



Flutter

모든 것이 위젯.

높은 생산성.

하나의 코드로 Android와 iOS 양 플랫폼에서 동작하는 앱 개발.

아름다운 UI.

Android의 Material 디자인과 iOS의 Cupertino 디자인 모두 사용 가능.

네이티브 성능.

Dart 언어.

01. 우리는 왜 플러터를 학습해야 하는 가

커널이란?

OS에서 가장 핵심적인 역할을 하는 부분

하드웨어와 소프트웨어를 연결시켜주는 역할

- **마이크로 커널 (Micro kernel) → fuchsia OS**
 - 커널의 필수적인 기능 중심
 - 그 외의 기능은 모듈화하여 조립식으로 사용
- **단일형 커널 (Monolithic kernel) → android**
 - 커널에 필요한 모든 기능을 한 곳에 위치
 - 커널이 한 곳에 모여 서로 얽혀 있어 업데이트와 수정의 어려움

What is the flutter

- Flutter는 프레임워크
- Fuchsia의 사용자 인터페이스와 애플리케이션을 만들기 위해 사용
- Flutter 앱 개발을 위한 프로그래밍 언어로 Dart 사용
- 크로스 플랫폼(안드로이드, IOS) 애플리케이션 개발
- 인터페이스나 사용자 경험을 하나로 통일

02. SDK 설치와 환경변수 설정하기

SDK란?

- Software Development kit
- 소스코드의 모음과 유틸리티
- 구글이 제공

bin - binary 의미

```

C:\ Command Shell
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Windows Resource Kits\Tools>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                Print this usage information.
-v, --verbose              Noisy logging, including all shell commands executed.
                           If used with "--help", shows hidden options. If used with "flut
                           diagnostic information. (Use "-vv" to force verbose logging in
-d, --device-id            Target device id or name (prefixes allowed).
  --version                Reports the version of this tool.
  --suppress-analytics      Suppress analytics reporting when this command runs.

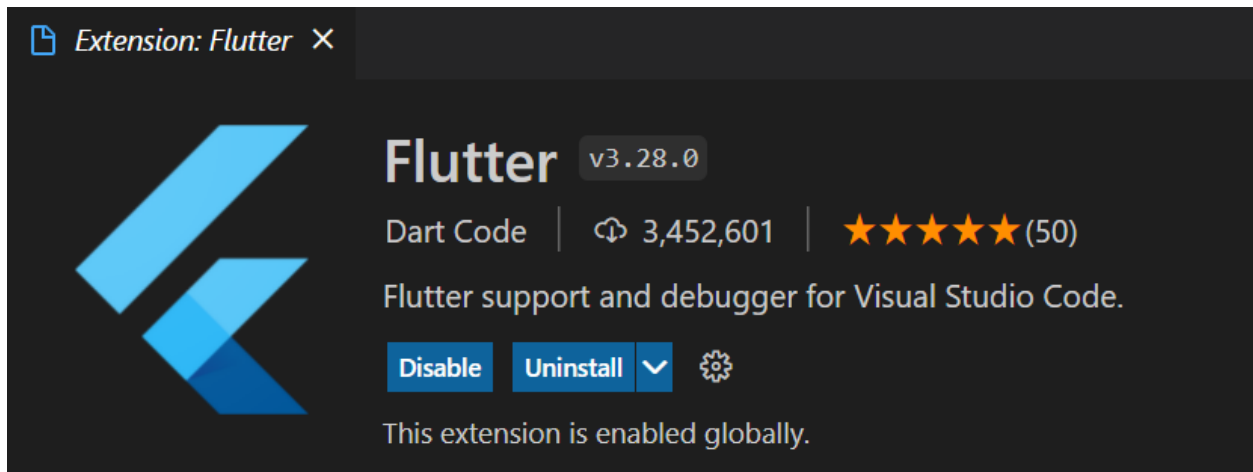
Available commands:
analyze                    Analyze the project's Dart code.
assemble                  Assemble and build Flutter resources.

