



Flutter

2일차



DGROID
LEE HAIM



Flutter doctor

flutter doctor : Flutter가 잘 설치되었는지 확인하는 명령어
- cmd에 입력해서 내용을 확인한다.

```
C:\Users\haim1>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.2.3, on Microsoft Windows [Version 10.0.19041.1288], locale ko-KR)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] VS Code (version 1.60.0)
[✓] Connected device (2 available)

• No issues found!
```

>> 체크표시로 만든다.

Flutter doctor

* 자주 뜨는 메시지

[!] **Android toolchain** - develop for Android devices (Android SDK version 31.0.0)

X cmdline-tools component is missing

Run ``path/to/sdkmanager --install "cmdline-tools;latest"``

See <https://developer.android.com/studio/command-line> for more details.

X Android license status unknown.

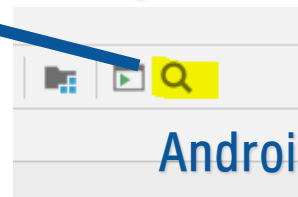
Run ``flutter doctor --android-licenses`` to accept the SDK licenses.

See <https://flutter.dev/docs/get-started/install/windows#android-setup> for more details.

```
[X] Android toolchain - develop for Android devices
X Unable to locate Android SDK.
  Install Android Studio from: https://developer.android.com/studio/index.html
  On first launch it will assist you in installing the Android SDK components.
  (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instructions).
  If the Android SDK has been installed to a custom location, please use
  `flutter config --android-sdk` to update to that location.
```

해결 : 안드로이드 스튜디오 > “sdk” **검색** > sdk manager > Android SDK > SDK Tool
> Android SDK Command-line Tools 설치

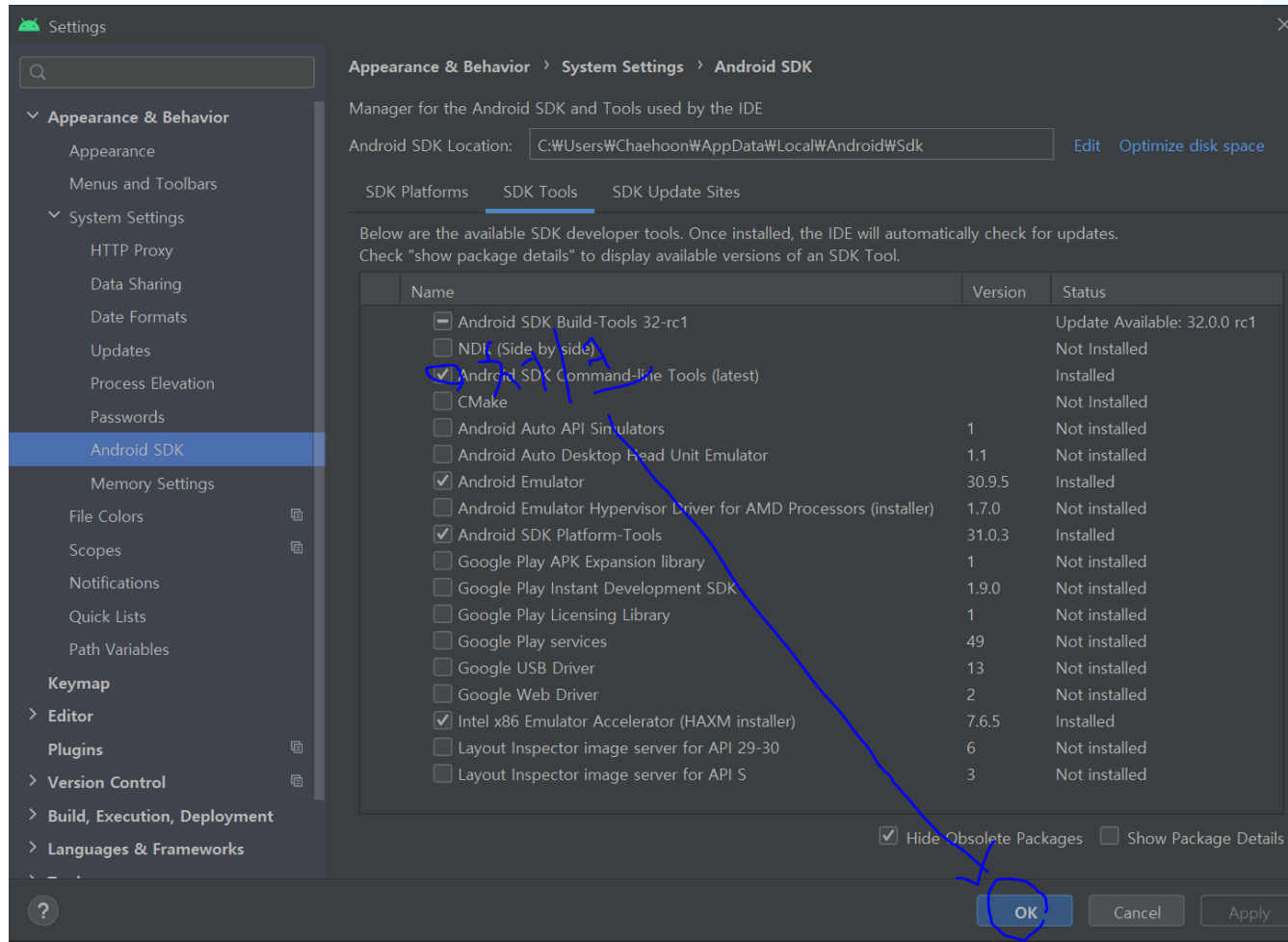
(다음 페이지에 사진 있음)



Android studio 오른쪽 상단

Flutter doctor

* 자주 뜨는 메시지



Android SDK Command-line Tools
를 설치한다!

