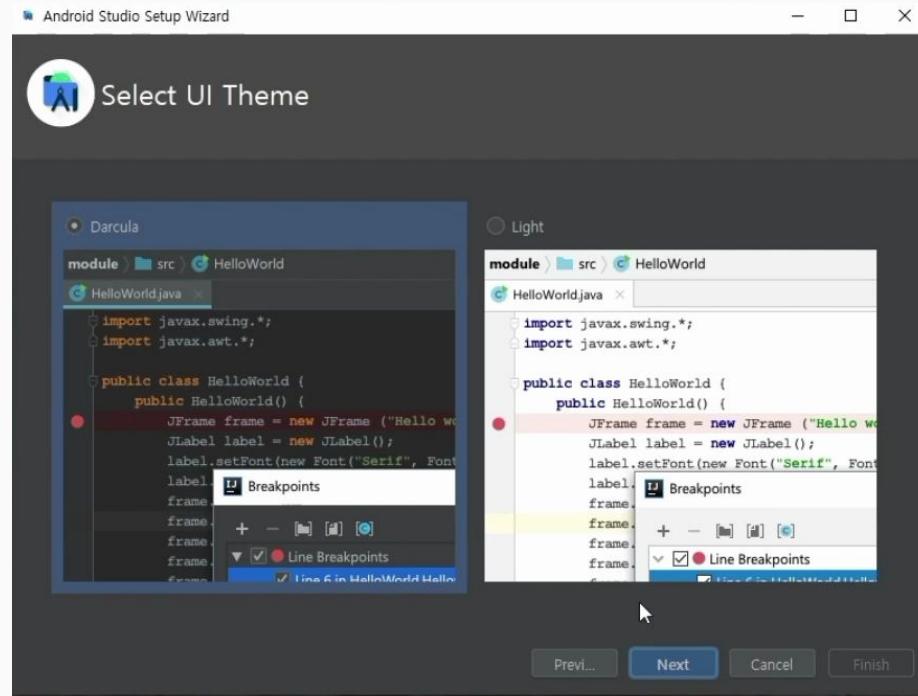
# 안드로이드 스튜디오 초기 설정

초기 설정에 필요한 추가 구성요소를 설치합니다. Install Type에서 **Standard**를 선택합니다.



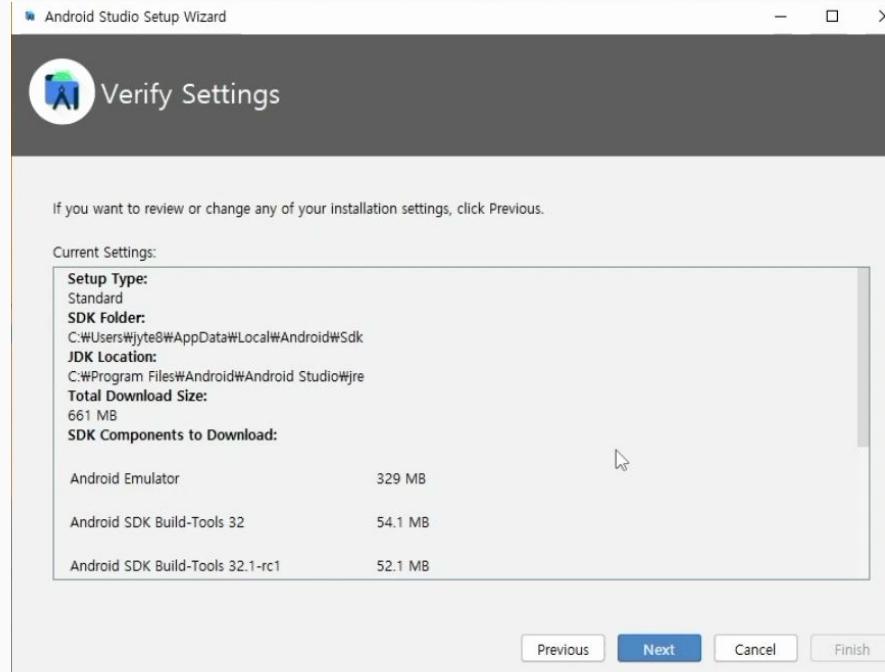
# 안드로이드 스튜디오 초기 설정

원하는 테마를 선택합니다.



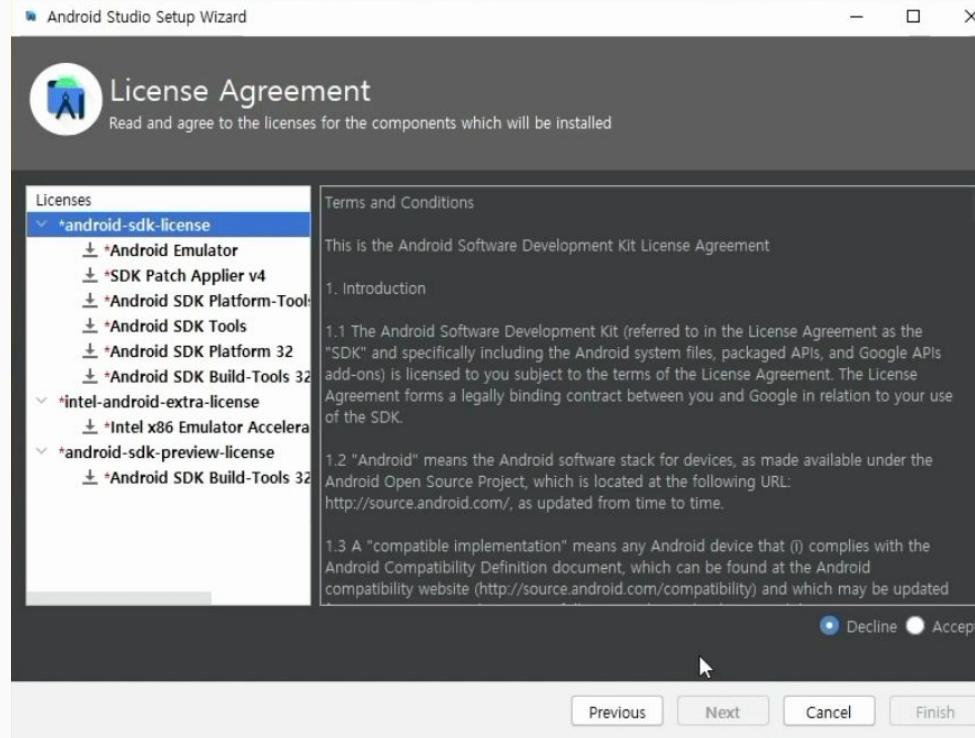
# 안드로이드 스튜디오 초기 설정

설치할 구성요소를 확인할 수 있습니다.



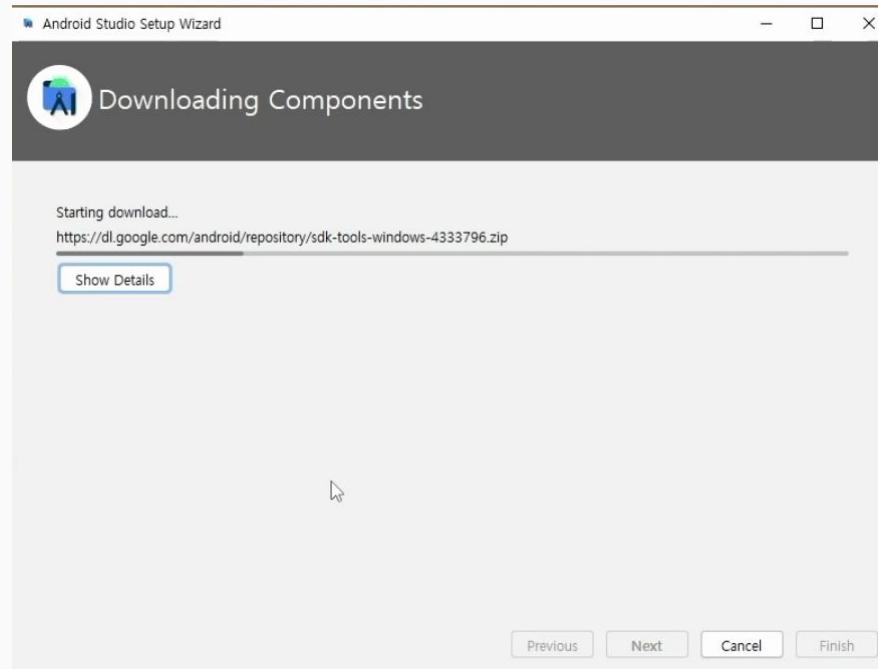
# 안드로이드 스튜디오 초기 설정

설치할 구성요소의 라이선스 약관을 수락합니다.



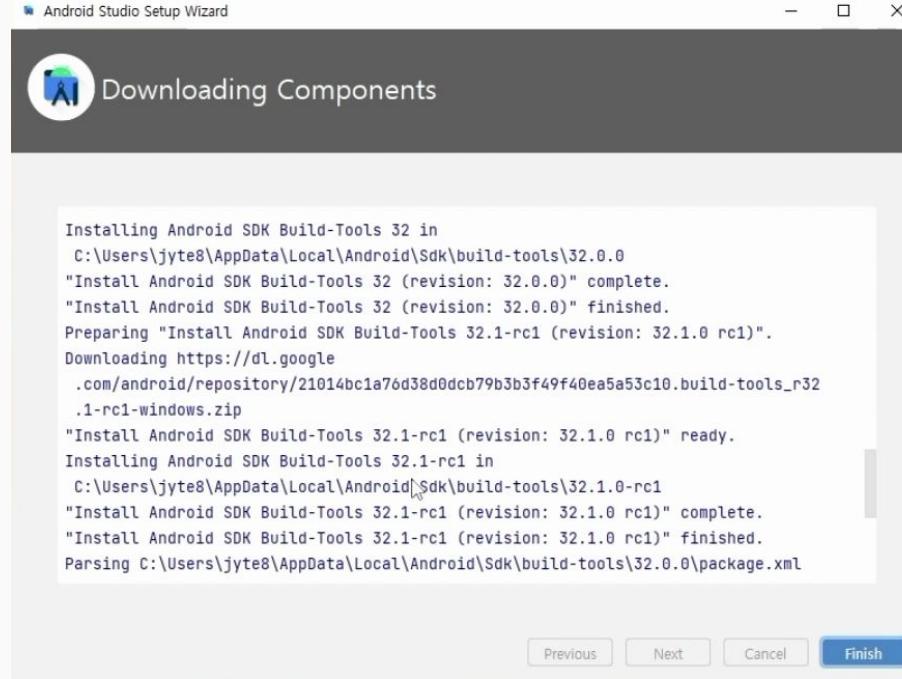
# 안드로이드 스튜디오 초기 설정

설치가 끝날때까지 잠시 기다려줍니다.



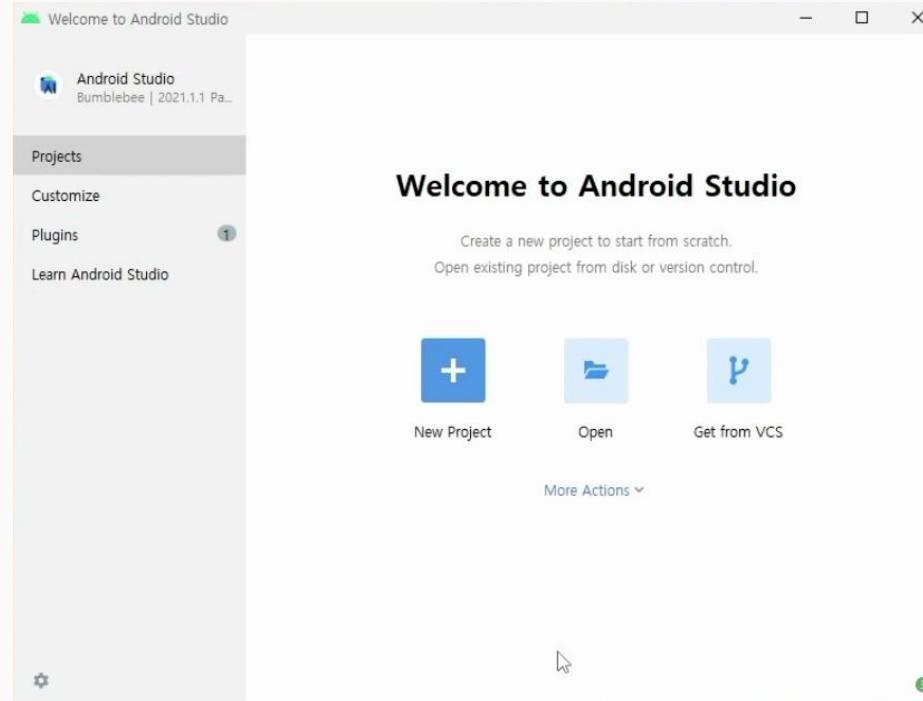
# 안드로이드 스튜디오 초기 설정

설치가 모두 끝나면, **Finish**를 눌러 설치를 마무리합니다.



# 안드로이드 스튜디오 초기 설정

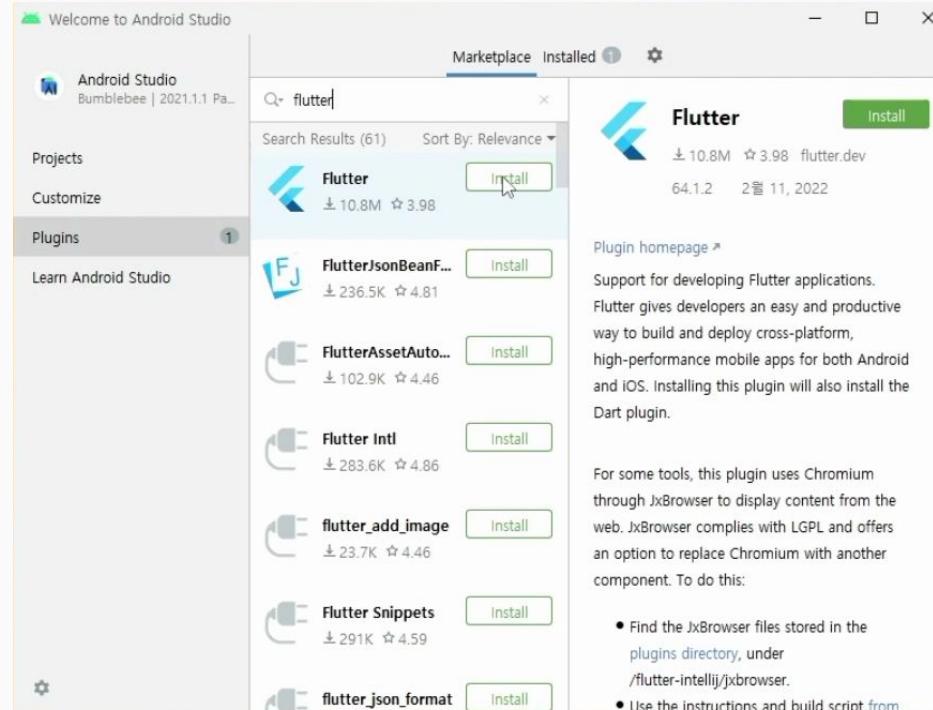
안드로이드 스튜디오 초기 화면이 실행됩니다.



