

GitHub Codespace – Flutter 개발환경 구축

본 문서에서는 Codespace 에서 Flutter 개발도구를 설치하여 사용하는 방법을 안내합니다.

목차

GitHub Codespace – Flutter 개발환경 구축.....	1
Flutter 설치.....	1
Flutter 프로젝트 생성 및 Web 플랫폼 테스트.....	2
Flutter – Android 용 앱 빌드.....	3
빌드된 앱 패키지 파일 다운로드 및 설치.....	6
추천 Codespaces 확장기능.....	8
유용한 문서 링크 모음.....	9

Flutter 개발환경이 구축된 Codespace 초기화

아래 링크를 접속하면 나오는 템플릿 저장소를 활용하여 Flutter 개발 환경이 구축 되어있는 Codespace 를 생성하여 사용할 수 있습니다.

<https://github.com/osamhack2022/flutter-devcontainer.git>

Flutter 프로젝트 생성, 앱 빌드 및 테스트, 개발 환경 구축 과정에 관한 내용이 다음 부분에서 계속됩니다. 이를 참고하셔서 Flutter 앱을 개발 및 테스트를 진행 하시기 바랍니다.

Flutter 설치

Flutter 는 공식 문서에 따르면 Snap 을 이용하여 설치할 것을 권장하고 있습니다. 그러나 GitHub Codespaces 는 컨테이너 환경으로 IDE를 제공하며, Snap 패키지 관리자는 컨테이너 환경 내부에 설치 및 사용이 어렵기 때문에, Flutter SDK를 수동으로 설치해 주어야 합니다.

수동 설치를 위해, Flutter 저장소를 Codespace에 복제합니다.

```
cd ~  
git clone https://github.com/flutter/flutter.git -b stable
```

Flutter SDK 에 대한 PATH를 잡아줍니다. (.bashrc 파일 가장 하단에 추가. 저장 후 source .bashrc 명령으로 적용)

```
export PATH="$PATH:`pwd`/flutter/bin"
```

flutter 명령을 실행할 수 있는지 확인합니다.

```
which flutter
```

개발에 필요한 기본 SDK 컴포넌트를 내려 받습니다.

```
flutter precache
```

Flutter 프로젝트 생성 및 Web 플랫폼 테스트

Flutter 프로젝트를 하나 생성하고, 웹 앱 플랫폼으로 테스트 해 보겠습니다.

먼저 Flutter 프로젝트를 하나 생성합니다. 여기서는 myapp 라는 이름으로 생성하겠습니다.

```
flutter create myapp
```

생성된 프로젝트 디렉터리에서 웹서버 모드로 앱을 실행합니다.

```
cd myapp  
flutter run -d web-server --web-hostname=0.0.0.0
```

터미널에서 보여주는 웹 서버 포트(여기서는 33193번 포트)를 확인 후, 터미널 탭 옆의 "Ports" 탭으로 이동합니다.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2
Launching lib/main.dart on Web Server in debug mode...
Waiting for connection from debug service on Web Server... 17.8s
lib/main.dart is being served at http://localhost:42555
The web-server device requires the Dart Debug Chrome extension for debugging. Consider using the Chrome or Edge devices for an improved development workflow.

🔥 To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".
codespace → /workspaces/notice/myapp (main X) $ flutter run -d web-server --web-hostname=0.0.0.0
Launching lib/main.dart on Web Server in debug mode...
Waiting for connection from debug service on Web Server... 17.9s
lib/main.dart is being served at http://0.0.0.0:33193
The web-server device requires the Dart Debug Chrome extension for debugging. Consider using the Chrome or Edge devices for an improved development workflow.