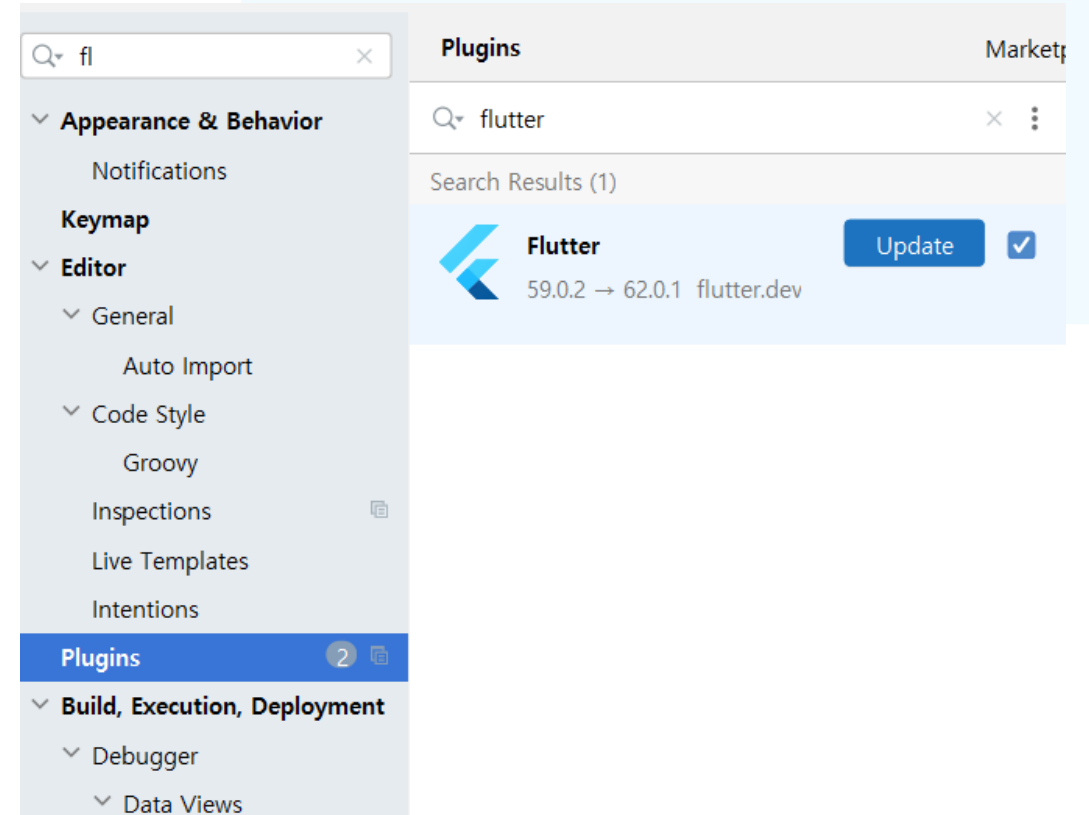
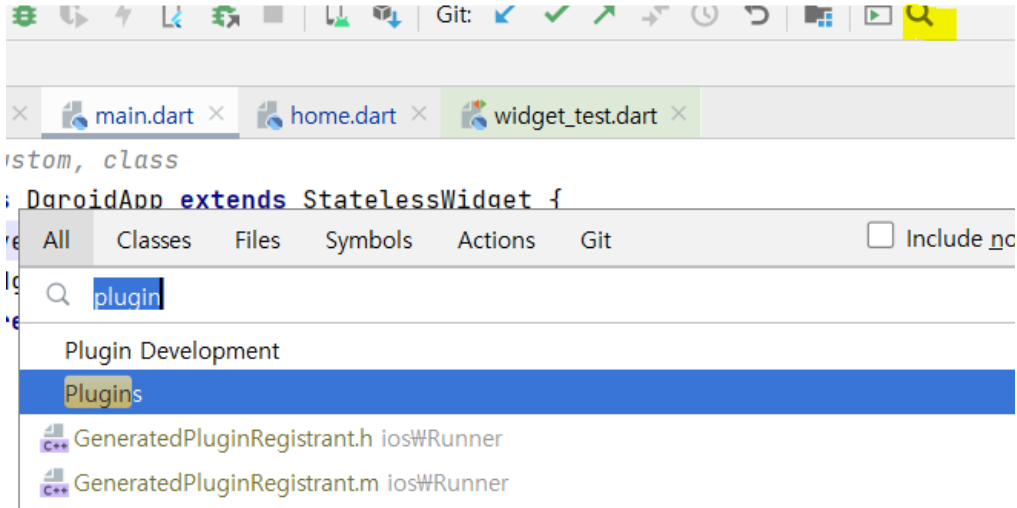
>> 이후 cmd를 켜다가 켜고,
flutter doctor 코드를 쳐서 확인.