```

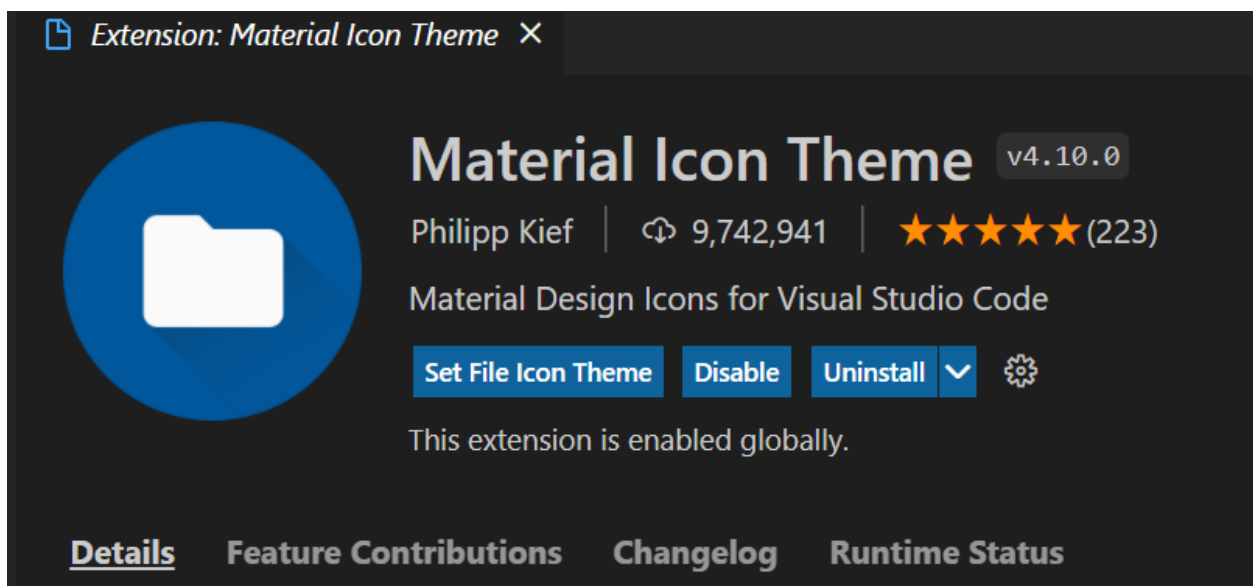
flutter - 환경변수 설정 완료 캡처

03. 안드로이드 스튜디오, 비주얼 스튜디오 코드, 에뮬레이터 설치하기

1. Android studio / 에뮬레이터 설치
2. Visual Studio Code(ide) 설치



visual studio 내 extension 설치 완료 1



visual studio 내 extension 설치 완료 2

04. 플러터 닥터 설정 및 프로젝트 생성하고 에뮬레이터에서 실행해보기

1. Flutter doctor로 실행환경 점검
2. Visual Studio Code로 프로젝트 생성 / 에뮬레이터 실행
3. Android studio로 프로젝트 생성 / 에뮬레이터 실행

```
Command Shell
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:\Program Files (x86)\Windows Resource Kits\Tools>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
✓ Flutter (Channel stable, 2.5.3, on Microsoft Windows [Version 10.0.19043.1288], locale ko-KR)
✓ Android toolchain - develop for Android devices (Android SDK version 31.0.0)
✓ Chrome - develop for the web
✓ Android Studio (version 2020.3)
✓ VS Code (version 1.62.0)
✓ Connected device (2 available)

• No issues found!

C:\Program Files (x86)\Windows Resource Kits\Tools>
```

에뮬레이터 선택 방법

- **Visual Studio**

ctrl+shift+p or command palette 메뉴에서 flutterse라고 검색 후 flutter:select device 메뉴
찾은 후 에뮬레이터 지정