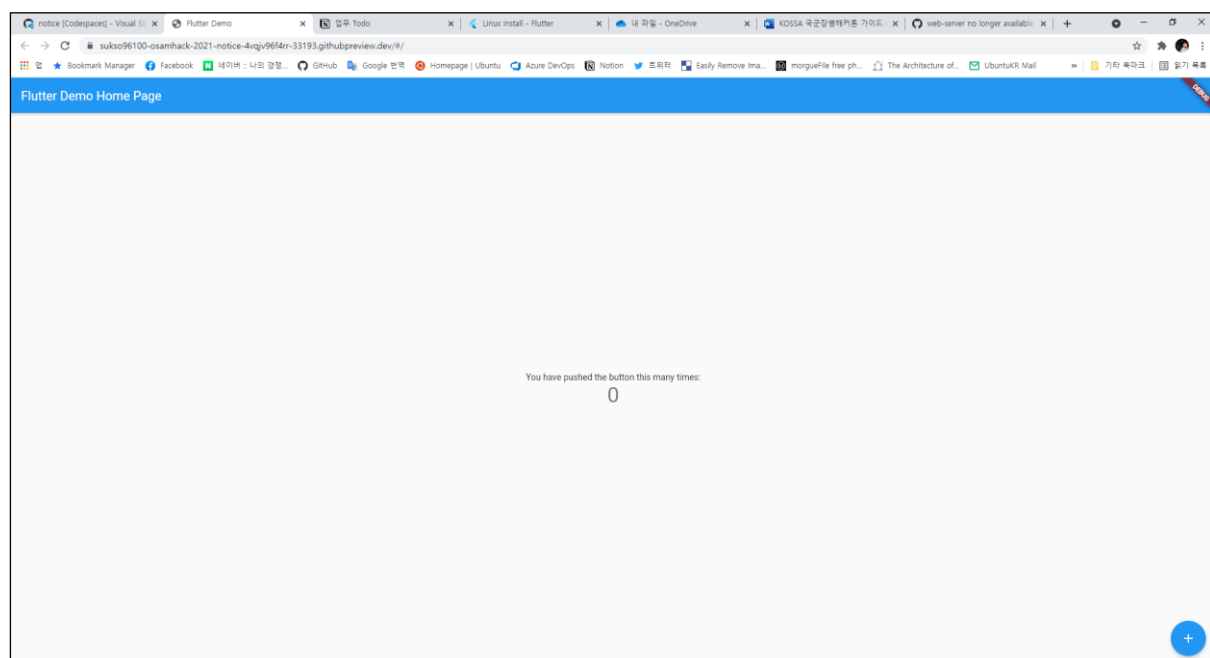
🔥 To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

```

노출된 포트 목록이 나옵니다. 터미널에서 본 포트에 해당하는 항목에서, Local Address 에 마우스를 올리면 지구본 버튼(Open in browser)이 나옵니다. 이를 클릭하여 Flutter 에서 띄운 웹서버에 접속합니다.

Port	Local Address	Running Process	Privacy	Origin
33193	https://suko96100-osamhack-2021-notice-4vqv964fr-33193.githubpreview.dev/#/	/home/codespace/flutter/bin/cache/dart-sdk/bi...	Private	Auto Forwarded
43181	https://suko96100-osamhack-2021-notice-4vqv964fr-43181.githubpreview.dev/#/	/home/codespace/flutter/bin/cache/dart-sdk/bi...	Private	Auto Forwarded

사진과 같이 샘플 앱이 구동되는 것을 확인하실 수 있습니다.



Flutter – Android 용 앱 빌드

Flutter는 크로스 플랫폼 앱 개발 프레임워크 입니다, 웹 뿐만 아니라 Android 및 iOS 용 앱도 빌드가 가능합니다. 여기서는 Android 앱 빌드를 위한 도구를 설치 및 구성해 보겠습니다.

아래 명령으로 Android SDK Command line Tools 를 설치합니다.

```
cd ~
wget https://dl.google.com/android/repository/commandlinetools-linux-7583922\_latest.zip
unzip commandlinetools-linux-7583922_latest.zip
mkdir android-sdk android-sdk/platform-tools android-sdk/cmdline-tools
mv cmdline-tools android-sdk/cmdline-tools/latest
rm commandlinetools-linux-7583922_latest.zip
```

Android SDK 에 대한 PATH를 잡아줍니다. (.bashrc 파일 가장 하단에 추가. 저장 후 source .bashrc 명령으로 적용)

```
export ANDROID_SDK_ROOT=$HOME/android-sdk
export PATH=$PATH:$ANDROID_SDK_ROOT/cmdline-tools/latest/bin
export PATH=$PATH:$ANDROID_SDK_ROOT/platform-tools
```

Android 11 (API Level 30) 에 해당하는 플랫폼 도구와 빌드 도구를 다운로드 합니다.

```
sdkmanager --sdk_root=${ANDROID_SDK_ROOT} "platform-tools" "platforms;android-30" "build-
tools;30.0.3"
Flutter SDK 에 Android SDK 경로를 설정해 줍니다.
flutter config --android-sdk $ANDROID_SDK_ROOT.
```

Flutter SDK 가 요구하는 Android SDK 설정이 잘 되었는지 확인하기 위해 아래 명령을 실행합니다.