New Flutter Project

- Android Studio

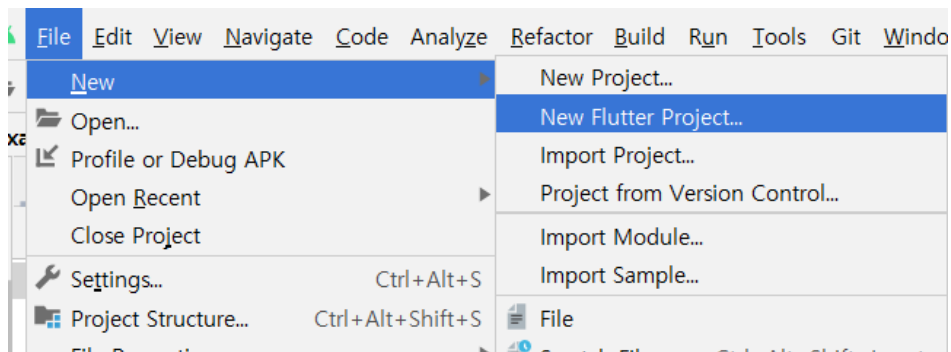
“Plugin” 검색 > Plugin > “Flutter” 검색 > 설치



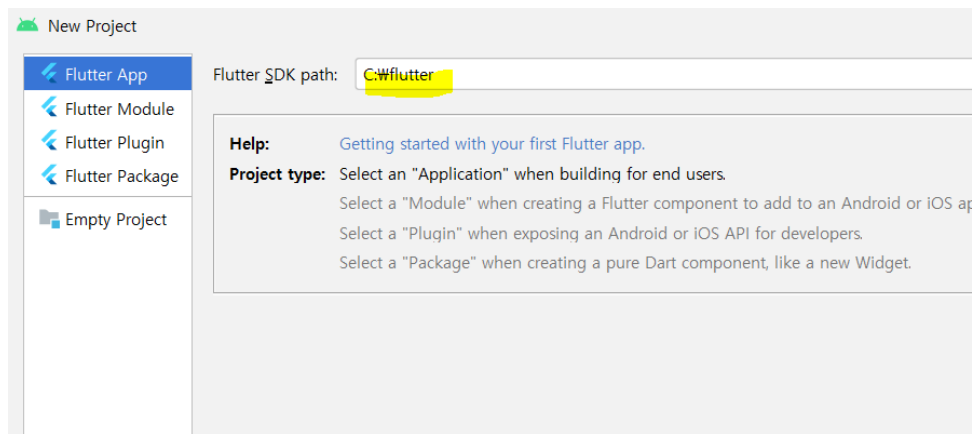


New Flutter Project

- Android Studio



File > New > New Flutter Project가 뜬다면 정상!



Flutter 폴더가 있는 위치를 입력하고 next를 눌러 새 프로젝트 형성.
(대부분) C:\flutter

tip : 시스템 환경 변수 편집에서 flutter\bin의 위치를 참고.