# Flutter 플러그인 추가하기

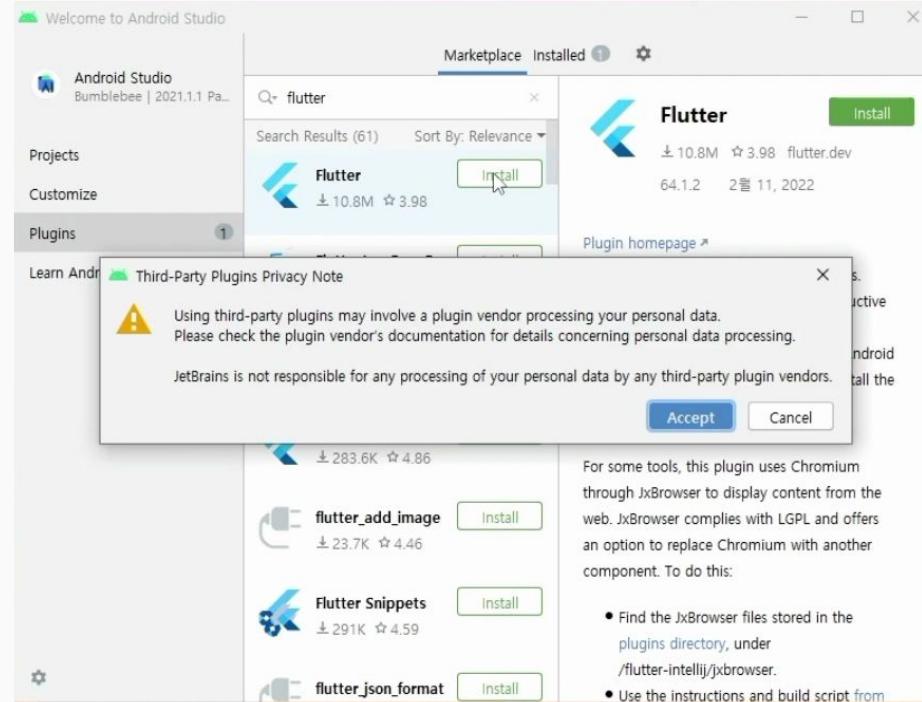
# Flutter 플러그인 추가하기

안드로이드 스튜디오 초기 화면 왼쪽의 **Plugins** 탭을 선택한 후, **Flutter** 플러그인을 검색합니다.



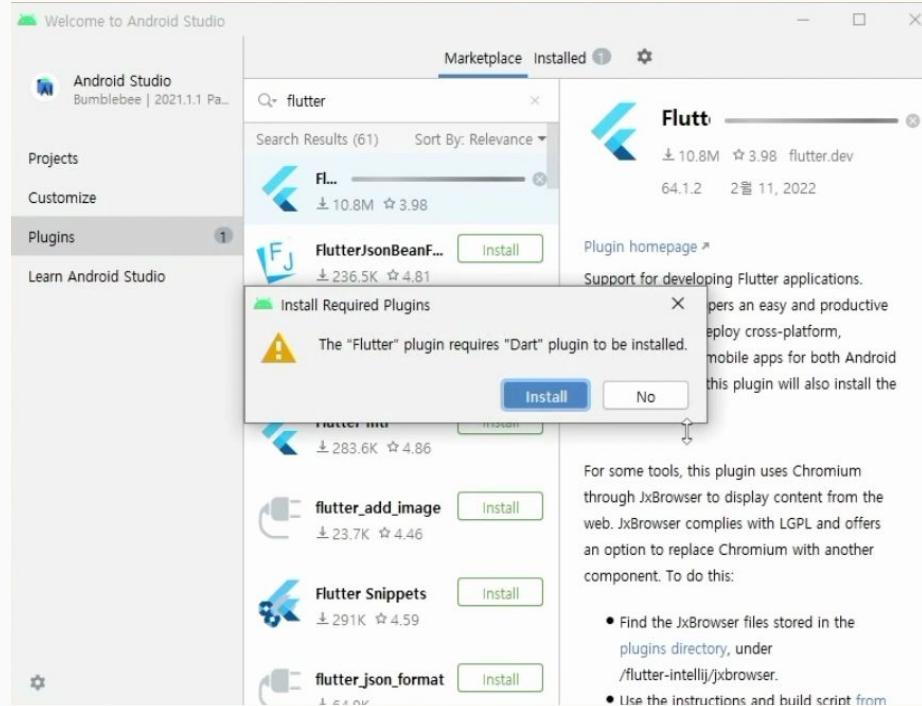
# Flutter 플러그인 추가하기

서드파티 플러그인 관련 경고가 나오면 **Accept**를 눌러 계속 진행합니다.



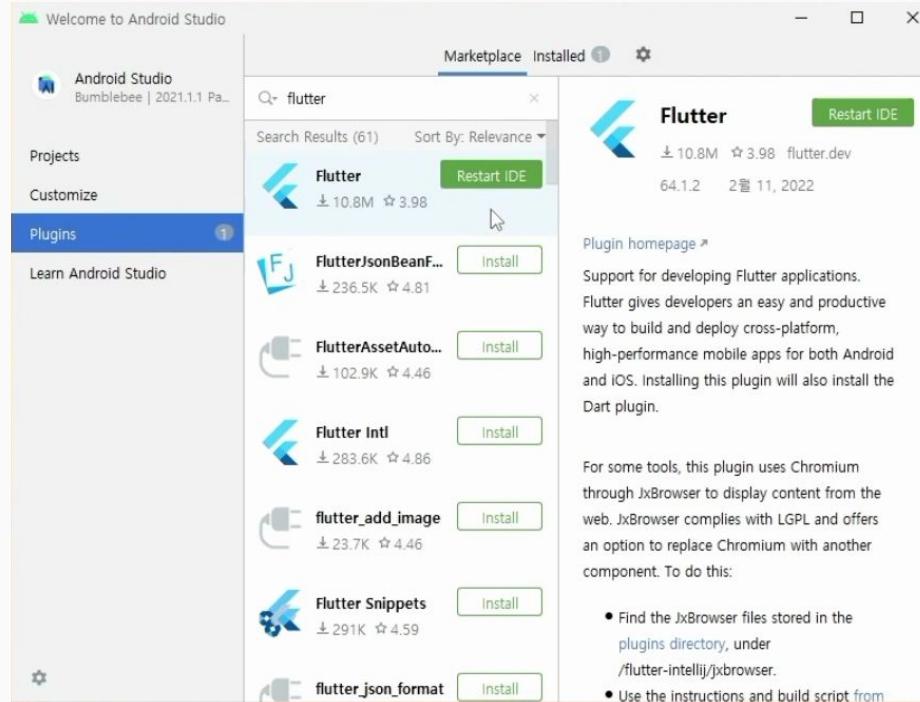
# Flutter 플러그인 추가하기

Flutter 플러그인을 사용하려면 Dart 플러그인도 함께 설치해야 합니다. **Install** 버튼을 눌러 설치합니다.



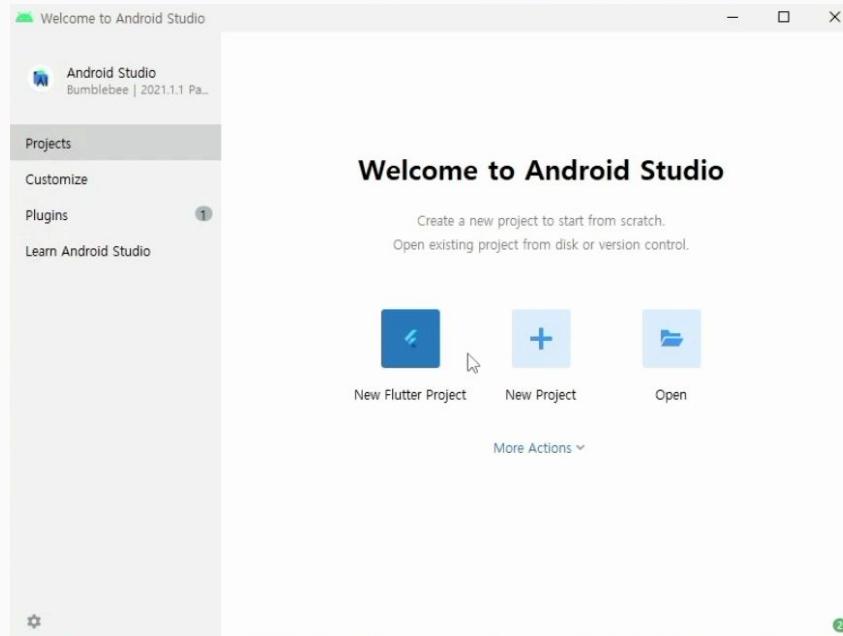
# Flutter 플러그인 추가하기

플러그인 설치가 완료되면, **Restart IDE** 버튼을 눌러 안드로이드 스튜디오를 재시작합니다.



# Flutter 플러그인 추가하기

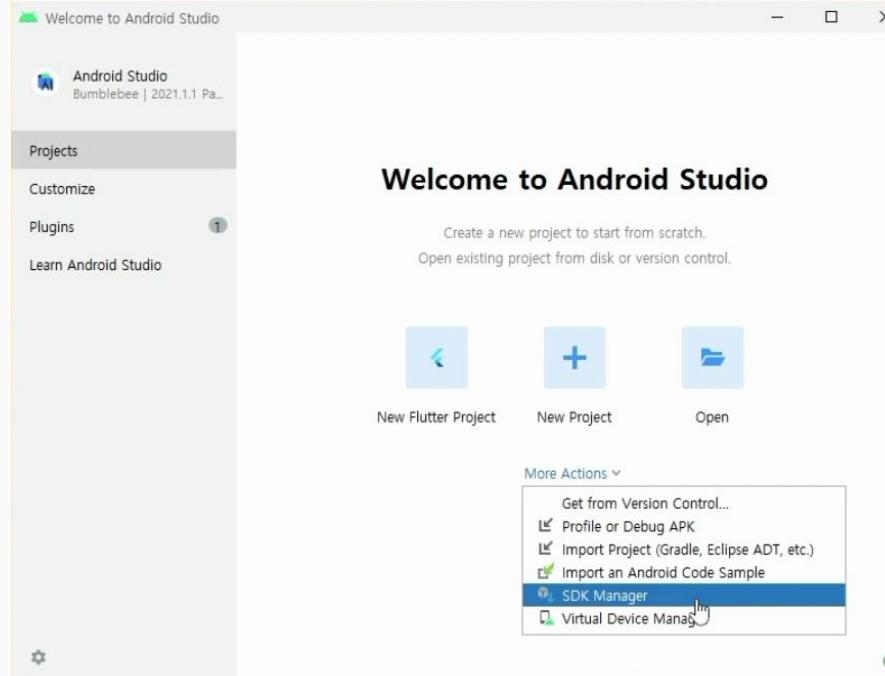
안드로이드 스튜디오 초기 화면에 **New Flutter Project** 버튼이 추가됩니다.



# 안드로이드 SDK 커맨드라인 도구 설치하기

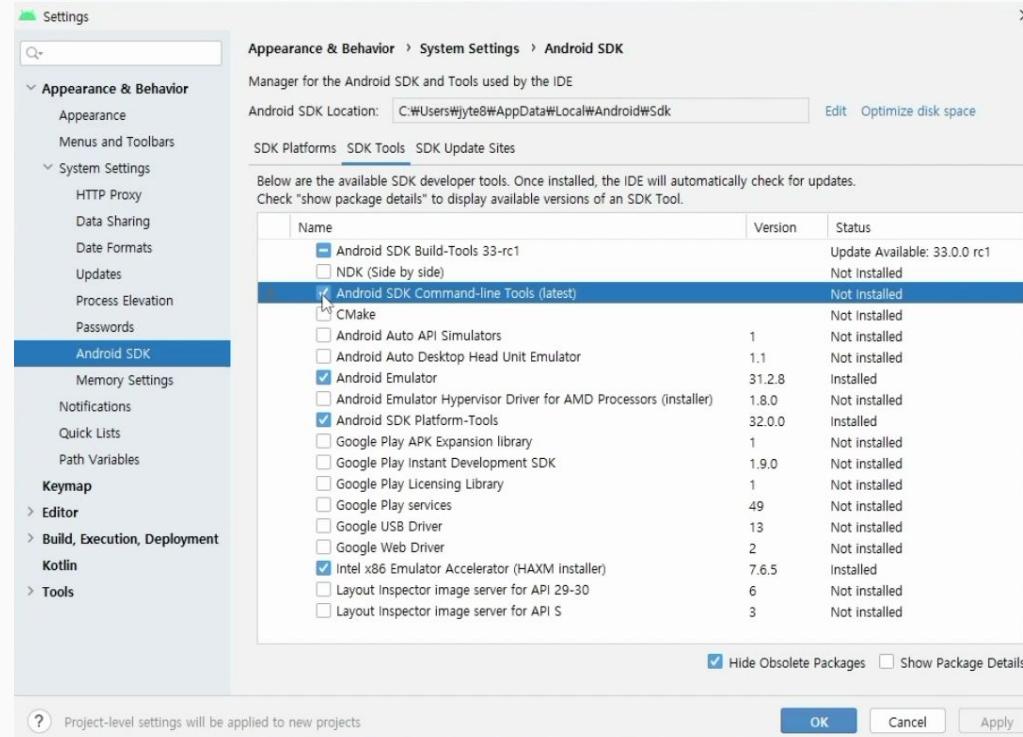
# 안드로이드 SDK 커맨드라인 도구 설치

안드로이드 스튜디오 초기 화면에서 **More Actions > SDK Manager**를 선택합니다.