```
flutter doctor
codespace →~ $ source .profile
codespace →~ $ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.2.3, on Linux, locale en_US.UTF-8)
[!] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
    ✗ Android license status unknown.
      Run `flutter doctor --android-licenses` to accept the SDK licenses.
      See https://flutter.dev/docs/get-started/install/linux#android-setup for more details.
[✗] Chrome - develop for the web (Cannot find Chrome executable at google-chrome)
    ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.
[!] Android Studio (not installed)
[!] Connected device
    ! No devices available

! Doctor found issues in 4 categories.
codespace →~ $
```

위와 같이 SDK 라이선스 동의가 완료되지 않았다고 나오면, 아래 명령으로 Android SDK 이용 라이선스에 동의 합니다.

flutter doctor --android-licenses

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
10.7 Special Terms for Pre-Release Materials. If so indicated in the description of the Evaluation Software, the Evaluation Software may contain Pre-Release Materials. Recipient hereby understands, acknowledges and agrees that: (i) Pre-Release Materials may not be fully tested and may contain bugs or errors; (ii) Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or FCC) have not been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments or may not be suitable for certain applications and (iv) MIPS can provide no assurance that it will ever produce or make 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
```

아래처럼 라이선스 동의가 완료되어, flutter doctor 명령 실행 시 Android toolchain 설정이 정상으로 나오면 안드로이드 앱 패키지로 빌드할 수 있습니다.

```
codespace → ~ $ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.2.3, on Linux, locale en_US.UTF-8)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[X] Chrome - develop for the web (Cannot find Chrome executable at google-chrome)
    ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.
[!] Android Studio (not installed)
[!] Connected device
    ! No devices available

! Doctor found issues in 3 categories.
```

아래 명령으로 *.apk 형식의 안드로이드 앱을 빌드합니다.

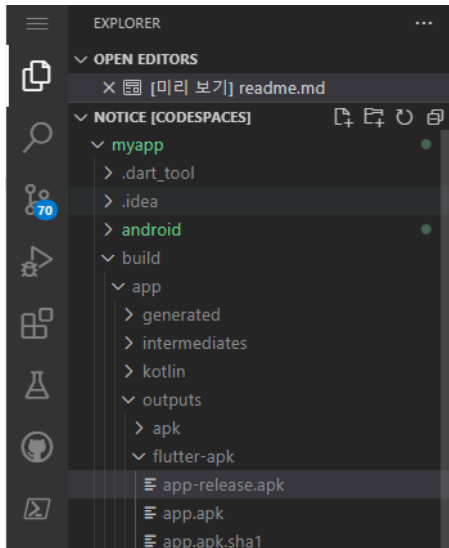
flutter build apk

```
codespace → /workspaces/notice/myapp (main X) $ flutter build apk

👉 Building with sound null safety 👉

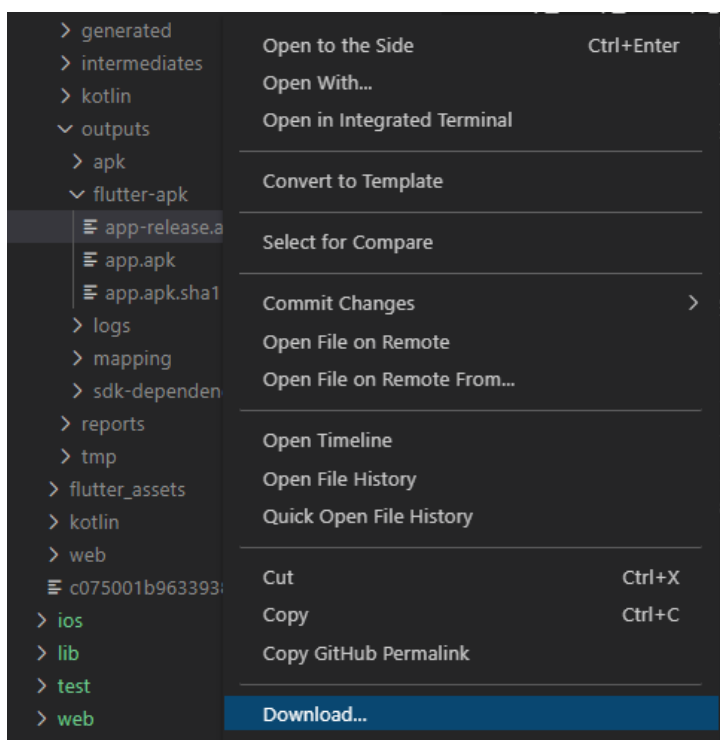
Warning: Mapping new ns http://schemas.android.com/repository/android/common/02 to old ns http://schemas.android.com/repository/android/common/01
Warning: Mapping new ns http://schemas.android.com/repository/android/generic/02 to old ns http://schemas.android.com/repository/android/generic/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/addon2/02 to old ns http://schemas.android.com/sdk/android/repo/addon2/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/repository2/02 to old ns http://schemas.android.com/sdk/android/repo/repository2/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/sys-img2/02 to old ns http://schemas.android.com/sdk/android/repo/sys-img2/01
Checking the license for package Android SDK Build-Tools 29.0.2 in /home/codespace/android-sdk/licenses
License for package Android SDK Build-Tools 29.0.2 accepted.
Preparing "Install Android SDK Build-Tools 29.0.2 (revision: 29.0.2)".
"Install Android SDK Build-Tools 29.0.2 (revision: 29.0.2)" ready.
Installing Android SDK Build-Tools 29.0.2 in /home/codespace/android-sdk/build-tools/29.0.2
"Install Android SDK Build-Tools 29.0.2 (revision: 29.0.2)" complete.
"Install Android SDK Build-Tools 29.0.2 (revision: 29.0.2)" finished.
Running Gradle task 'assembleRelease'...
Running Gradle task 'assembleRelease'... Done                      90.2s
✓ Built build/app/outputs/flutter-apk/app-release.apk (15.4MB).
codespace → /workspaces/notice/myapp (main X) $
```