- **Android Studio**

device select → run

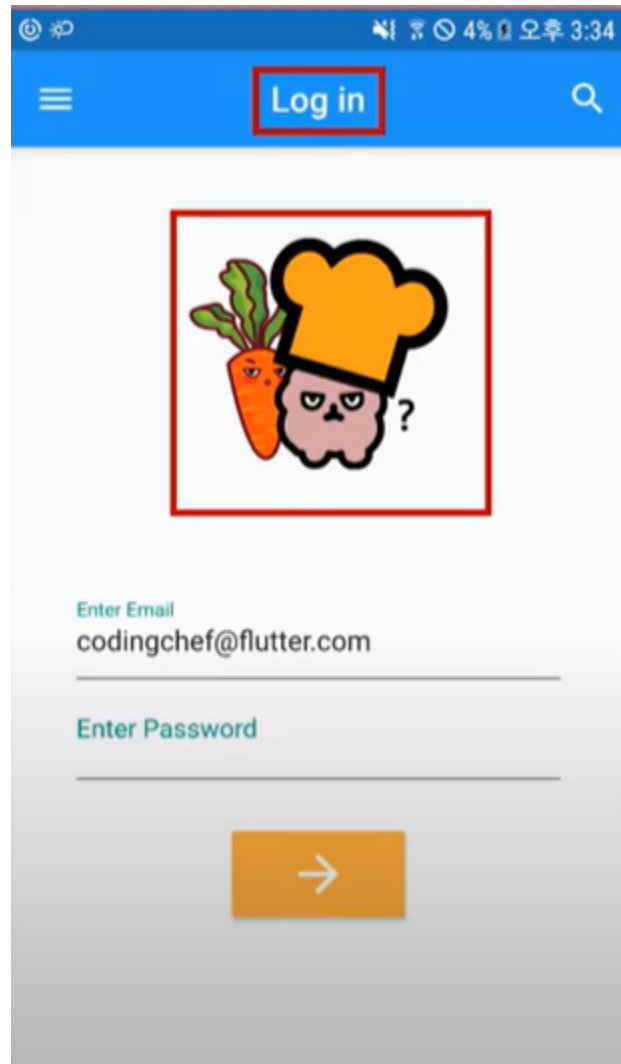
05. 플러터에서 제일 중요하다는 위젯이란 무엇일까요?

Flutter 상에서 위젯이란?

Widget

1. 독립적으로 실행되는 작은 프로그램
2. 주로 바탕화면 등에서 날씨나 뉴스, 생활 정보 등을 보여줌

- 3. 그래픽이나 데이터 요소를 처리하는 함수를 가지고 있음
- 1. UI를 만들고 구성하는 모든 기본 단위 요소
- 2. 눈에 보이지 않는 요소들까지 위젯 (ex. center, colymn, padding)
- 3. Everything is a widget
- 1. **Stateless widgets(상대가 없는 정적인 위젯) vs. Stateful widgets(계속 움직임이나 변화가 있는 위젯)**
 - Stateful → Value 값을 지속적으로 추적 보존
 - Stateless → 이전 상호작용의 어떠한 값도 저장하지 않음
 - 어떤 변화(모양, 상태)를 유발시키는 value 값을 가지지 않음
- **Stateless Widgets**
 - 스크린상에 존재만 할 뿐 아무것도 하지 않음
 - 어떠한 실시간 데이터도 저장하지 않음