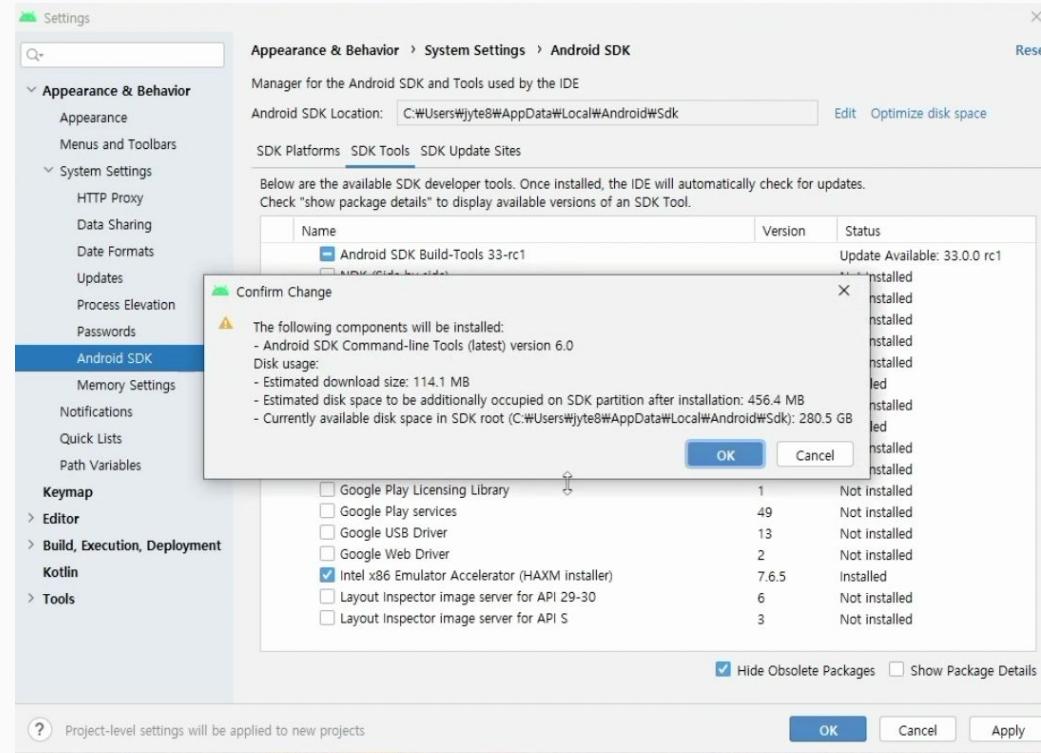
# 안드로이드 SDK 커맨드라인 도구 설치

SDK Tools 탭에서 **Android SDK Command-line Tools**를 선택한 후, **Apply** 버튼을 누릅니다.



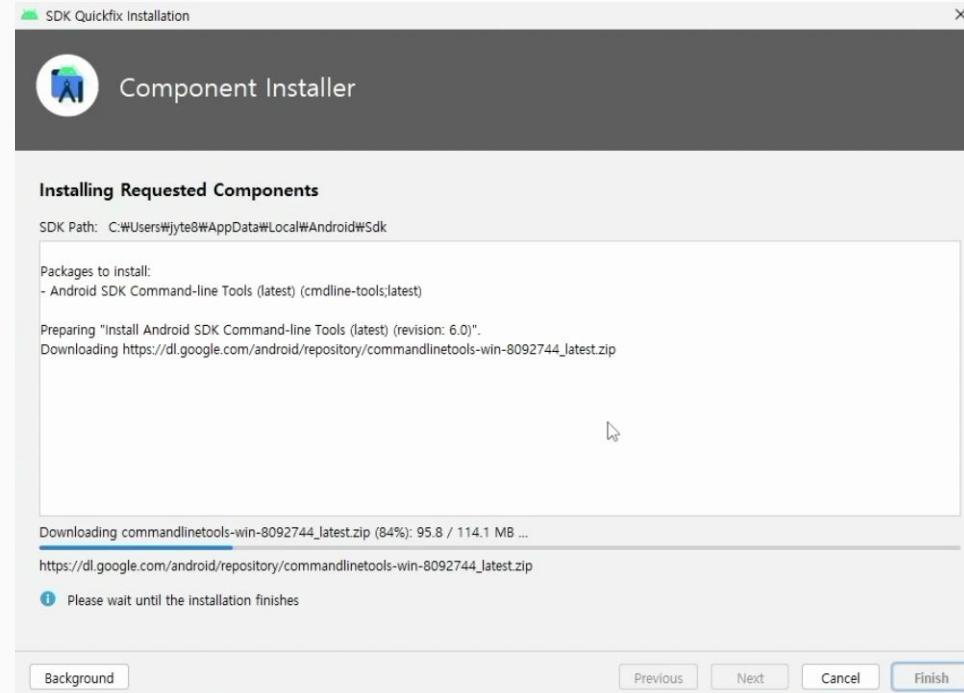
# 안드로이드 SDK 커맨드라인 도구 설치

설치할 항목을 확인하는ダイ얼로그가 표시되면, **OK** 버튼을 눌러 설치를 계속합니다.



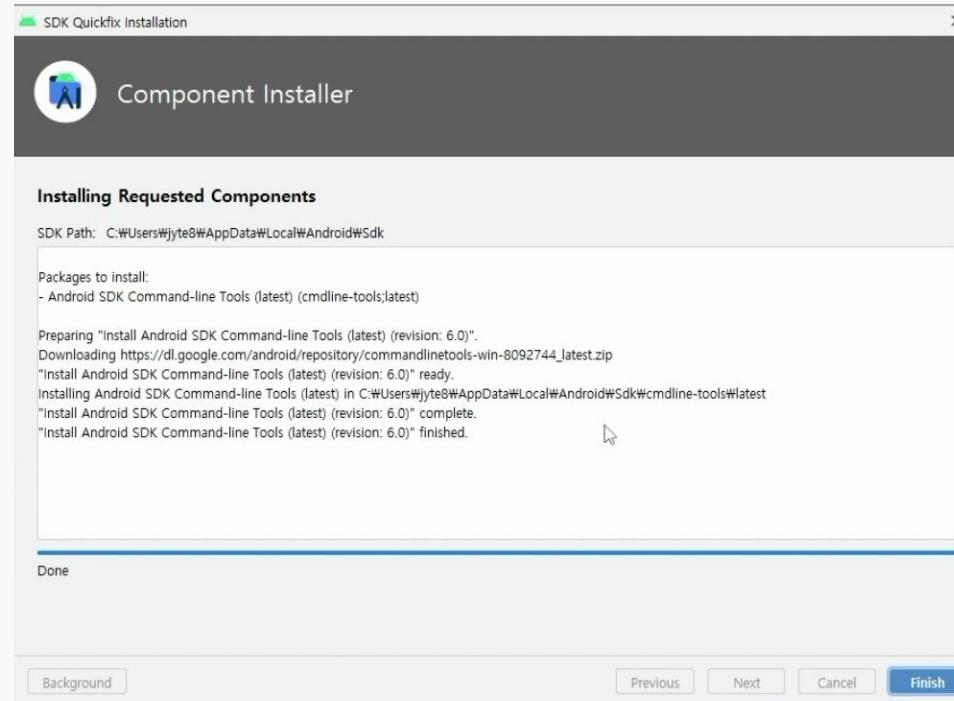
# 안드로이드 SDK 커맨드라인 도구 설치

설치가 끝날때까지 잠시 기다려줍니다.



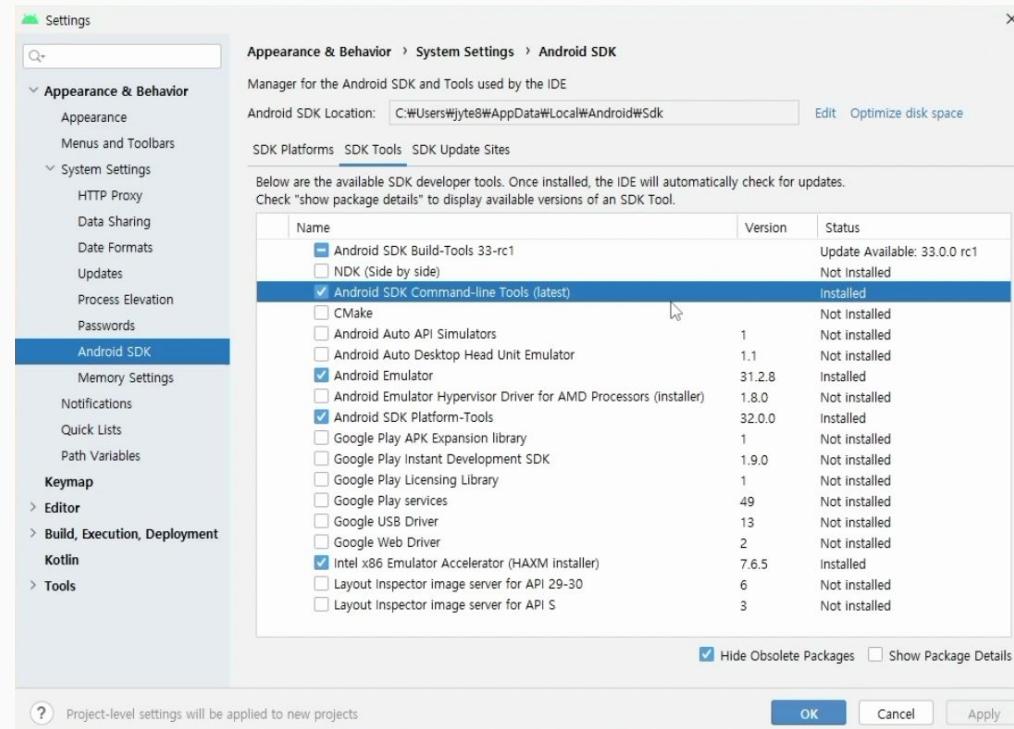
# 안드로이드 SDK 커맨드라인 도구 설치

Finish를 눌러 설치를 종료합니다.



# 안드로이드 SDK 커맨드라인 도구 설치

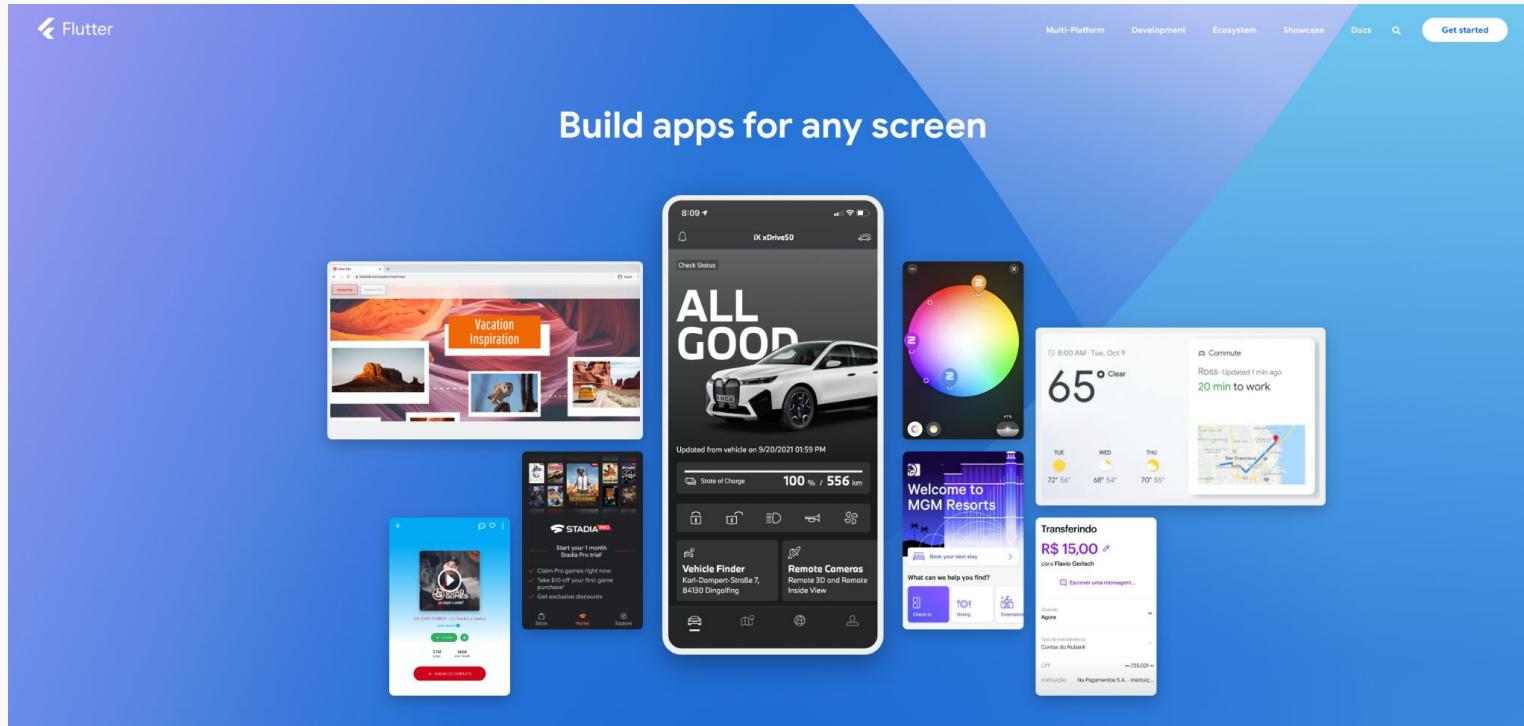
설치가 잘 되었다면, Status가 **Installed**로 바뀝니다.



# Flutter SDK 설치하기

# Flutter SDK 설치

[flutter.dev](#) 페이지로 이동 후, **Get started**를 누릅니다.



# Flutter SDK 설치 (Windows)

Windows 를 선택합니다.

Set up an editor ›

## Install

Get started > Install

Select the operating system on which you are installing Flutter:



Windows



macOS



Linux



Chrome OS

⚠ Important: If you're in China, first read Using Flutter in China.

Set up an editor ›

# Flutter SDK 설치 (Windows)

Get the Flutter SDK 섹션에서 [SDK 다운로드](#) 버튼을 누릅니다.

## Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter\\_windows\\_3.10.5-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

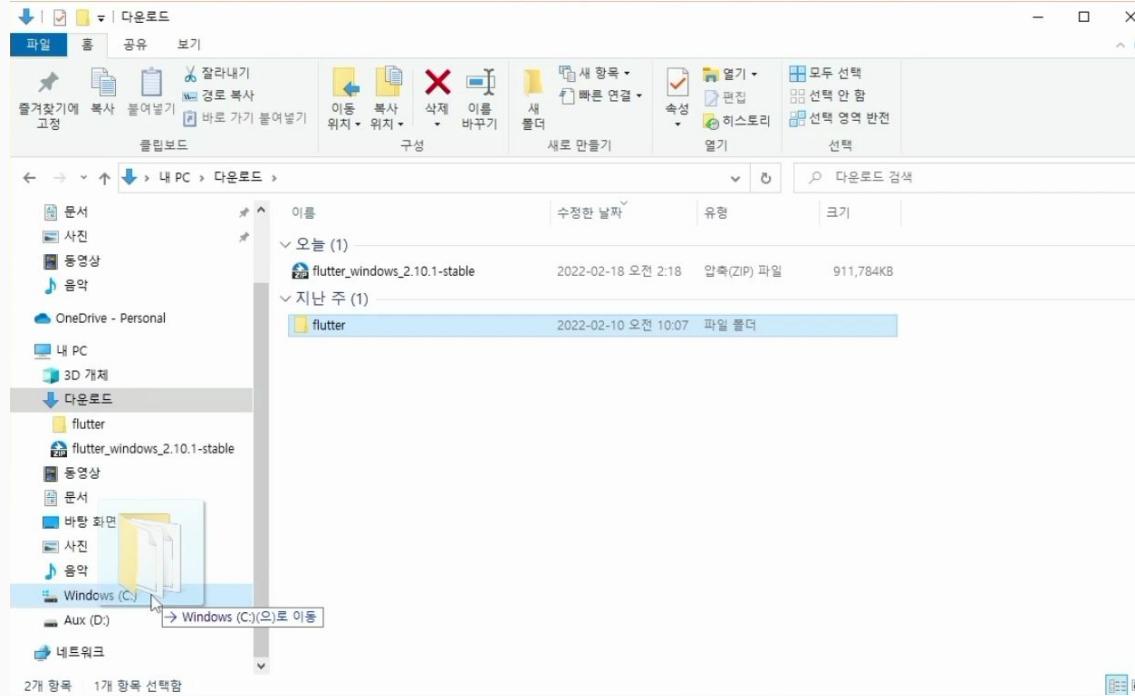
2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

**⚠ Warning:** Do not install Flutter to a path that contains special characters or spaces.

**⚠ Warning:** Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

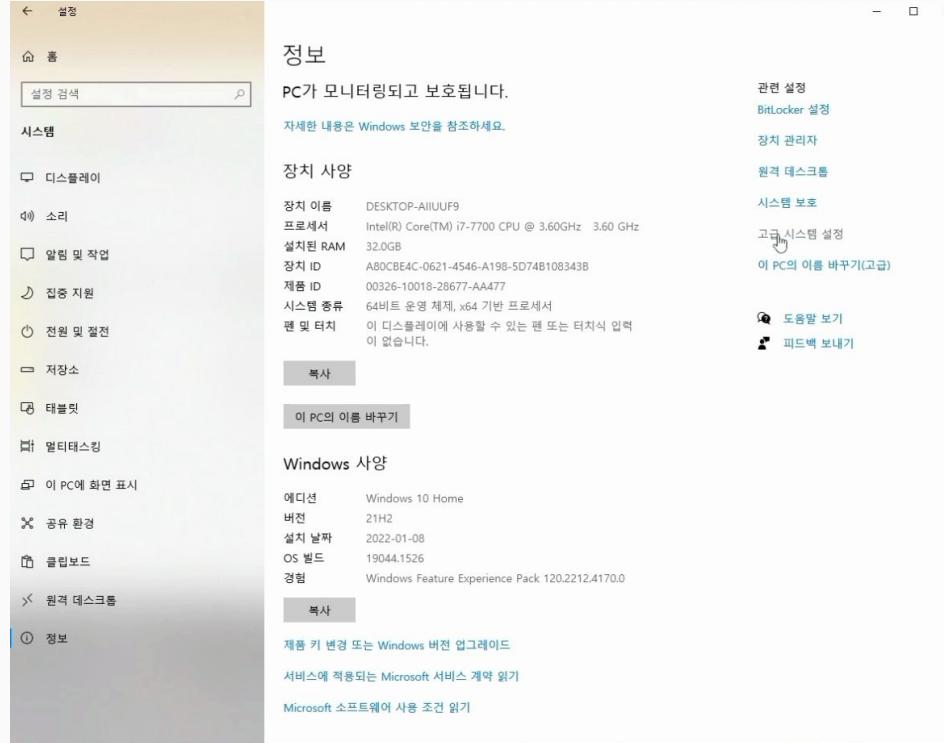
# Flutter SDK 설치 (Windows)

다운로드한 SDK를 압축 해제한 후, flutter 폴더를 [C: 드라이브 루트](#)로 이동합니다.



# Flutter SDK 설치 (Windows)

설정 > 정보 > 고급 시스템 설정을 선택합니다.



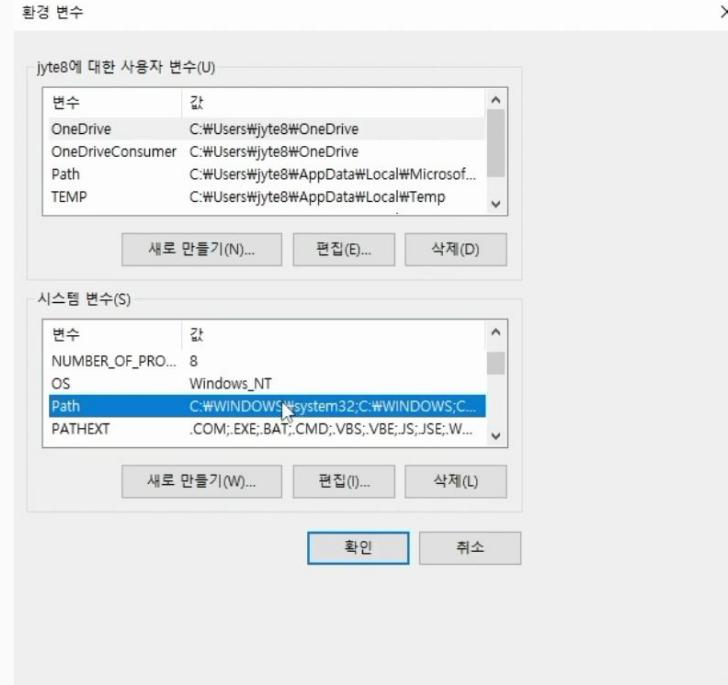
# Flutter SDK 설치 (Windows)

시스템 속성 창에서 **환경 변수**를 선택합니다.



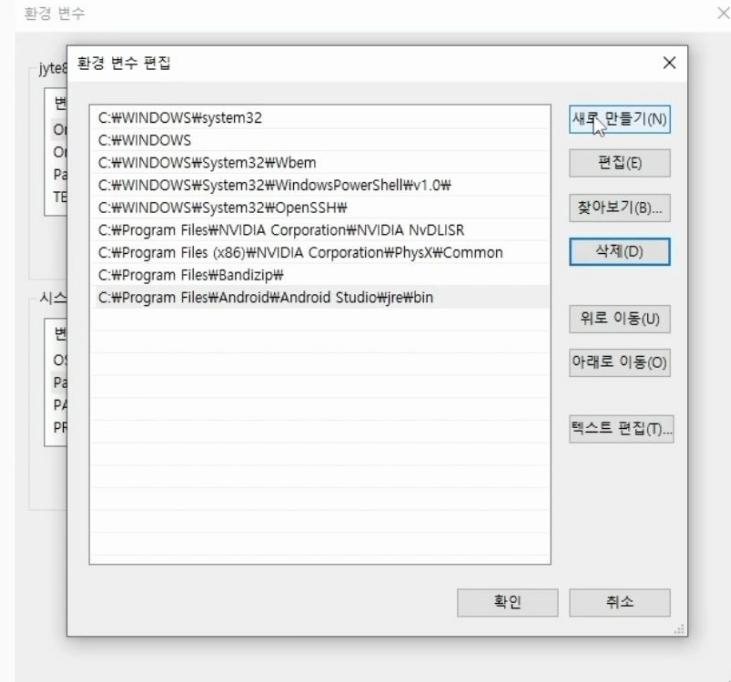
# Flutter SDK 설치 (Windows)

시스템 변수에서 **Path**를 선택한 후, **편집** 버튼을 선택합니다.



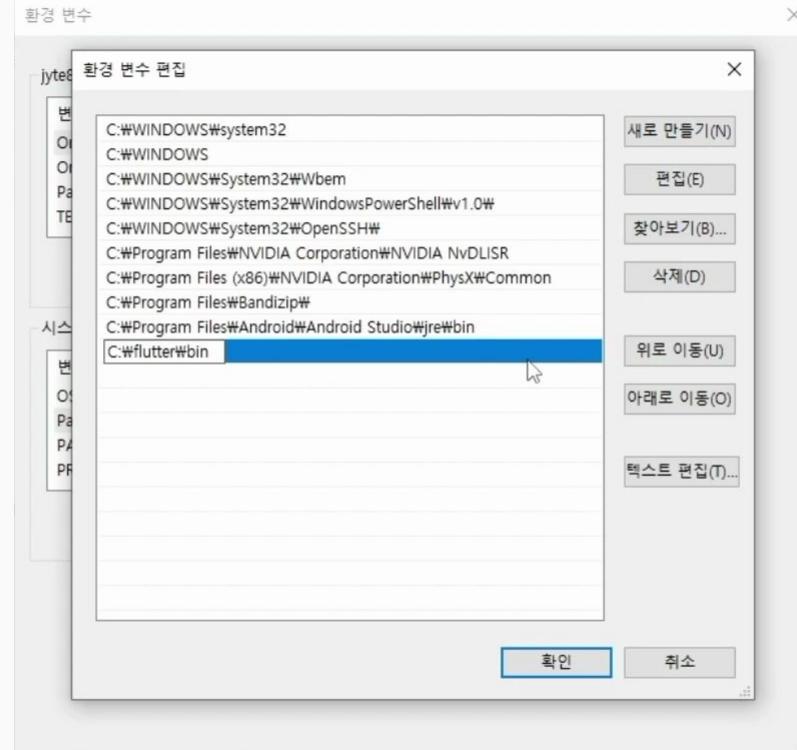
# Flutter SDK 설치 (Windows)

환경 변수 편집 창에서 **새로 만들기** 버튼을 선택합니다.



# Flutter SDK 설치 (Windows)

C:\flutter\bin 을 입력한 후, 확인 버튼을 누릅니다.



# Flutter SDK 설치 (Windows)

환경 변수 창에서 **확인** 버튼을 눌러 변경사항을 저장합니다.



# Git 설치 (Windows만)



# git --distributed-even-if-your-workflow-isnt

Search entire site...

[About](#)

[Documentation](#)

[Downloads](#)

- GUI Clients
- Logos

[Community](#)

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Download for Windows

[Click here to download](#) the latest (2.37.1) 32-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **12 days ago**, on 2022-07-12.

### Other Git for Windows downloads

[Standalone Installer](#)

[32-bit Git for Windows Setup.](#)

**64-bit Git for Windows Setup.**

[Portable \("thumbdrive edition"\)](#)

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

### Using winget tool

Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version 2.37.1. If you want the newer version, you can build it from [the source code](#).

[git-scm.com/download/windows](https://git-scm.com/download/windows)

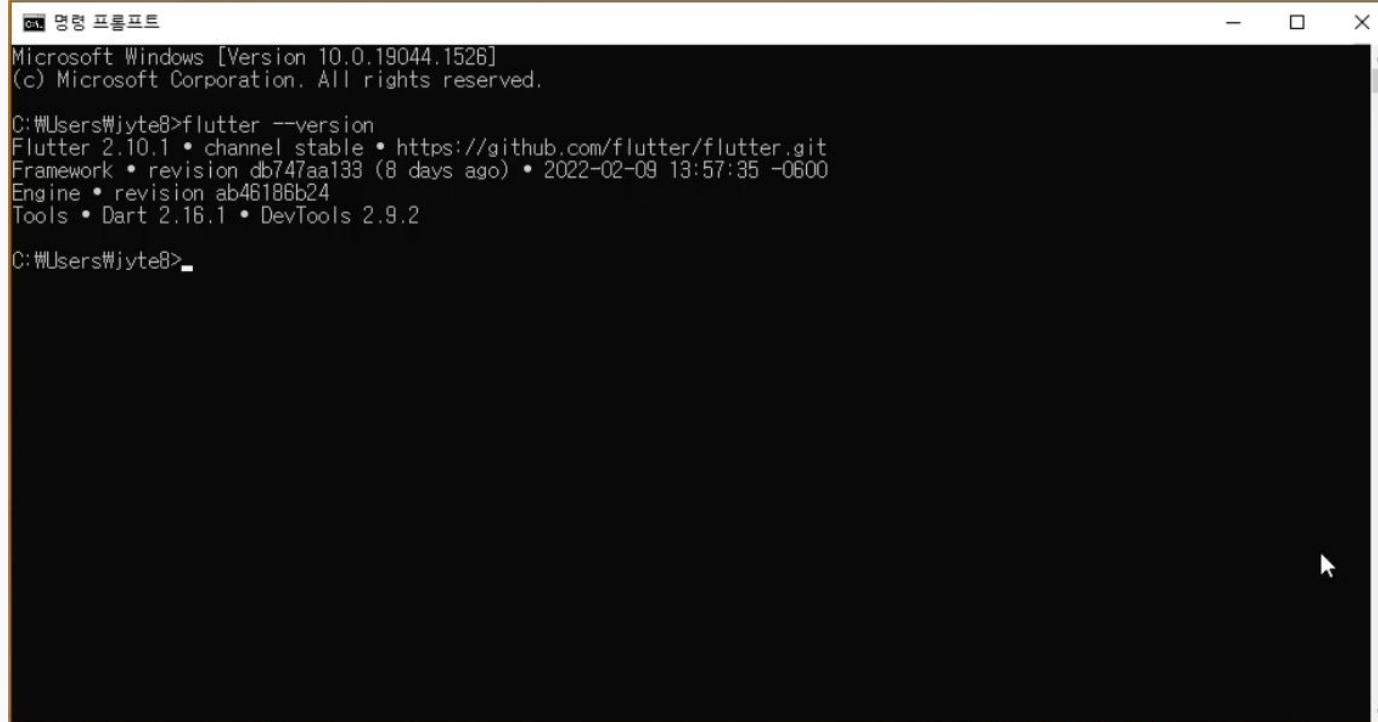
# Git 설치 (Windows만)

설치 마법사를 실행한 후, **기본 옵션 그대로** 설치를 완료합니다.



