

Flutter

김 순곤

soongon@hucloud.co.kr

Flutter 독차

- Flutter 개발 소개
- 개발을 위한 설치와 셋업
- Flutter 앱 처음부터 작성하기
- 위젯 사용하여 멋진 UI 만들기
- 상태있는 위젯, StatefulWidget - 주사위 굴리기 앱 작성
- HTTP로 데이터 가져오기 - 날씨 앱 작성
- 플러터 기본을 넘어..

- Flutter 란?
- 왜 Flutter 인가?
- Flutter 앱 해부

1장. Flutter 개발소개

개발자 분류

- **프론트엔드 개발자**
 - 화면 개발 (앱개발자, 웹개발자, 데스크톱 앱)
- **백엔드 개발자**
 - 데이터, 기능에 대한 서비스 개발
- **데이터 사이언티스트(엔지니어)**
 - 데이터에 대한 관리와 제공 담당
- **데브옵스 개발자**
 - 서비스의 배포와 운영 담당

모바일 개발자

- 모바일 앱 개발 방식 - 화면 + 데이터(기능)

- 네이티브 앱

- ❖ Android, iOS - 최고의 성능과 사용성

- 하이브리드 앱 - ionic

- ❖ 웹 개발을 그대로 변환, 빠른 개발과 느린 성능

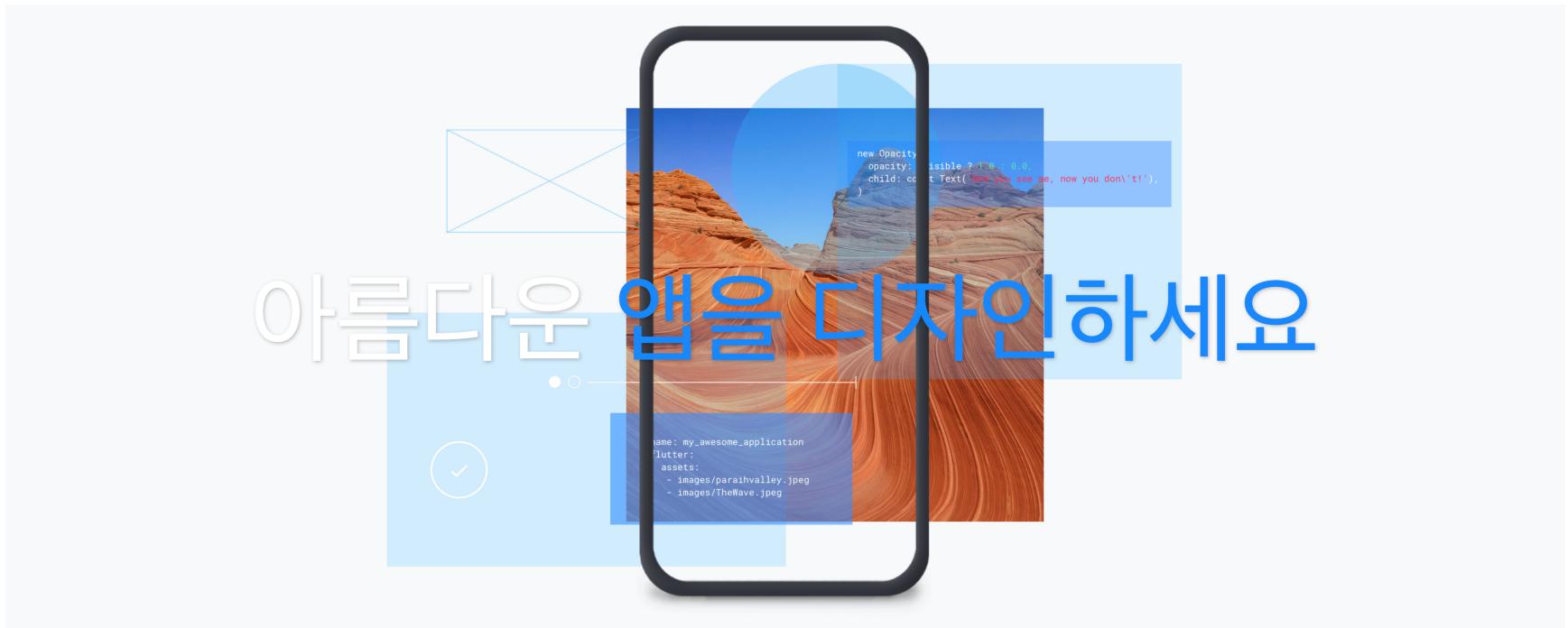
- 크로스 플랫폼 앱

- ❖ 빠른 개발과 높은 성능 보장

- ❖ React Native vs. Flutter (dart)