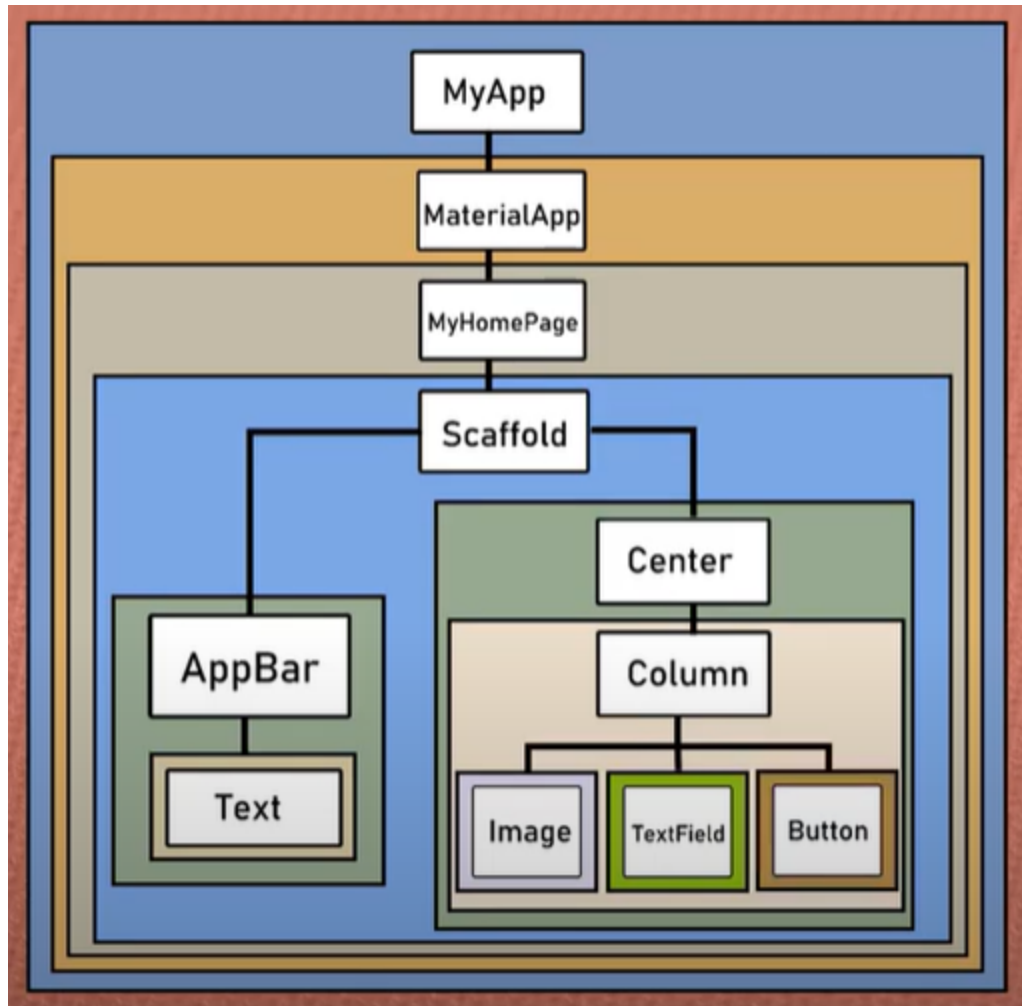
- **Stateful Widgets**

- 사용자의 interaction에 따라서 모양이 바뀜
- 데이터를 받게 되었을 때 모양이 바뀜

ex. 이메일 입력창

- **Flutter Widget tree**

- 위젯들은 tree 구조로 정리될 수 있음
- 한 widget 내에 얼마든지 다른 widget들이 포함될 수 있음
- widget은 부모 위젯과 자식 widget으로 구성
- 부모 widget을 widget container라고 부르기도 함



06&07&08. 프로젝트 폴더와 기본 코드 이해하기 1 &2

- **pubspec.yaml**
프로젝트의 메타데이터 저장
버전, 사용환경, 라이브러리 정의
- **lib > main.dart**


```

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

void main() => runApp(MyApp()); //괄호 안에는 widget

// 직접 MyApp() 위젯 만들기
// MyApp은 앱의 뼈대 역할 (Stateless, 정적인 역할)

///stl
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    return MaterialApp(
      title: 'First app',
      theme: ThemeData(
        primarySwatch: Colors.blue //Swatch는 견본 의미 (여기서는 앱의 색상 견본 의미)
      ), //ThemData

      home: MyHomePage(),
      //home은 앱이 정상적으로 시작되었을 때 가장 먼저 보여지는 경로
      //따라서 현재 앱이 실행된다면, MyHomePage 위젯 내용이 가장 먼저 보여질 것
    );
  }
}

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context){
    return Scaffold( // scaffold 발판, 빈도화지

      appBar: AppBar(
        title: Text('Hello World'),
      ), //AppBar
      body: Center(
        child: Column( //위젯 내 모든 요소를 세로로 배치
          children: <widget>[
            Text('Hello'),
            Text('Hello'), //위젯 간 심표로 구분
            Text('Hello');
          ], //<Widget>[]
        ), //Column
      ), //Center
    );
  }
}

```

- 클래스명과 함수명

클래스는 보통 대문자로 시작 : MyApp()

함수는 보통 소문자로 시작: main(), runApp()

- Custom 위젯을 만들 때는 항상 stateless 인지 stateful 인지 고민해보기