# Flutter SDK 구성 완료하기 (Windows)

명령 프롬프트를 실행한 후, **flutter --version** 명령을 실행한 후, SDK 버전이 출력되는지 확인합니다.



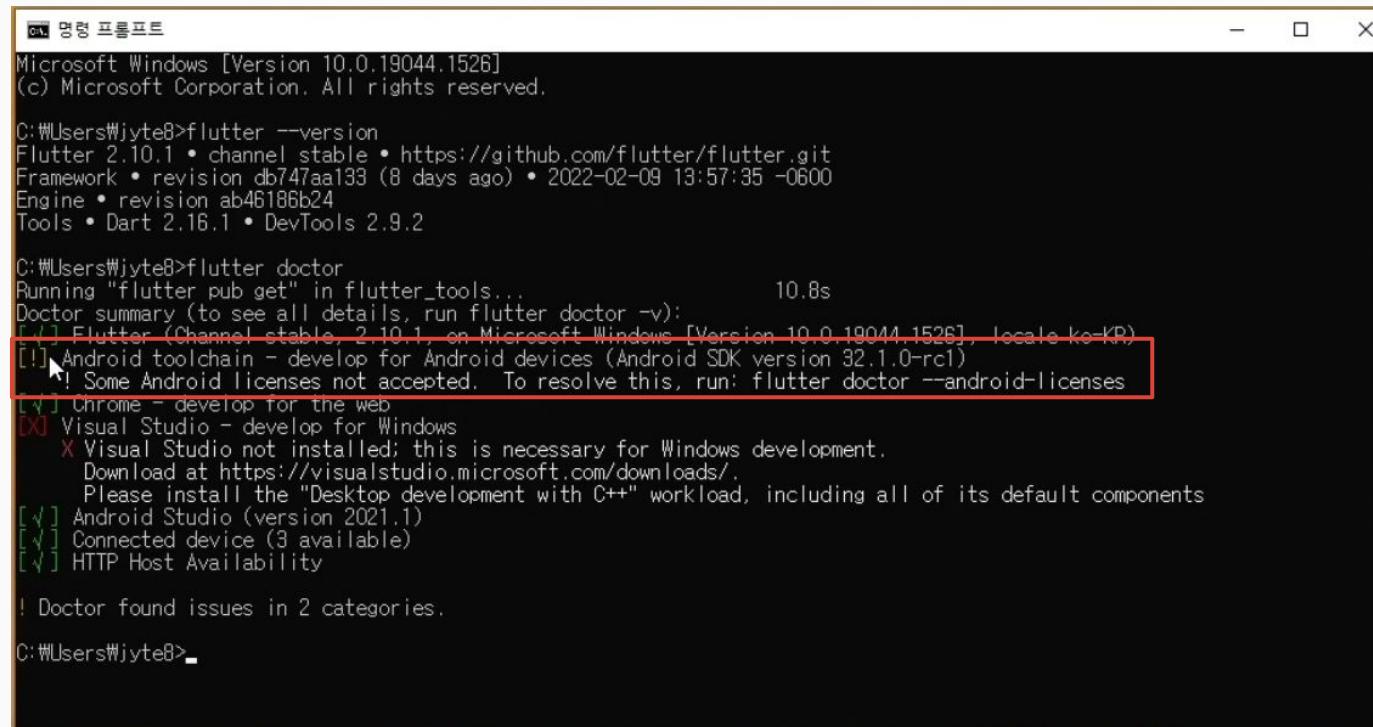
```
명령 프롬프트
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jyte8>flutter --version
Flutter 2.10.1 • channel stable • https://github.com/flutter/flutter.git
Framework • revision db747aa133 (8 days ago) • 2022-02-09 13:57:35 -0600
Engine • revision ab46186b24
Tools • Dart 2.16.1 • DevTools 2.9.2

C:\Users\jyte8>
```

# Flutter SDK 구성 완료하기 (Windows)

**flutter doctor** 명령을 실행하여 SDK 구성 상태를 확인합니다.



```
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jyte8>flutter --version
Flutter 2.10.1 • channel stable • https://github.com/flutter/flutter.git
Framework • revision db747aa133 (8 days ago) • 2022-02-09 13:57:35 -0600
Engine • revision ab46186b24
Tools • Dart 2.16.1 • DevTools 2.9.2

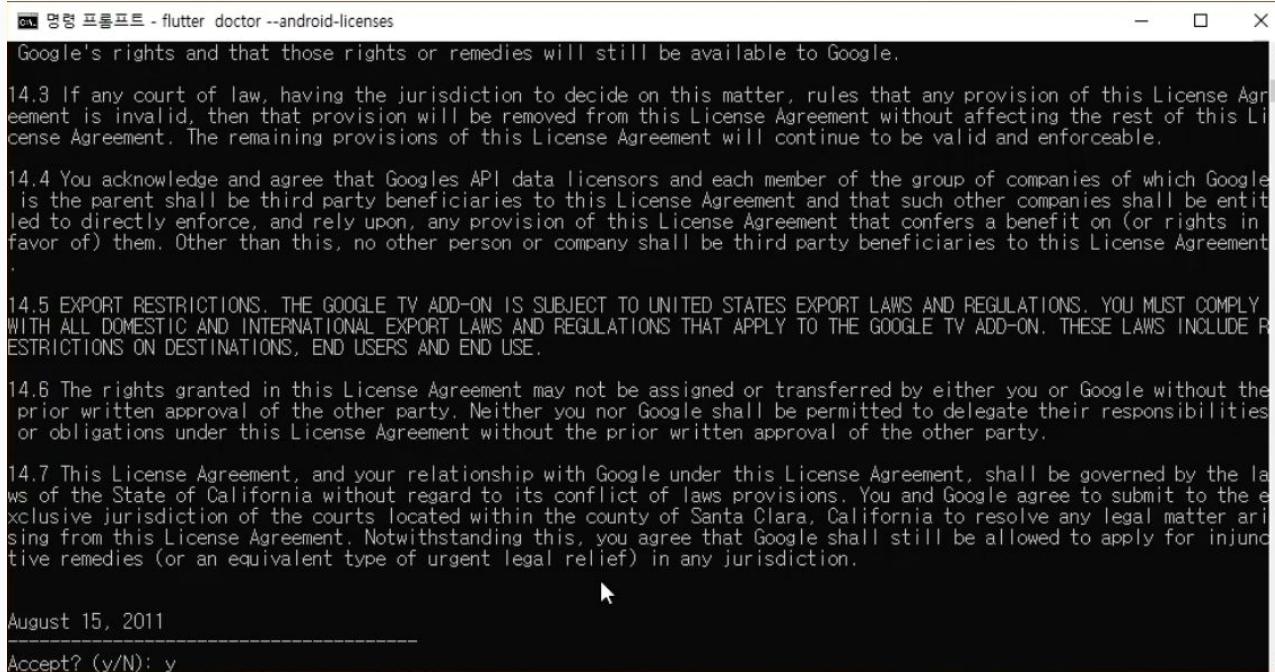
C:\Users\jyte8>flutter doctor
Running "flutter pub get" in flutter_tools...                                10.8s
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.10.1, on Microsoft Windows [Version 10.0.19044.1526], locale ko-KR)
[!] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[✗] Visual Studio - develop for Windows
    X Visual Studio not installed; this is necessary for Windows development.
        Download at https://visualstudio.microsoft.com/downloads/.
        Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2021.1)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

! Doctor found issues in 2 categories.

C:\Users\jyte8>
```

# Flutter SDK 구성 완료하기 (Windows)

**flutter doctor --android-licenses** 명령을 실행하여 안드로이드 SDK 라이선스를 수락합니다.



# Flutter SDK 구성 완료하기 (Windows)

라이선스 수락이 끝나면 Android toolchain 항목 에러가 사라집니다.

```
C:\ 명령 프롬프트
ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED AS IS AND AS AVAILABLE, POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRANTY OF ANY KIND.

10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software license agreement.

Accept? (y/N): y
All SDK package licenses accepted

C:\Users\jyte8>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.10.1, on Microsoft Windows [Version 10.0-19044-1526], locale ko-KR)
[✗] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
    ✗ Visual Studio - develop for Windows
        X Visual Studio not installed; this is necessary for Windows development.
        Download at https://visualstudio.microsoft.com/downloads/.
        Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2021.1)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

! Doctor found issues in 1 category.

C:\Users\jyte8>
```

# Flutter SDK 설치 (macOS)

macOS 를 선택합니다.

Set up an editor ›

## Install

Get started > Install

Select the operating system on which you are installing Flutter:



Windows



macOS



Linux



Chrome OS

**Important:** If you're in China, first read Using Flutter in China.

Set up an editor ›

# Flutter SDK 설치 (macOS)

Get the Flutter SDK 섹션에서 [사용중인 컴퓨터](#)에 맞는 [SDK를 다운로드](#) 합니다.

## Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

Intel

Apple Silicon

[flutter\\_macos\\_3.10.5-stable.zip](#)

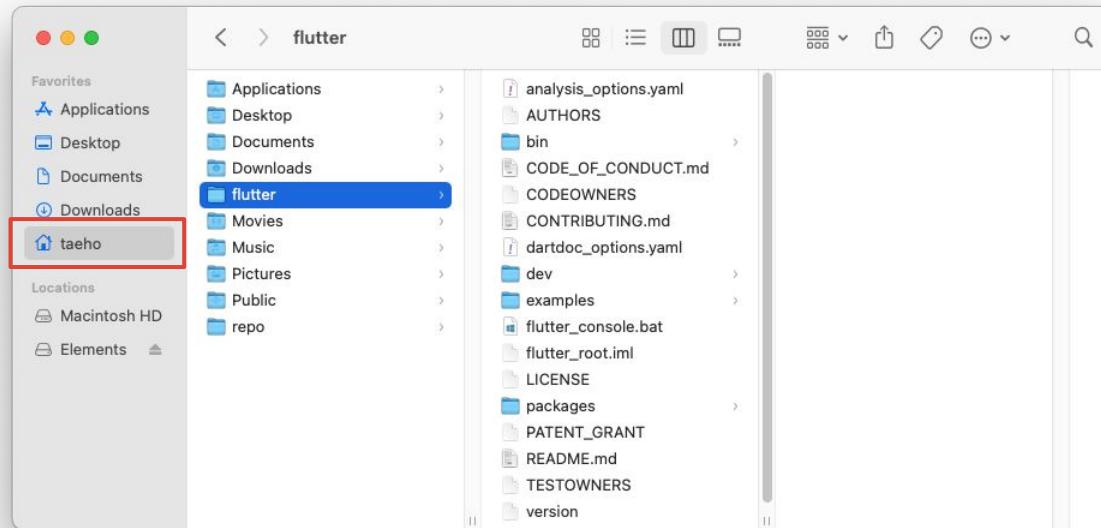
[flutter\\_macos\\_arm64\\_3.10.5-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

💡 Tip: To determine whether your Mac uses an Apple silicon processor, refer to [Mac computers with Apple silicon on apple.com](#)

# Flutter SDK 설치 (macOS)

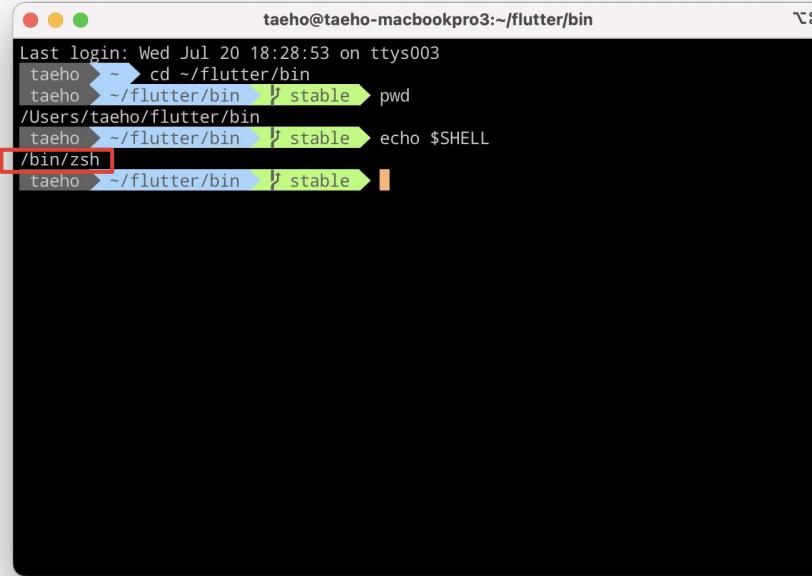
다운로드한 SDK를 압축 해제한 후, flutter 폴더를 [홈 폴더](#)로 이동합니다.



# Flutter SDK 설치 (macOS)

터미널을 실행한 후, `echo $SHELL` 명령을 입력하여 사용중인 셸 종류를 확인합니다.

- /bin/bash: Bash shell 사용중
- /bin/zsh: Z shell 사용중



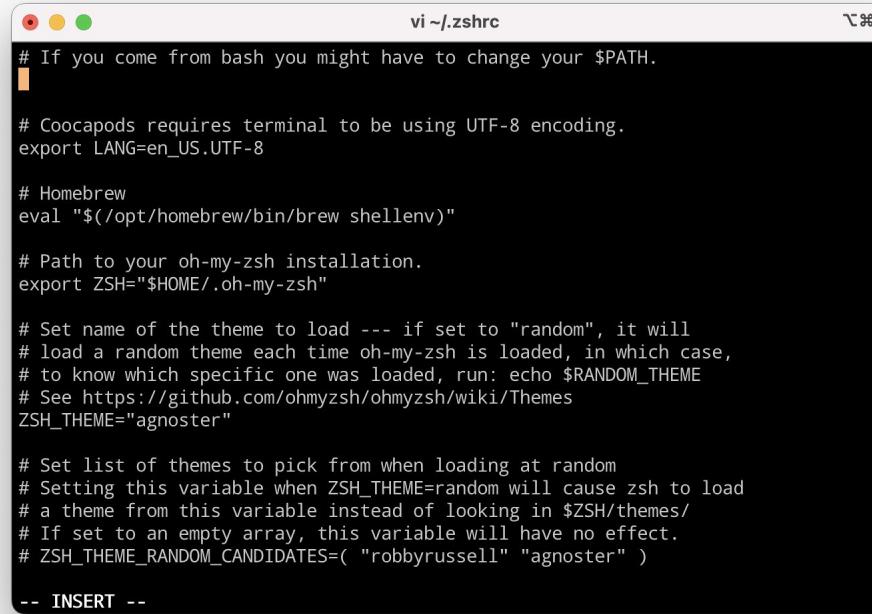
```
taeho@taeho-macbookpro3:~/flutter/bin
Last login: Wed Jul 20 18:28:53 on ttys003
taeho ~ cd ~/flutter/bin
taeho ~/flutter/bin ↵ stable ↵ pwd
/Users/taeho/flutter/bin
taeho ~/flutter/bin ↵ stable ↵ echo $SHELL
/bin/zsh
taeho ~/flutter/bin ↵ stable ↵
```

The screenshot shows a macOS terminal window titled "taeho@taeho-macbookpro3:~/flutter/bin". The user has run the command `echo $SHELL`, which returns the value `/bin/zsh`. The line containing `/bin/zsh` is highlighted with a red box.