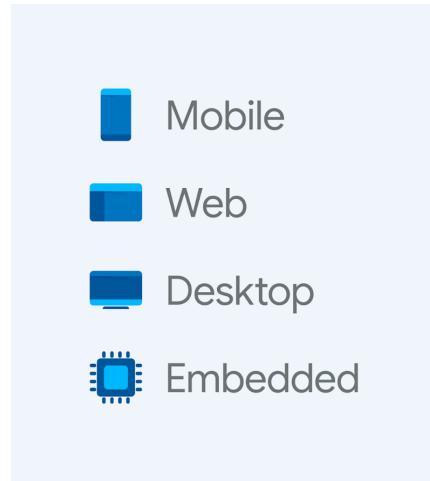
Flutter

- flutter.dev



왜 Flutter 인가?

- One Codebase - dart
 - Android
 - iOS
 - Windows / MacOS / Linux desktop App
 - Web application



Debug

deploy

Dart

- 오직 Dart 언어만 익히면 됨!!

- java와 90프로 유사.. 자바보다 쉽다. 자바스크립트 비슷(70%)
- 배우기 쉽고, 가장 모던한 언어
- 객체지향 & 함수형 프로그래밍 언어

- Flutter 앱은 Dart로 개발
- 즉, Dart 라이브러리, 객체를 통해서 개발

다트를 사용하는 이유

- 구글이 소유하고 유지보수
- UI에 최적화된 언어
 - 마크업을 사용하지 않고 다트 언어만으로 UI 작성
 - 배우기 쉽고, 사용하기 쉬움

<https://dart.dev/>

플러터 학습 순서

- 개발환경 설정
 - Flutter SDK 설치, IDE 설치
- 다트 언어 학습
 - 공식 다트 문서
- 플러터 기본 이해
 - 위젯, 레이아웃, 상태관리, 네비게이션
 - 네트워킹 API
- 실제 프로젝트 빌드

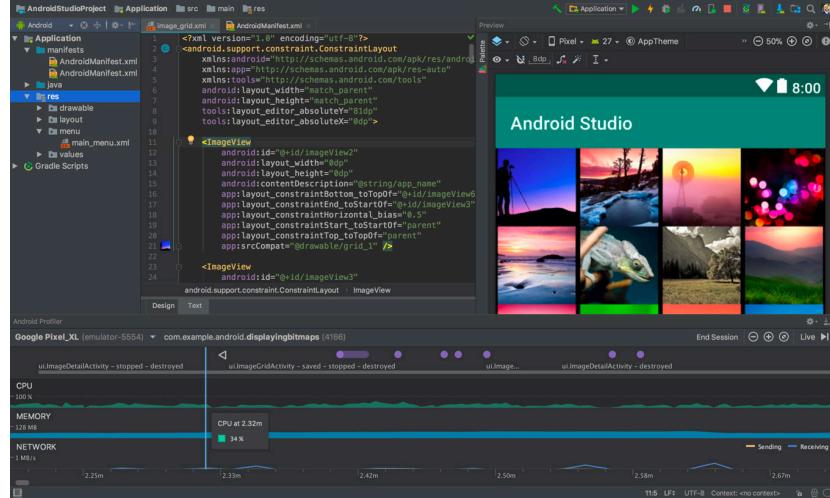
- Flutter 앱 개발 사전준비
- Flutter SDK 설치
- 안드로이드 스튜디오 설치
- 안드로이드 에뮬레이터 설치