빌드가 완료되어 생성된 파일은, 터미널에 출력된 것처럼 프로젝트 폴더 안의 build/app/outputs/flutter-apk/ 안에 있는 app-release.apk 파일 입니다.



빌드 된 앱 패키지 파일 다운로드 및 설치

Codespace 내부에 생성된 파일을 받거나 공유하려면 여러 방법이 있는데, 여기서는 두 가지 방법을 소개합니다. 하나는 해당 파일을 단순히 마우스 우클릭 하여 다운로드 하는 것입니다. 이 경우 Codespace 에 접속중인 PC 로만 파일을 받을 수 있습니다.



두 번째는 간단히 파일 다운로드 서버를 여는 것입니다. Codespaces 에 미리 포함된 Python 내장 Http 서버 모듈을 활용합니다.

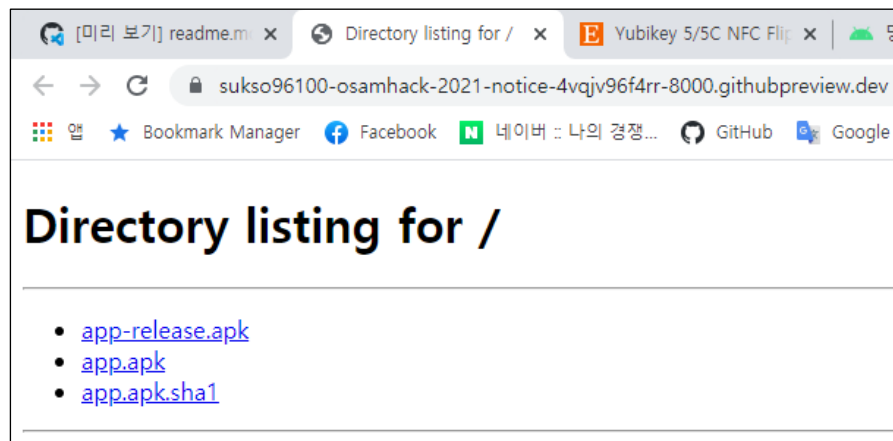
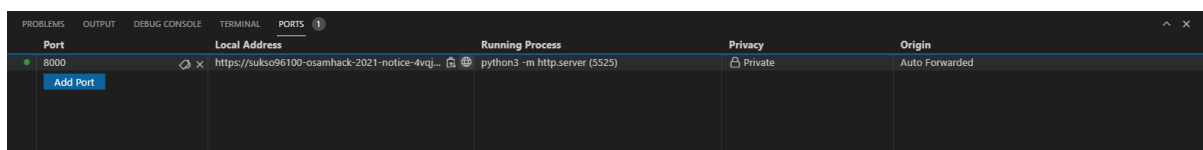
먼저 터미널에서 공유할 *.apk 파일이 있는 경로에 접근합니다.

```
cd myapp/build/app/outputs/flutter-apk/
```