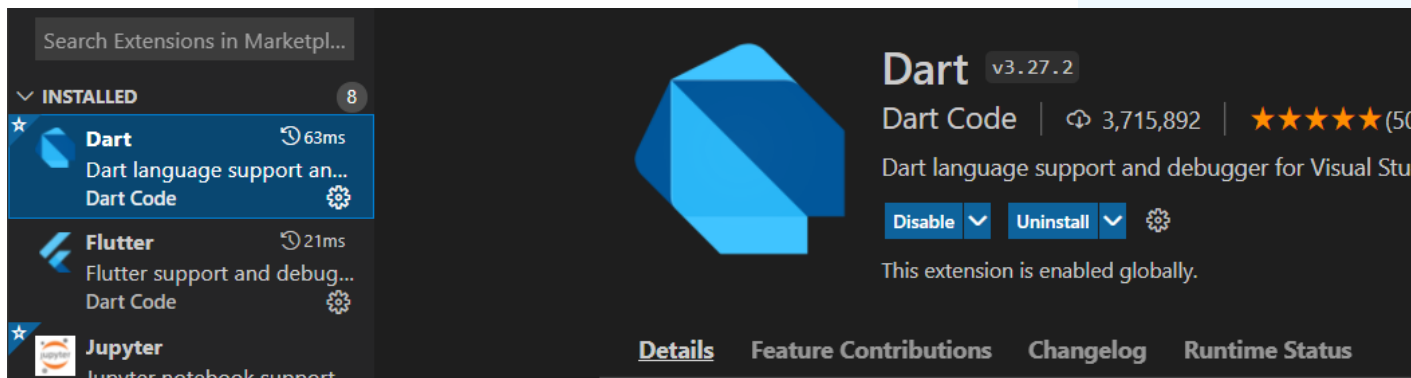
New Flutter Project

- Visual Studio Code

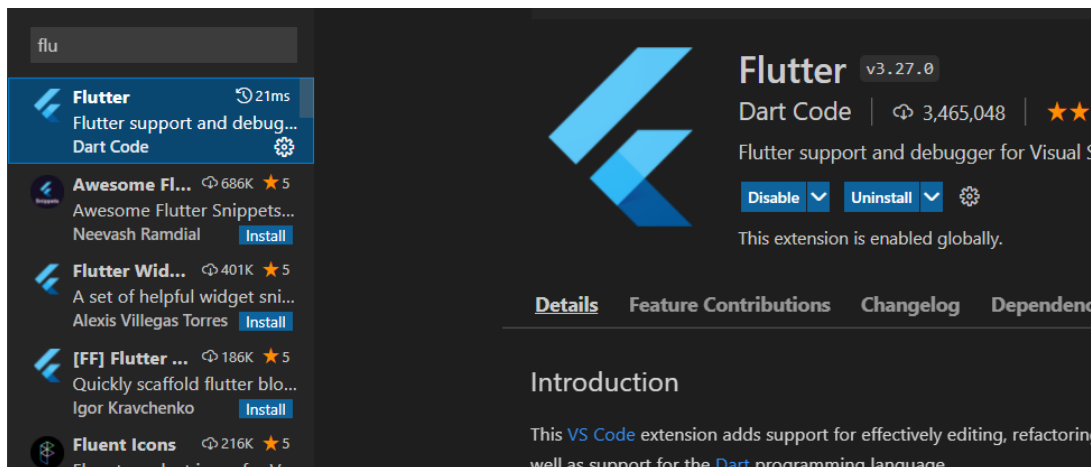
1)



<< 클릭 후



2) Dart 검색 후 설치

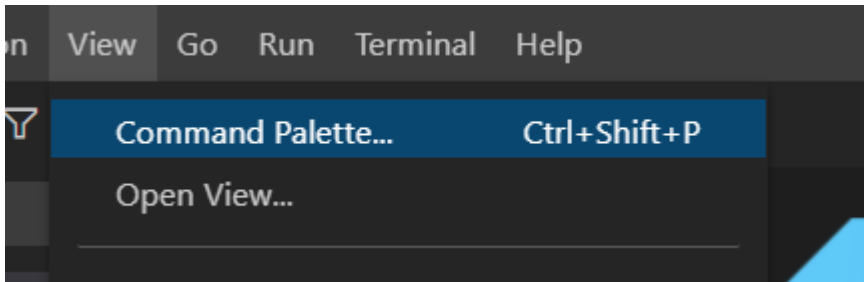


3) Flutter 검색 후 설치



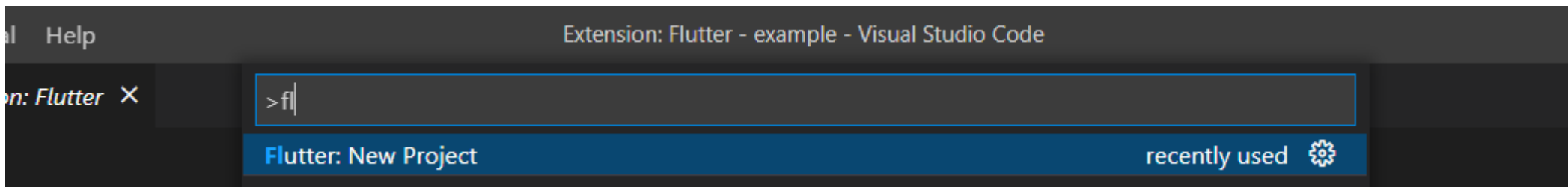
New Flutter Project

- Visual Studio Code



4)

View > Command Palette 클릭
(한글패치 시 '보기' 에 있음)



5) 새로이 뜨는 검색창에
flutter을 검색하고 밑에 뜨는 New Project 클릭 > 원하는 위치 선정(한글 폴더 비추천) 후 프로젝트 생성



New Flutter Project

- Visual Studio Code

실시간으로 바뀌는 어플 화면 확인을 위해 **에뮬레이터 설치** :

<https://visualstudio.microsoft.com/ko/vs/msft-android-emulator/>

>> 이 때 설치하지 못하는 오류가 발생한다면, Visual Studio Code를 포기하고
Android Studio를 새로 설치하여 진행하는 것을 추천.

- 컴퓨터 사양문제일 가능성이 큼.

참고: Visual Studio Emulator for Android가 릴리스된 후
최신 Android OS 이미지 및 Google Play 서비스에 대한 액세스를 제공하므로 가능하다면
환영 미리 보기를 사용해 Hyper-V에