# Flutter SDK 설치 (macOS)

셀 종류에 따라 다음 명령어를 입력합니다.

- Bash shell: `vi ~/.bash_profile` 입력 후, **i**를 눌러 편집 모드로 전환
- Z shell: `vi ~/.zshrc` 입력 후, **i**를 눌러 편집 모드로 전환



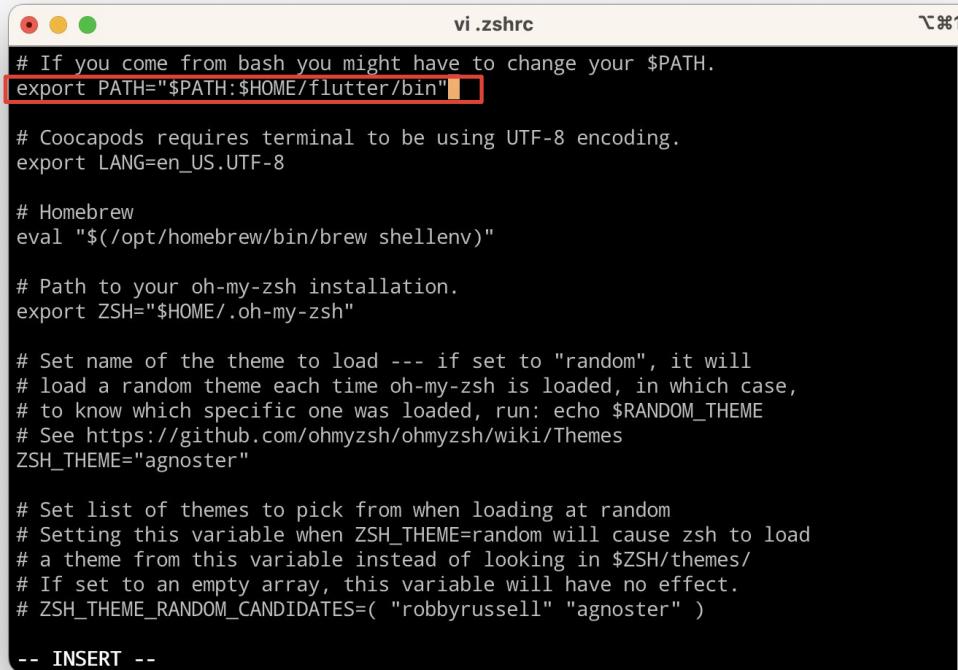
The screenshot shows a terminal window titled "vi ~/.zshrc". The window has a dark background and contains the following text:

```
# If you come from bash you might have to change your $PATH.  
#  
# Coocapods requires terminal to be using UTF-8 encoding.  
export LANG=en_US.UTF-8  
  
# Homebrew  
eval "$( /opt/homebrew/bin/brew shellenv )"  
  
# Path to your oh-my-zsh installation.  
export ZSH="$HOME/.oh-my-zsh"  
  
# Set name of the theme to load --- if set to "random", it will  
# load a random theme each time oh-my-zsh is loaded, in which case,  
# to know which specific one was loaded, run: echo $RANDOM_THEME  
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes  
ZSH_THEME="agnoster"  
  
# Set list of themes to pick from when loading at random  
# Setting this variable when ZSH_THEME=random will cause zsh to load  
# a theme from this variable instead of looking in $ZSH/themes/  
# If set to an empty array, this variable will have no effect.  
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )  
  
-- INSERT --
```

# Flutter SDK 설치 (macOS)

PATH에 Flutter SDK가 설치된 폴더를 넣어줍니다.

- `export PATH="$PATH:$HOME/flutter/bin"`



```
# If you come from bash you might have to change your $PATH.
export PATH="$PATH:$HOME/flutter/bin"

# Coocapods requires terminal to be using UTF-8 encoding.
export LANG=en_US.UTF-8

# Homebrew
eval "$( /opt/homebrew/bin/brew shellenv )"

# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"

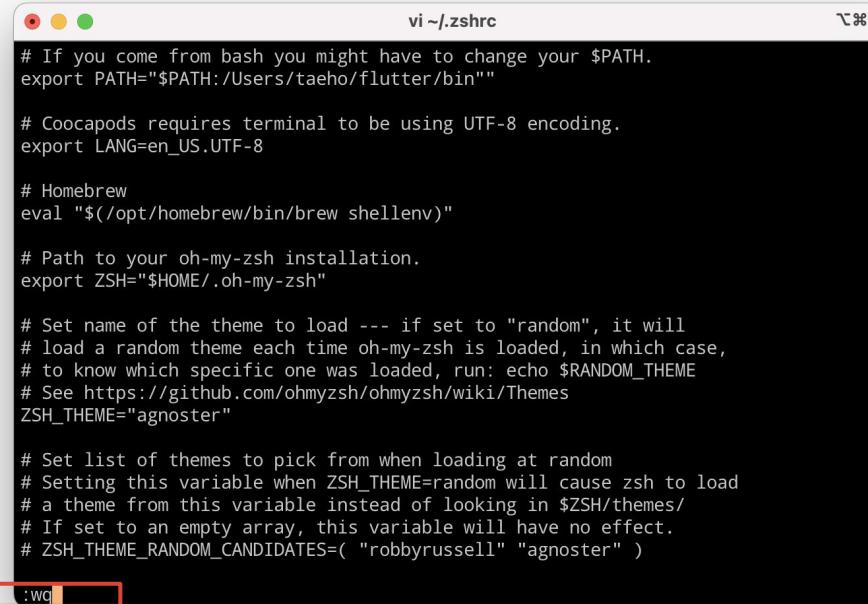
# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="agnoster"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

-- INSERT --
```

# Flutter SDK 설치 (macOS)

Esc > :(콜론) > wq > 엔터를 순서대로 입력하여 편집한 내용을 저장합니다.



```
vi ~/.zshrc
^[[1~
# If you come from bash you might have to change your $PATH.
export PATH="$PATH:/Users/taeho/flutter/bin"

# Cocoapods requires terminal to be using UTF-8 encoding.
export LANG=en_US.UTF-8

# Homebrew
eval "$( /opt/homebrew/bin/brew shellenv )"

# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="agnoster"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

:wq
```

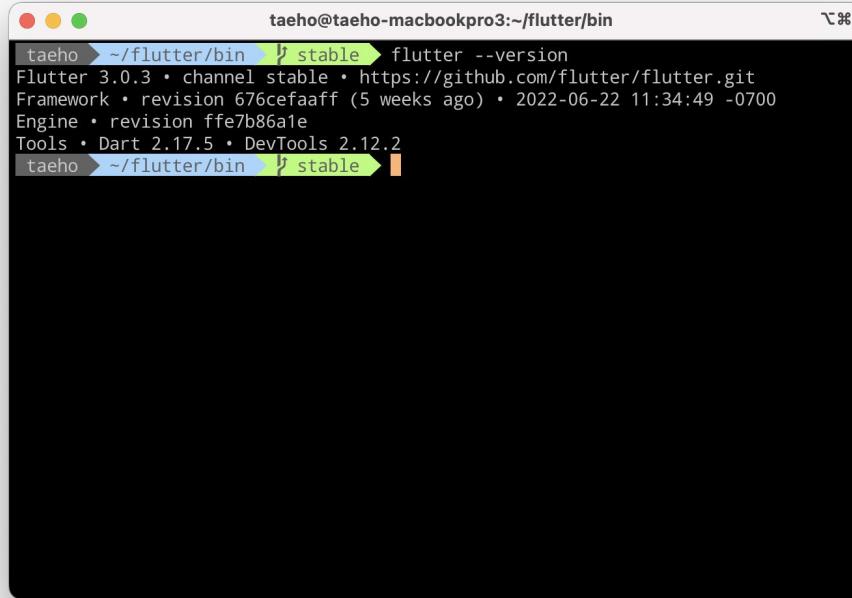
## Flutter SDK 설치 (macOS)

(Apple Silicon 맥 사용자만) Flutter SDK의 일부 도구는 Apple Silicon 용으로 작성되지 않았습니다. 이를 실행하기 위해 Rosetta를 설치해야 합니다.

**/usr/sbin/softwareupdate --install-rosetta --agree-to-license** 명령을 실행하여 로제타를 설치합니다.

# Flutter SDK 구성 완료하기 (macOS)

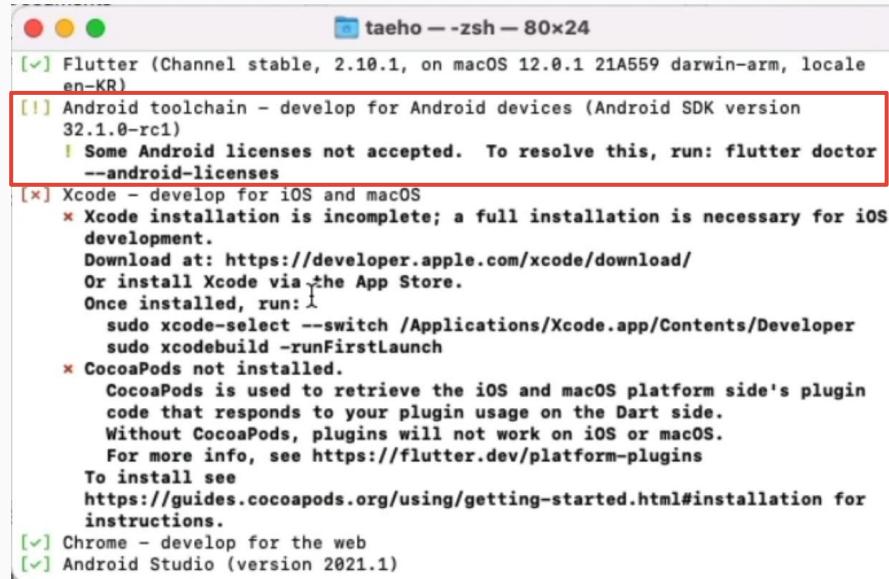
새로운 PATH 설정이 적용된 터미널을 실행하기 위해 터미널을 종료했다 다시 실행합니다. 터미널이 실행되었다면, **flutter --version** 명령을 실행하여 플러터 버전이 출력되는지 확인합니다.



```
taeho@taeho-macbookpro3:~/flutter/bin
taeho ➤ ~/flutter/bin stable ➤ flutter --version
Flutter 3.0.3 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 676cefaff (5 weeks ago) • 2022-06-22 11:34:49 -0700
Engine • revision ffe7b86a1e
Tools • Dart 2.17.5 • DevTools 2.12.2
taeho ➤ ~/flutter/bin stable ➤
```

# Flutter SDK 구성 완료하기 (macOS)

**flutter doctor** 명령을 실행하여 SDK 구성 상태를 확인합니다.



```
[✓] Flutter (Channel stable, 2.10.1, on macOS 12.0.1 21A559 darwin-arm, locale en-KR)
[!] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✗] Xcode - develop for iOS and macOS
    ✗ Xcode installation is incomplete; a full installation is necessary for iOS development.
        Download at: https://developer.apple.com/xcode/download/
        Or install Xcode via the App Store.
        Once installed, run:
            sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer
            sudo xcodebuild -runFirstLaunch
    ✗ CocoaPods not installed.
        CocoaPods is used to retrieve the iOS and macOS platform side's plugin code that responds to your plugin usage on the Dart side.
        Without CocoaPods, plugins will not work on iOS or macOS.
        For more info, see https://flutter.dev/platform-plugins
        To install see
            https://guides.cocoapods.org/using/getting-started.html#installation for instructions.
[✓] Chrome - develop for the web
[✓] Android Studio (version 2021.1)
```

# Flutter SDK 구성 완료하기 (macOS)

`flutter doctor --android-licenses` 명령을 실행하여 안드로이드 SDK 라이선스를 수락합니다.

```
taeho -- zsh -- 80x24
generally available a production version of the Pre-Release Materials . MIPS is not under any obligation to develop and/or release or offer for sale or license a final product based upon the Pre-Release Materials and may unilaterally elect to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liability whatsoever to Recipient or any other person.

ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED "AS IS" AND "AS AVAILABLE", POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRANTY OF ANY KIND.

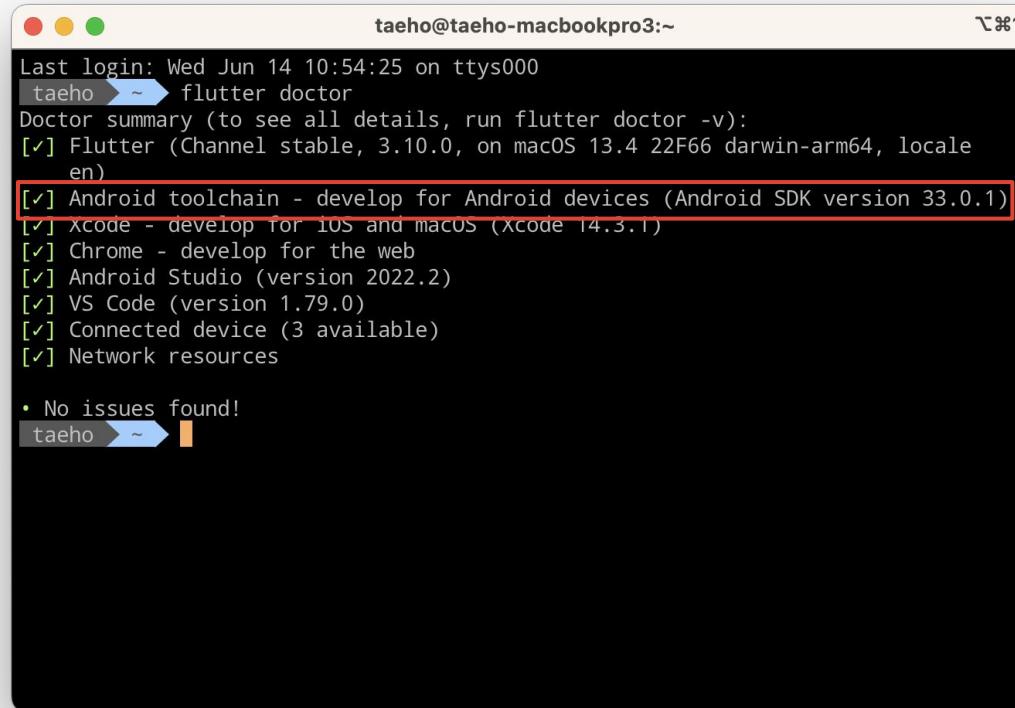
10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software license agreement.

-----
Accept? (y/N): y
All SDK package licenses accepted

taeho@taehos-mbp ~ %
```

# Flutter SDK 구성 완료하기 (macOS)

라이선스 수락이 끝나면 Android toolchain 항목 에러가 사라집니다.