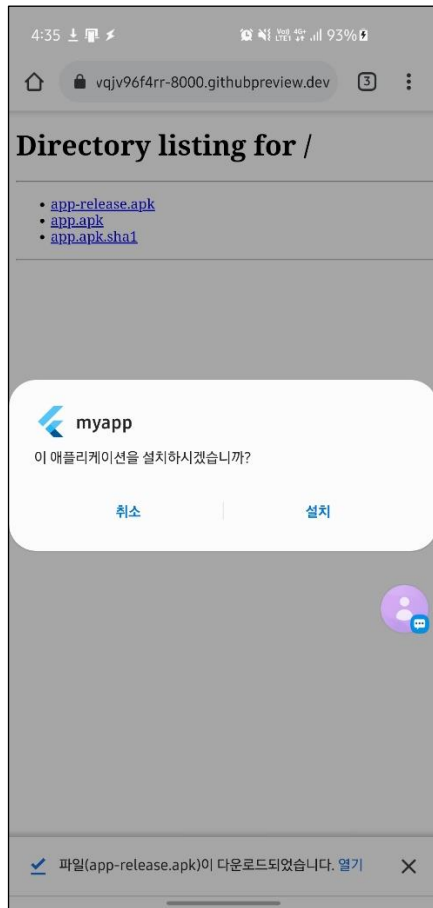
아래 명령으로 Python http server 를 구동합니다.

```
python3 -m http.server
```

이후 포트 목록에 나타나는 URL에 접속하여 파일을 받을 수 있습니다. 모바일에서도 해당 URL에 접속하여 USB 케이블 등 연결 없이 빌드된 앱 패키지 파일을 바로 내려받을 수 있습니다.



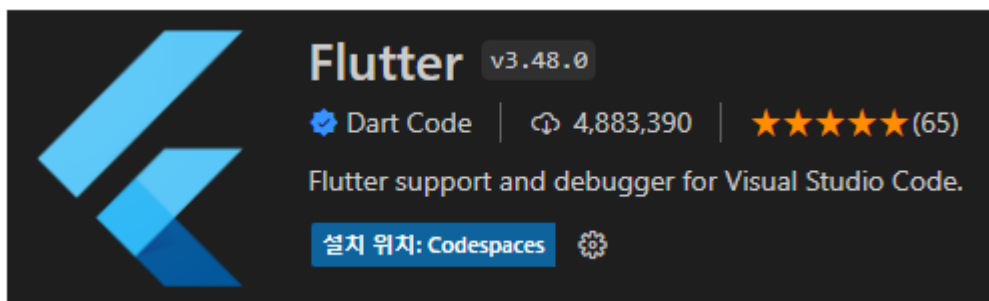
안드로이드 단말기에서 앱을 내려받아 설치하면, 아래 사진처럼 앱이 실행되는 것을 확인할 수 있습니다.



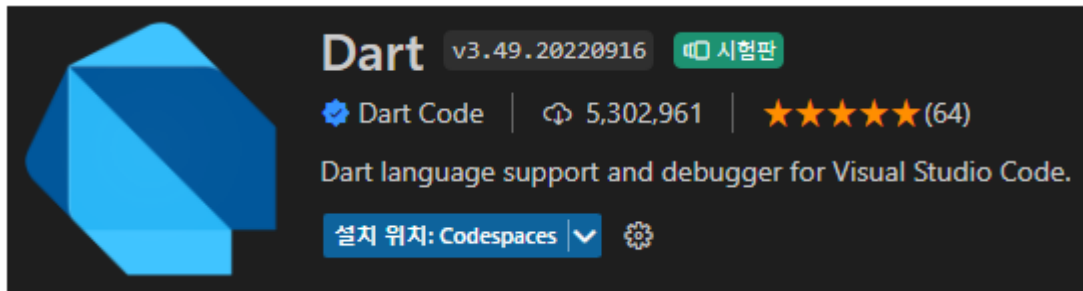
추천 Codespaces 확장기능

아래 VS Code/GitHub Codespaces 확장기능을 설치하면 Flutter 개발을 좀 더 효율적으로 할 수 있습니다.

Flutter – 개발자: Dart Code, 확장 ID: dart-code.flutter



Dart - 개발자: Dart Code, 확장 ID: dart-code.dart-code



유용한 문서 링크 모음

아래는 Flutter 앱 개발에 참고 할 만한 문서 링크 모음 입니다.

- Flutter documentation - <https://flutter.dev/docs>
- Flutter Linux install - <https://flutter.dev/docs/get-started/install/linux>
- Flutter Cookbook - <https://flutter.dev/docs/cookbook>
- Flutter 샘플 앱 소스코드 모음 - <https://flutter.github.io/samples>