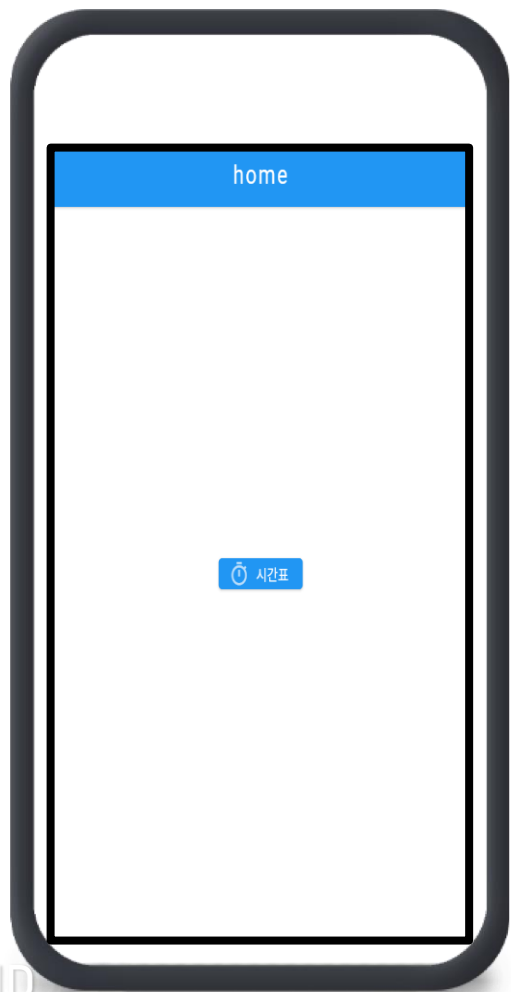
Visual Studio Emulator for Android
무료로 제공되는 업계 최고의 고속 Android
버그

에뮬레이터 다운로드

[설명서 >](#)

>> 다운로드 클릭 후 설치

Flutter 구조



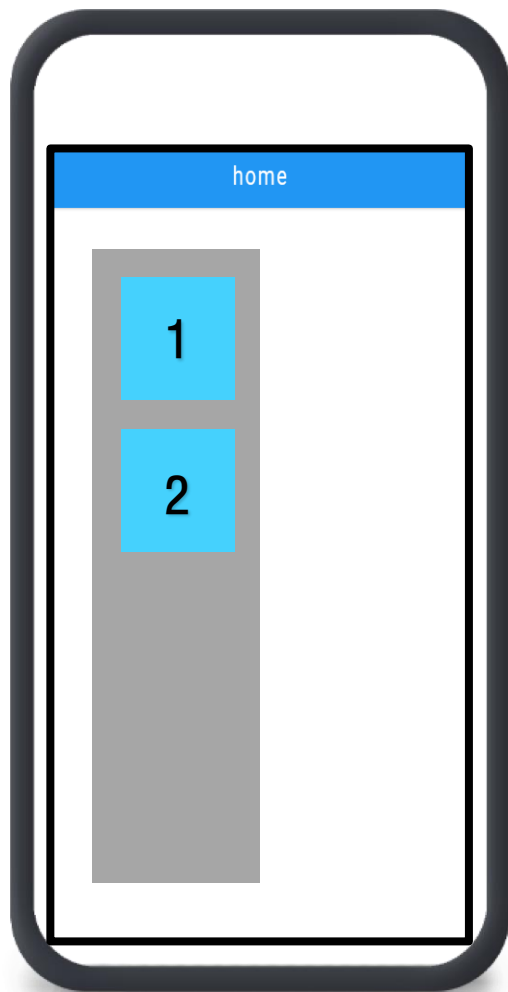
- AppBar

- Body

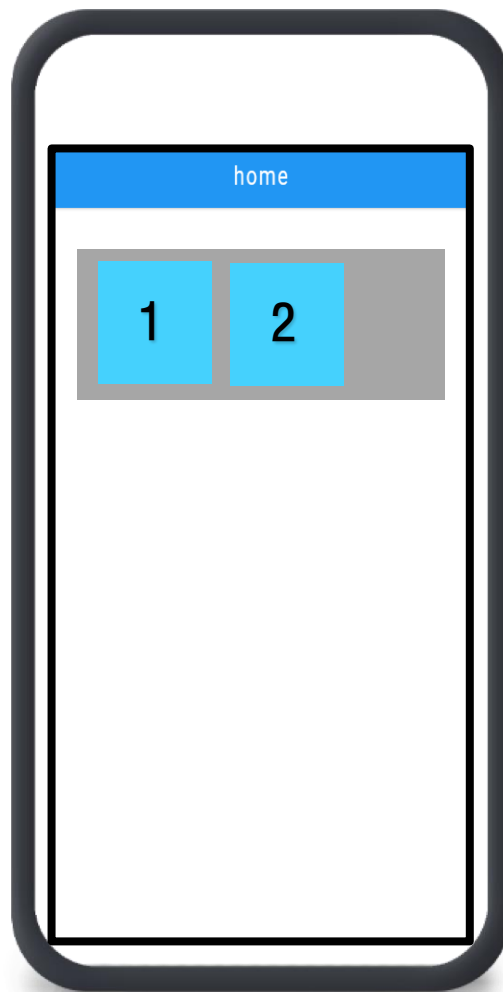
Scaffold

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      title: Text("home",
        style: TextStyle(
          color: Colors.white,
          letterSpacing: 2.0,
          fontSize: 23.0,
        ), // TextStyle
      ), // Text
    backgroundColor: Colors.blue,
    centerTitle: true,
    elevation: 2.0,
    iconTheme: IconThemeData(color: Colors.white),
  ), // AppBar
  body: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
```

Flutter 구조



Column



Row

Body 내부에 레이아웃을 형성할 수 있다.*

Column은 세로
Row는 가로로 형성되는 레이아웃이다.

각각의 레이아웃은 왼쪽 그림과 같이
코드의 순서에 따라 들어가는 위치 순서가 달라진다.