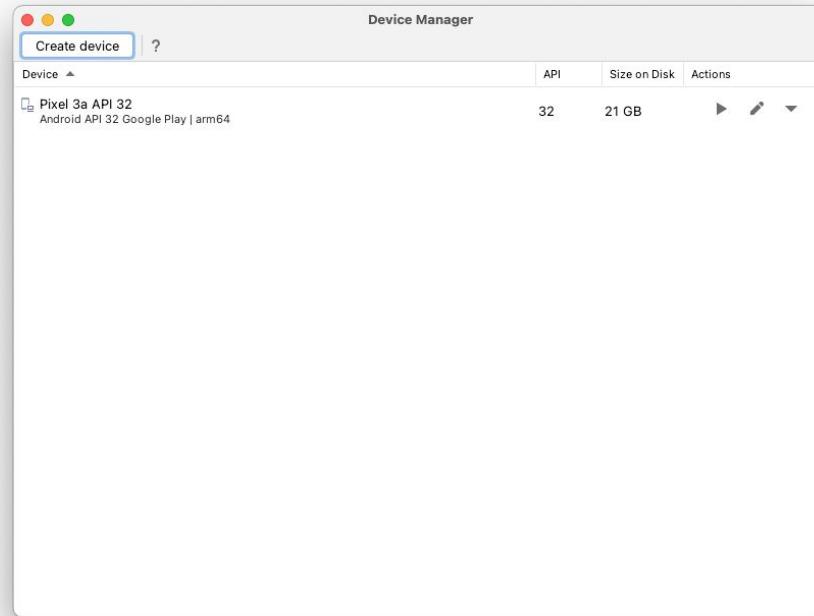
```
Last login: Wed Jun 14 10:54:25 on ttys000
taeho ➤ ~ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.0, on macOS 13.4 22F66 darwin-arm64, locale en)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.1)
[✓] Xcode - develop for iOS and macOS (Xcode 14.3.1)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.79.0)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
taeho ➤ ~
```

# 안드로이드 에뮬레이터 설정하기

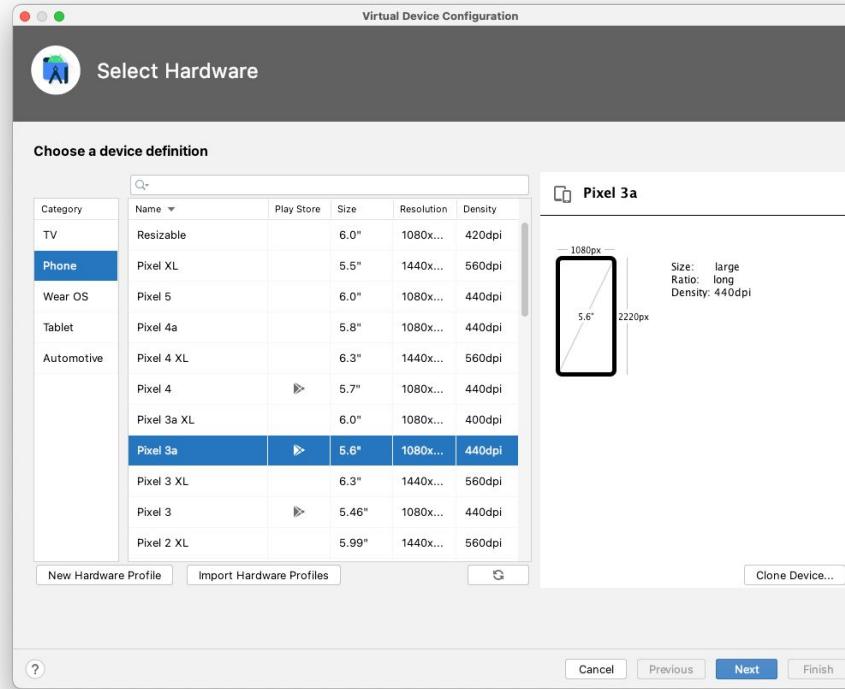
# 안드로이드 에뮬레이터 설정하기

안드로이드 스튜디오 시작 화면에서 **More actions > Virtual Device Manager**를 선택합니다. 다음, 상단의 **Create device** 버튼을 누릅니다.



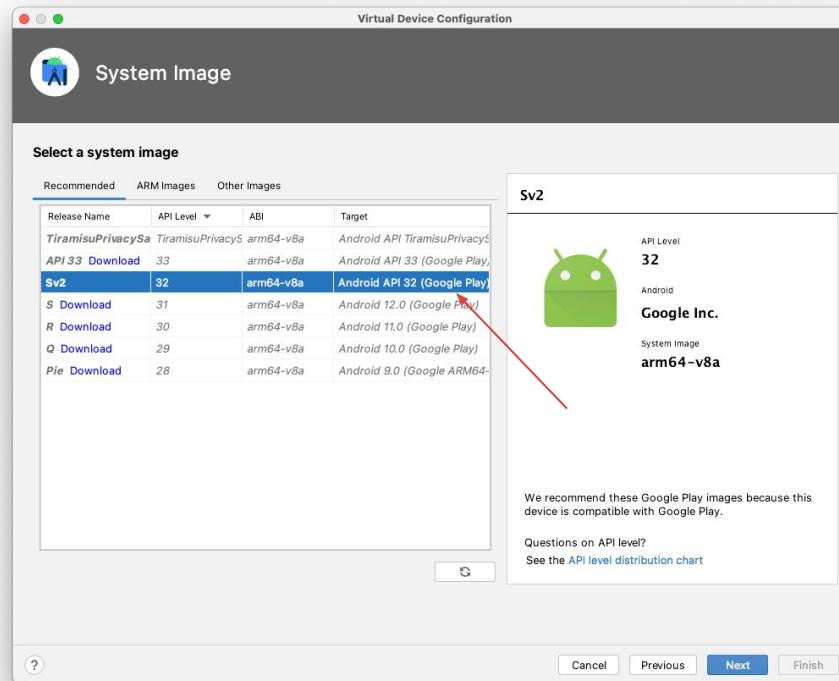
# 안드로이드 에뮬레이터 설정하기

Select Hardware 창에서 **Pixel 3a**를 선택합니다.



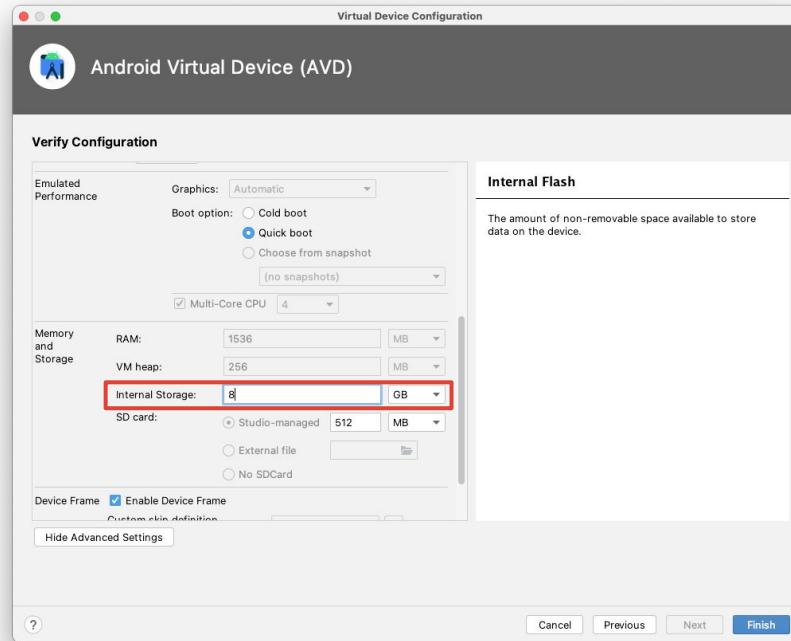
# 안드로이드 에뮬레이터 설정하기

System image 창에서 **Sv2**를 선택합니다. (Target에 Google Play가 있는지 확인합니다)



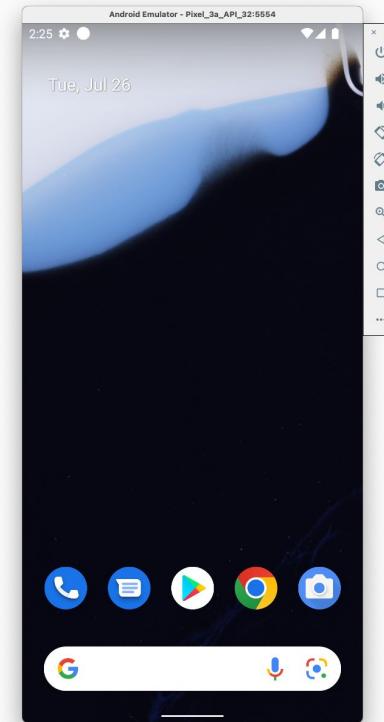
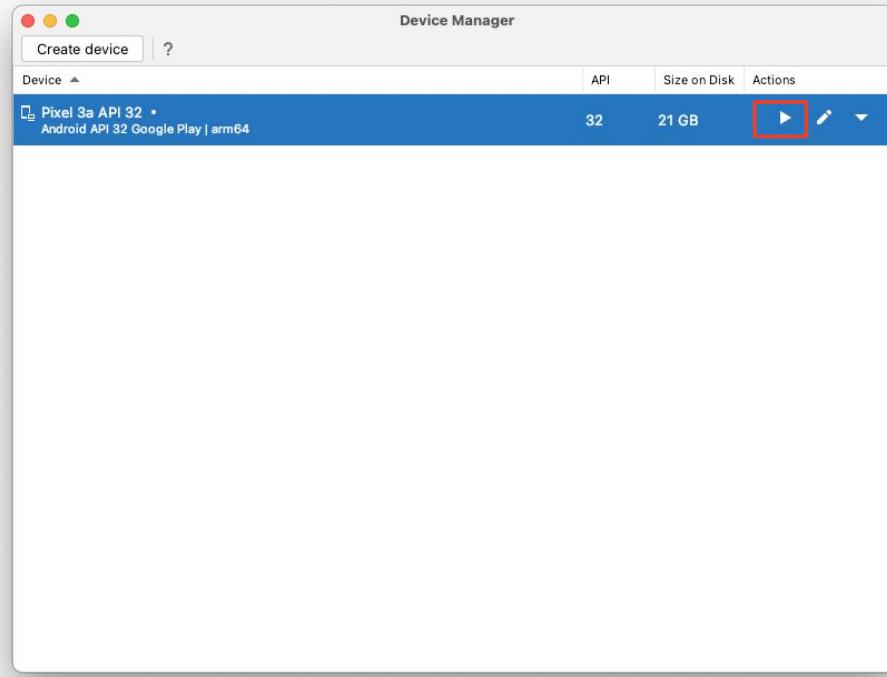
# 안드로이드 에뮬레이터 설정하기

Android Virtual Device 창에서 하단의 Show Advanced Settings 버튼을 누른 후, Internal Storage를 **8GB**로 설정하고 Finish 버튼을 눌러줍니다.