```
iconName: IconData(color: C
), // AppBar
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment
    children: [
      Row(
        mainAxisAlignment: MainAxisAlignment
        children: [
          ElevatedButton.icon(onPre
            print("버튼이 클릭되었습니다"),
            icon: Icon(Icons.time
              color: Colors.blue[10
            label: Text("시간표",
```

* 레이아웃 : 이미지, 텍스트 등이 들어갈 방식과 같은 개념

앱 제목 변경하기

main.dart 파일 내

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      title: "app title", // 앱을 총칭하는 이름  
    );  
  }  
}
```

>> 앱 제목을 변경한다. 어플 화면 상에서 달라지는 것은 없을 수 있다.

앱 제목 변경하기

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text(widget.title),  
    ), // AppBar  
    body: Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[
```

>> AppBar 아래의 제목을 변경한다. 위에 뜬 바에 글자가 달라진다.

```
return Scaffold(  
  appBar: AppBar(  
    title: Text("dgroid_study"),  
  ), // AppBar
```

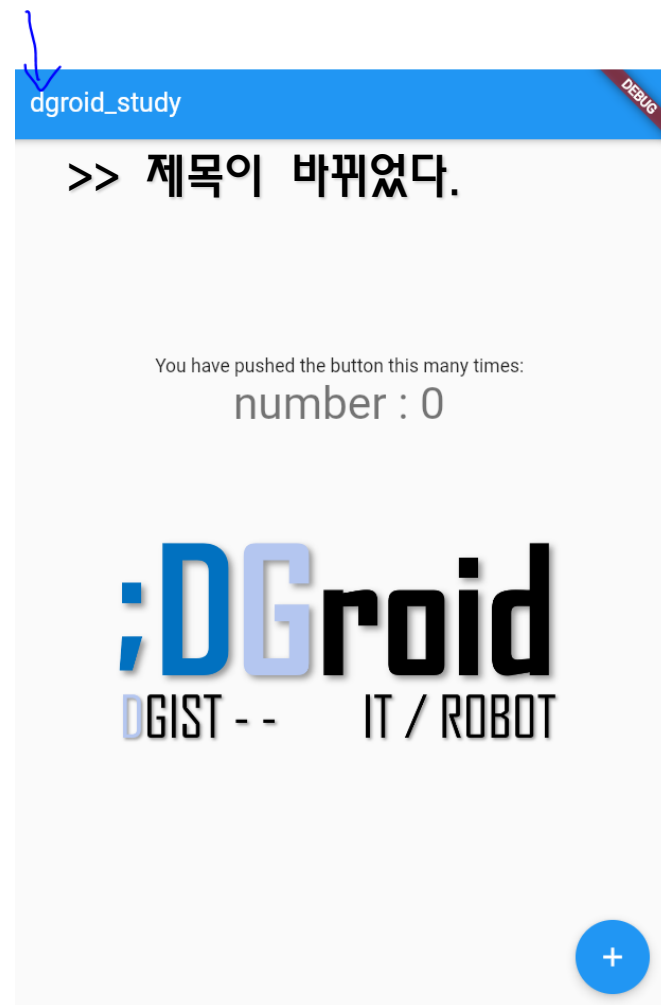
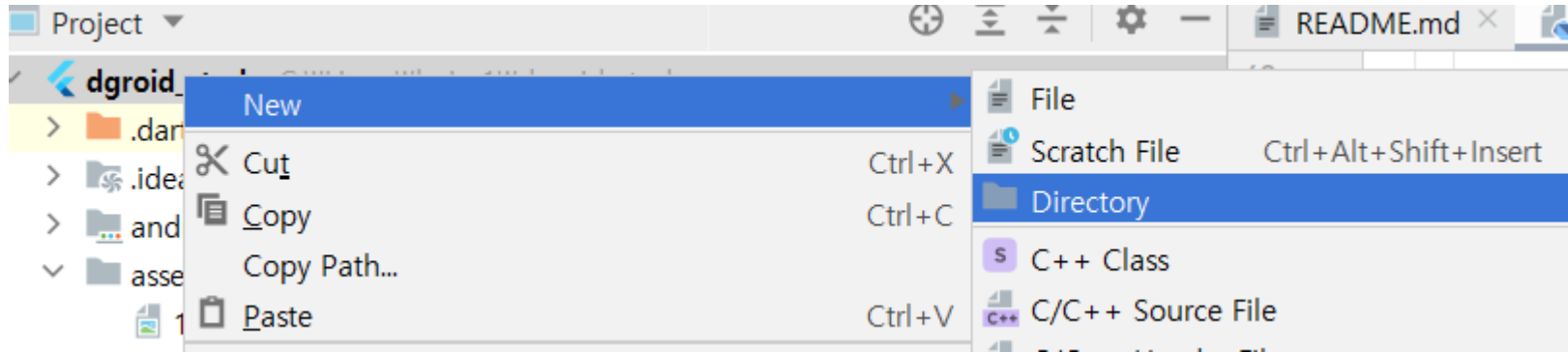
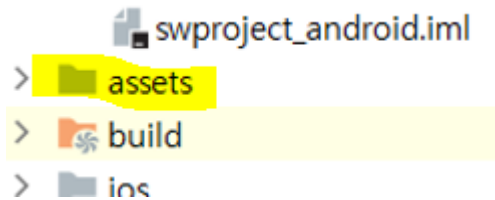




그림 삽입하기



>> **assets**라는 폴더를 새로 만든다.



>> 원하는 그림을 찾아 폴더 안으로 이동(복사)한다.

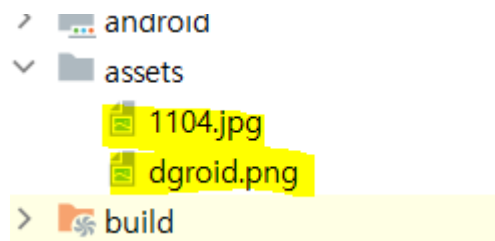
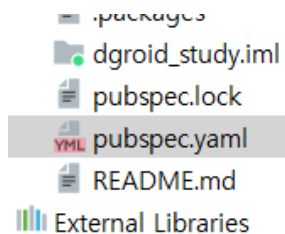


그림 삽입하기



>> **pubspec.yaml** 파일을 찾는다.

To add assets to your application, add an assets section, like this:

assets:

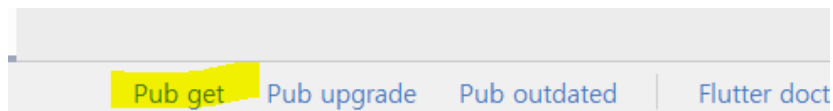
- assets/

>> 주석처리 된 **assets:** 부분의 주석처리를 지우고
왼쪽과 같이 입력한다.

*# An image asset can refer to one or more resolution-specific "variants".
<https://flutter.dev/assets-and-images/#resolution-aware>.*

For details regarding adding assets from package dependencies, see

- **assets** 파일에 있는 모든 이미지를 사용하겠다는 뜻이다.



>> 우측 상단의 **Pub get**을 누르고

Messages: [dgroid_study] Flutter ×
<C:\flutter\bin\flutter.bat> --no-color pub
Running "flutter pub get" in dgroid_study.
Process finished with **exit code 0**

>> 좌측 하단에 **exit code 0**이 뜨면 성공이다.



그림 삽입하기

```
- body: Center(  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
  
      SizedBox(height: 20,),  
      Text(  
        'You have pushed the button this many times:',  
      ), // Text  
      Text(  
        'number : $_counter',  
        style: Theme.of(context).textTheme.headline4,  
      ), // Text  
      Image(image: AssetImage("assets/dgroid.png"),  
        width: 500,), // Image
```

main.dart 에서
body > Center > child > Column > children 내부에
왼쪽과 같이 입력한다.

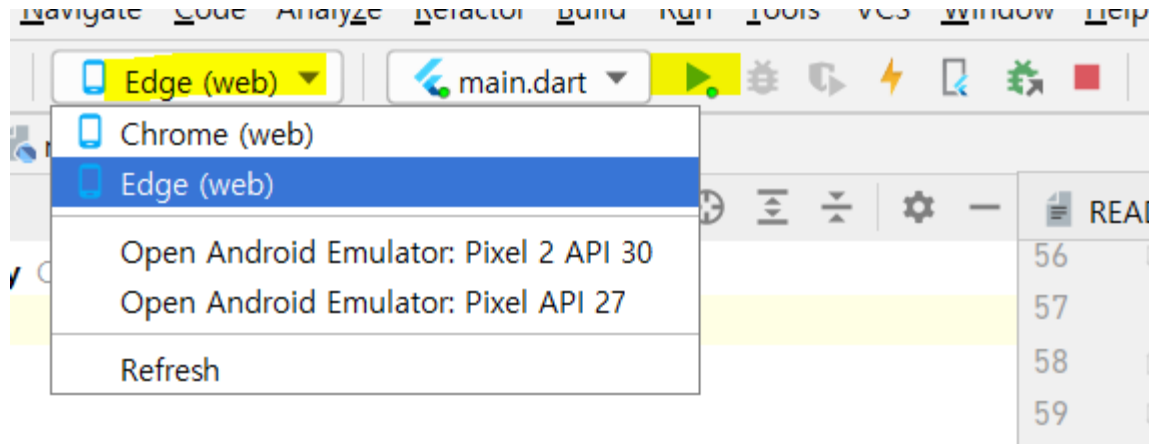
>> 되도록이면 코드 타이핑 시 뜨는 자동완성을 사용한다.
>> 괄호를 닫는 부분 뒤에 반드시 ,를 붙이도록 한다.

Image(image: AssetImage(“확장자포함 이미지 주소”),
width: 크기,),

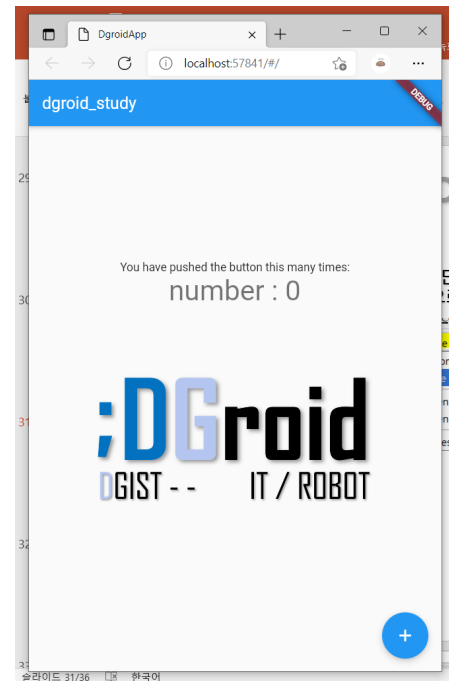
어플 실행하기

좌측 상단에 에뮬레이터를 선택한다.

기본적으로 깔려있는 **Edge(web)**, **Chrome(web)**을 사용하는 것을 추천한다.

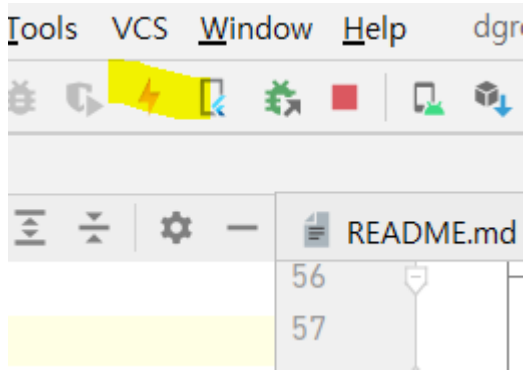


Run 버튼을 눌러 한동안 기다린다.
오류 발생 시 **좌측 하단**을 읽어본다.



앱 화면이 뜨면 성공이다.

Hot-Reload

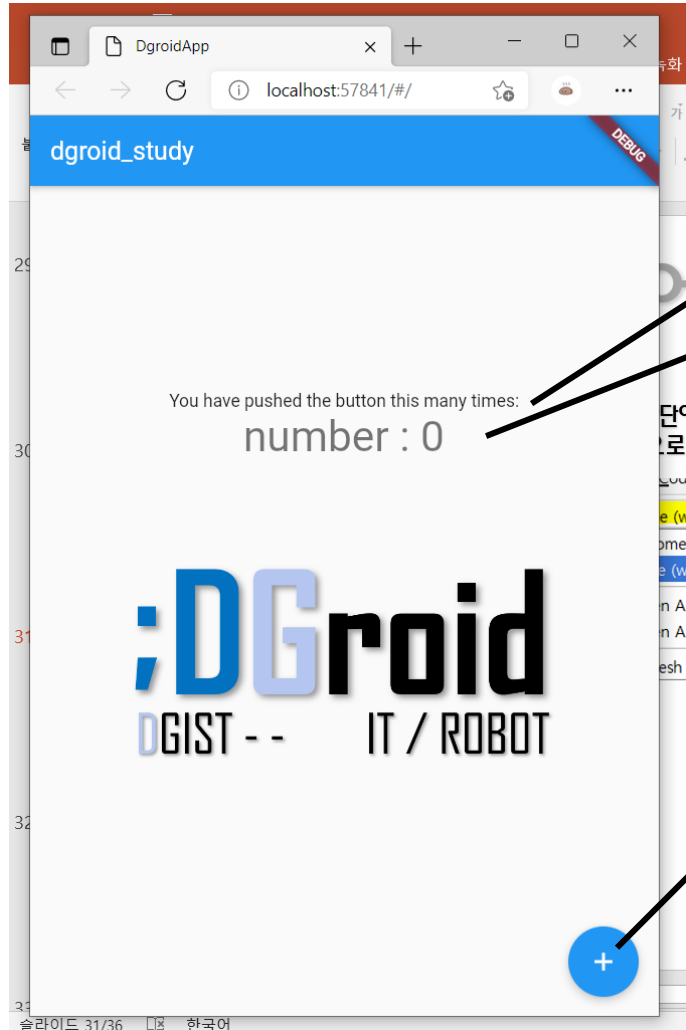


Hot-Reload란, 변경된 코드를 찾아 순식간에 어플로 띄워주는 것을 말한다. 에뮬레이터가 실행되고 있는데 **코드를 변경**했을 경우, Hot-Reload를 사용하여 변경된 부분을 빨리 확인한다.

Ctrl+S를 사용하여 파일을 **저장**하여도 Hot-Reload가 된다. 단, **사진**을 새로 넣었을 경우에는 에뮬레이터 전체를 껐다 실행하여야 된다.



Flutter 기본 테마 설명



```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text('dgroid_study'),  
    ),  
    body: Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
          Text(  
            'You have pushed the button this many times:',  
          ),  
          Text(  
            'number : $_counter',  
            style: Theme.of(context).textTheme.headline4,  
          ),  
          Image(image: AssetImage('assets/dgroid.png'),  
            width: 500,),  
        ],  
      ),  
    ),  
    floatingActionButton: FloatingActionButton(  
      onPressed: _incrementCounter,  
      tooltip: 'Increment',  
      child: Icon(Icons.add),  
    ),  
  ),  
);  
}
```

문자열을 출력한다.

`$_counter`이라는 변수가 저장하고 있는 값을 앱에 출력한다.

버튼을 누르면 `_incrementCounter` 함수가 작동한다.

```
void _incrementCounter() {  
  setState(() {  
    _counter++;  
  });  
}
```

`_incrementCounter` 함수 :
사용자 정의 함수로써,
`_counter`이라는 변수가 1씩 증가한다.