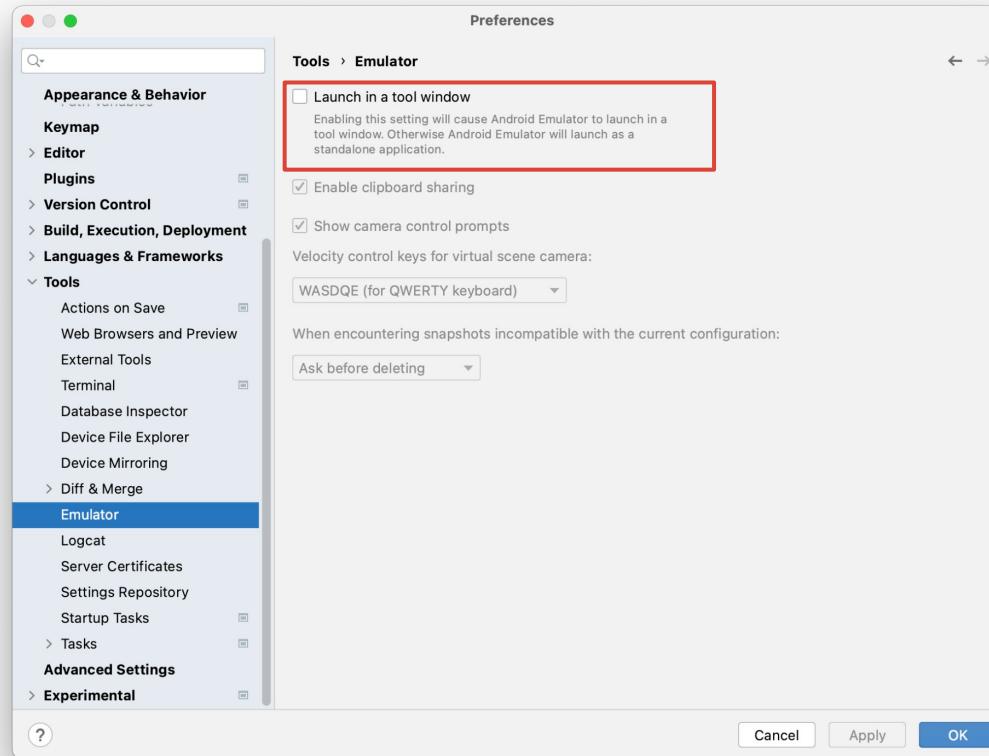
# 안드로이드 에뮬레이터 설정하기

Actions 탭에서 실행 (재생 버튼)을 누르면 에뮬레이터가 실행됩니다. (최초 실행시 2~3분가량 소요)



# 안드로이드 에뮬레이터를 별도 창에서 실행되도록 하기

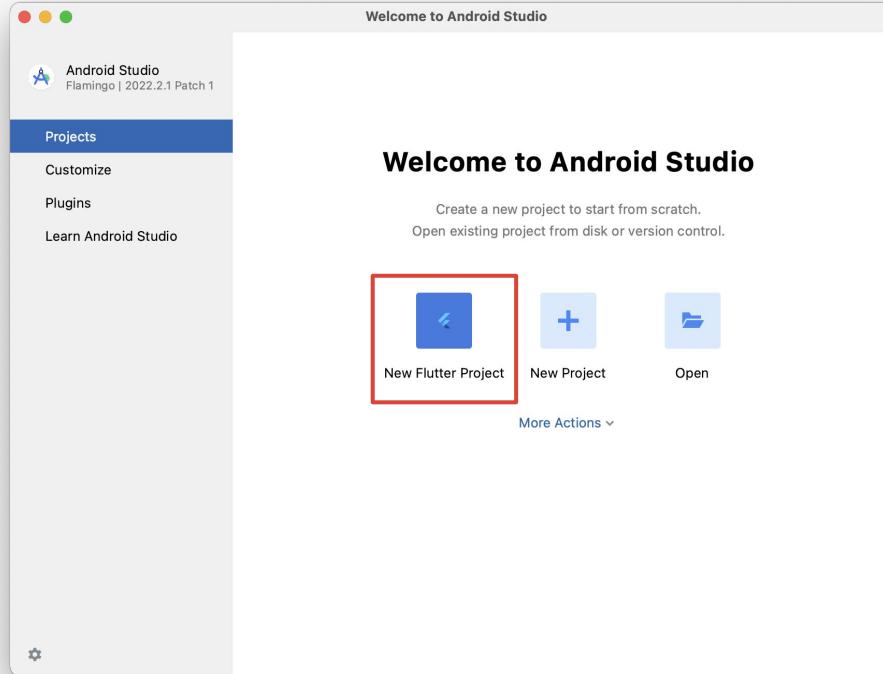
Settings > Tools > Emulator 화면에서 [Launch in a tool window](#)를 체크 해제합니다.



# 새 프로젝트 만들기

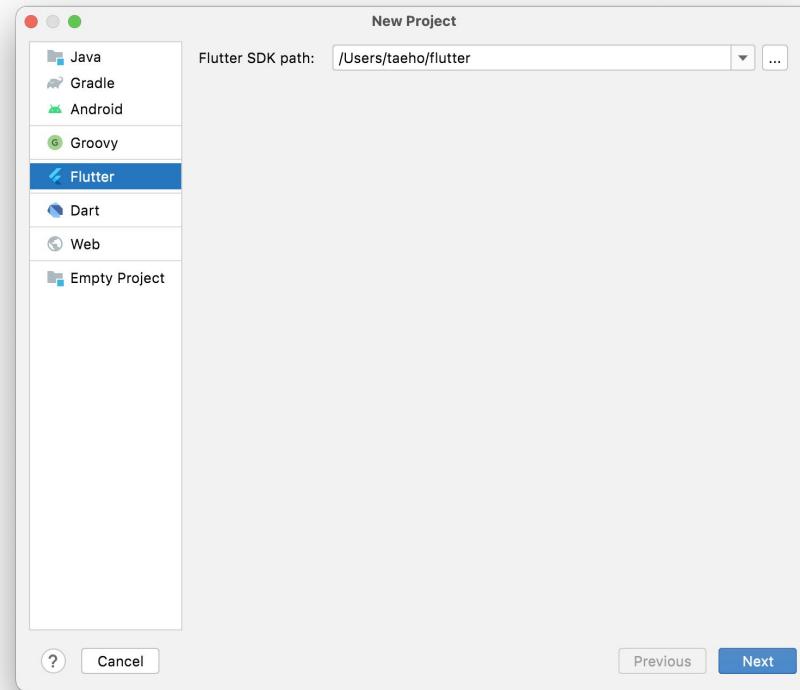
# 새 프로젝트 만들기

안드로이드 스튜디오 초기 화면에서 **New Flutter Project** 를 선택합니다.



# 새 프로젝트 만들기

SDK path에 앞에서 Flutter SDK를 설치한 경로가 나오는지 확인한 후, **Next** 버튼을 누릅니다.



## 새 프로젝트 만들기

다음과 같이 프로젝트 정보를 입력한 후, **Create** 버튼을 눌러 프로젝트를 생성합니다. (언급하지 않은 항목은 기본값 그대로 둡니다)

- Project name: hello\_world
- Organization: edu.skku.sco
- Platforms: Android, iOS

# 새 프로젝트 만들기

The screenshot shows the Android Studio interface with a Flutter project named "hello\_world".

**Project Structure:** The left sidebar shows the project structure with files like .gitignore, .idea, android, ios, lib, test, README.md, External Libraries, and Scratches and Consoles.

**Main.dart Code:** The main.dart file contains the following code:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".
}
```

**Toolbars and Side Panels:** The top bar includes tabs for README.md and main.dart, device selection, and various developer tools like Device Manager, Flutter Inspector, and Flutter Performance. The bottom bar includes tabs for Version Control, TODO, Problems, Dart Analysis, Terminal, Logcat, Profiler, App Inspection, Event Log, Layout Inspector, and Device File Explorer. A status bar at the bottom right indicates the terminal uses LF, UTF-8, 2 spaces, and a line number of 11.

# 새 프로젝트 만들기

## 플러터 프로젝트의 주요 구성요소

- android, ios 폴더: 안드로이드 및 iOS 앱을 만들 때 필요한 코드 (플러터로 구현된 부분을 감싸는 역할)
- lib 폴더: 플러터 소스 코드를 담고있는 폴더 (대부분의 코드는 여기에서 작성합니다)
- pubspec.yaml: 프로젝트 이름, 버전, 사용하는 라이브러리 등 프로젝트 구성에 필요한 주요 정보 포함

# 새 프로젝트 만들기

```
name: hello_world
description: A new Flutter project.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
version: 1.0.0+1

environment:
  sdk: '>=3.0.0 <4.0.0'

dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0

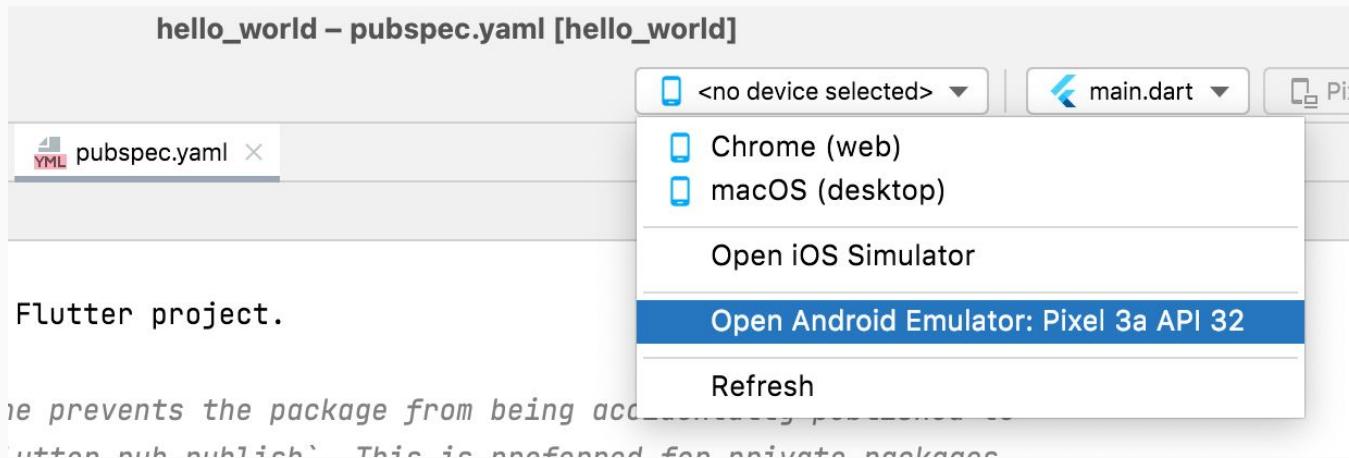
flutter:
  uses-material-design: true
```

pubspec.yaml

# 프로젝트 동작 원리 살펴보기

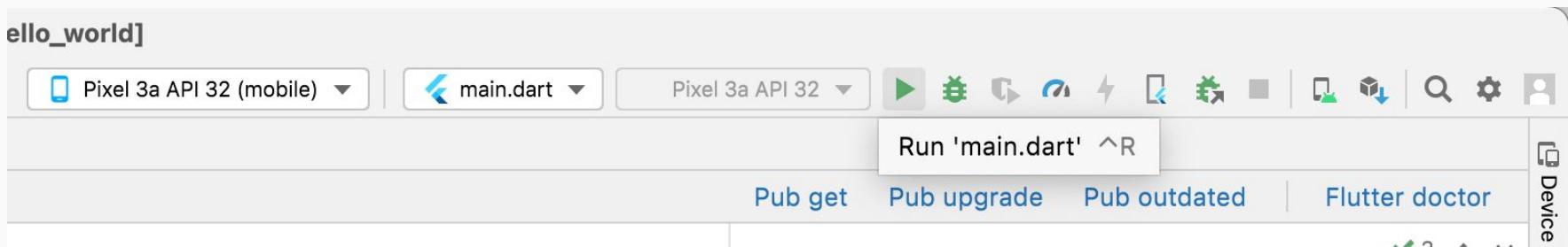
## 프로젝트 동작 원리 살펴보기

안드로이드 스튜디오 상단의 Flutter Device Selection 콤보박스를 누른 후, [Open Android Emulator](#)를 선택하여 안드로이드 에뮬레이터를 실행합니다. (메뉴가 표시되지 않는 경우 Refresh 선택 후 잠시 대기)



# 프로젝트 동작 원리 살펴보기

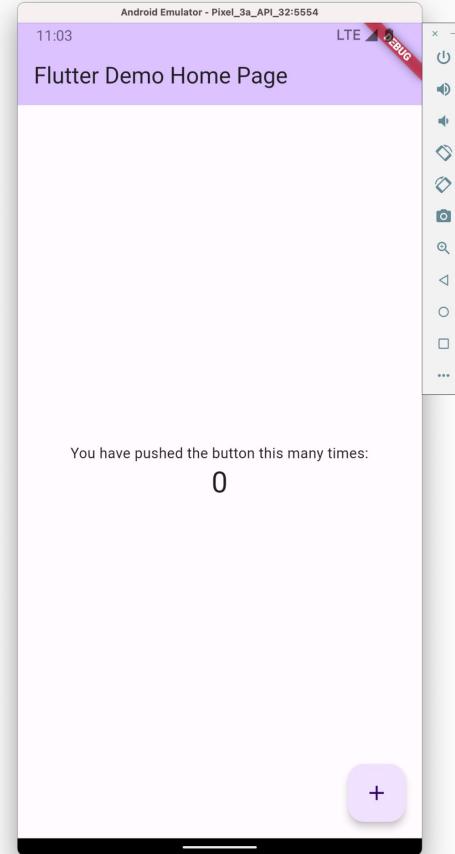
안드로이드 에뮬레이터가 실행된 후, **Run 'main.dart'** 버튼을 눌러 생성된 프로젝트를 에뮬레이터에서 실행합니다.



# 프로젝트 동작 원리 살펴보기

안드로이드 에뮬레이터에서 앱이 실행되는 모습을 확인합니다.

(주: 앱을 처음으로 실행하는 경우 빌드 도구를 새로 다운로드 한 후  
빌드를 진행하므로 앱이 실행되기 까지 몇 분가량 소요될 수 있습니다)



# 프로젝트 동작 원리 살펴보기

main.dart 파일을 열어 어떤 코드로 앱이 구성되어있는지 확인합니다.

- `void main()`
  - 최초로 실행되는 코드로, 애플리케이션 코드를 화면에 표시합니다.
- `class MyApp extends StatelessWidget`
  - 애플리케이션을 구성하는 코드로 앱의 첫 화면, 테마 등을 정의합니다.
- `class MyHomePage extends StatefulWidget`
  - 앱에서 가장 처음으로 보여줄 화면을 구성합니다. 상태에 따라 표시되는 화면을 변경할 수 있습니다.
- `class _MyHomePageState extends State<MyHomePage>`
  - 상태에 따라 화면을 그려주는 코드를 담고 있습니다.

# 프로젝트 동작 원리 살펴보기

## MyApp 클래스를 구성하는 주요 요소

- `MaterialApp`
  - 머티리얼 디자인을 따르는 앱을 구성할 때 사용합니다.
- `MaterialApp.theme`
  - 앱의 테마 색상을 결정합니다.
- `MaterialApp.home`
  - 앱에서 가장 처음으로 표시할 화면(위젯)을 지정합니다.

# 프로젝트 동작 원리 살펴보기

MyHomePage 클래스를 구성하는 주요 요소

- State<MyHomePage> createState()
  - 위젯의 상태에 따라 화면을 그려주는 클래스를 지정합니다.

# 프로젝트 동작 원리 살펴보기

## \_MyHomePageState 클래스를 구성하는 주요 요소

- `Widget build(BuildContext context)`
  - 위젯에서 표시할 화면을 그려주는 코드를 작성합니다.
- `Scaffold`
  - 머티리얼 디자인을 따르는 화면을 편리하게 구성할 수 있게 도와주는 위젯입니다.
- `AppBar`
  - 화면 상단부 (화면 이름, 액션 버튼 등)을 구성할 때 사용하는 위젯입니다.
- `Text`
  - 문자열을 표시할 때 사용하는 위젯입니다.
- `FloatingActionButton`
  - 화면 하단의 액션 버튼을 표시할 때 사용하는 위젯입니다.
- `setState()`
  - 화면을 다시 그려줍니다. 이 함수가 호출되어야만 변경된 값을 화면에 반영할 수 있습니다.