2장. 개발을 위한 설치와 셋업

Flutter 앱 개발용 플랫폼

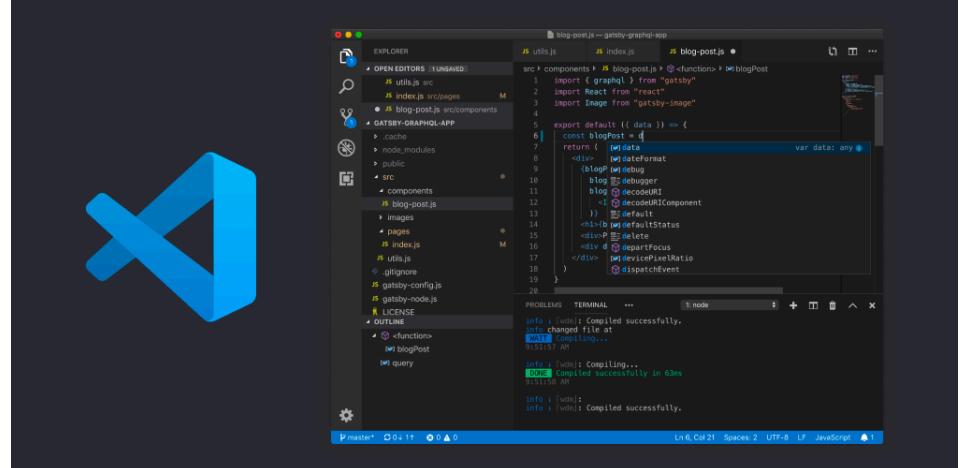


Flutter 개발용 IDE



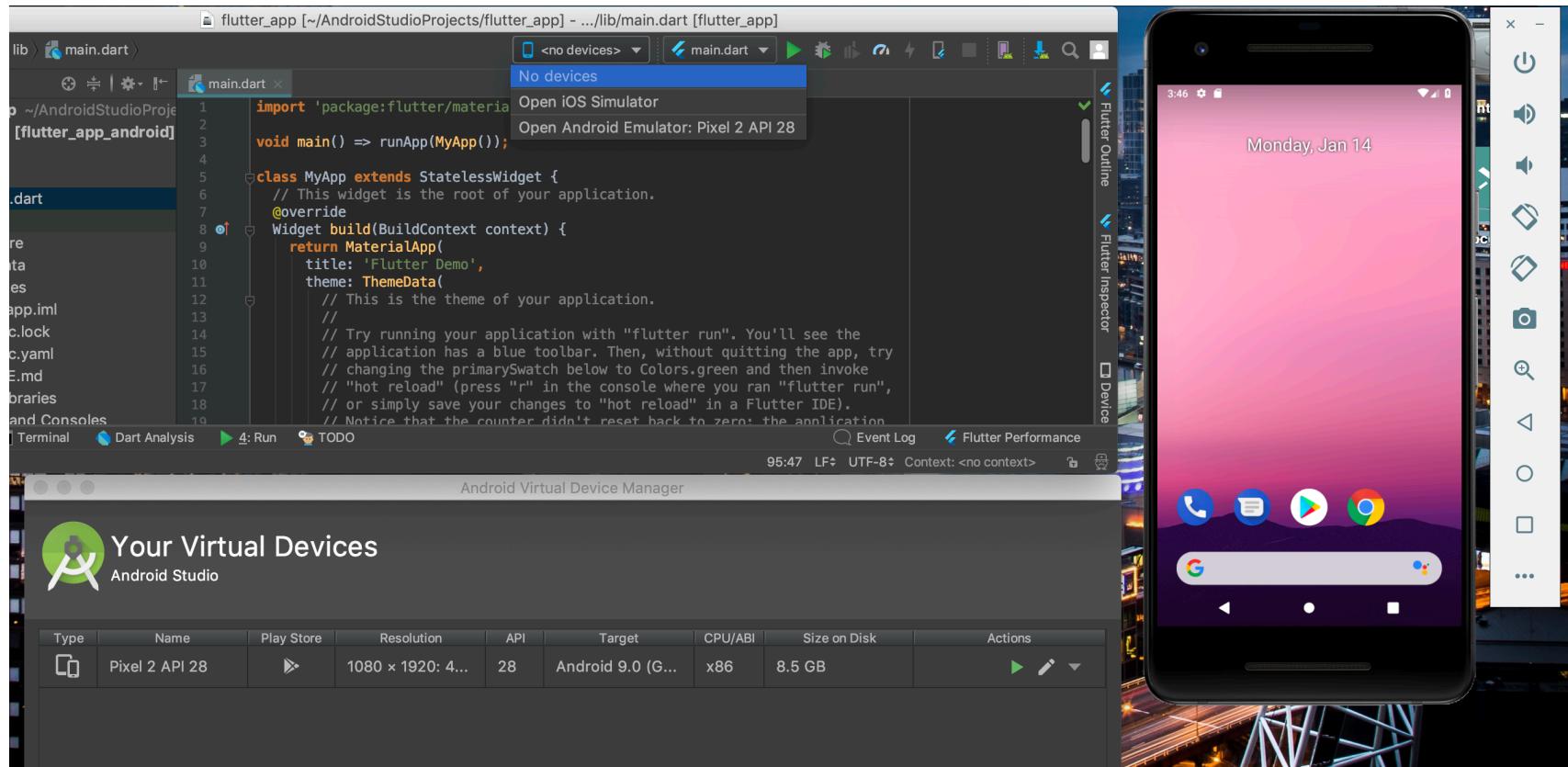
안드로이드 스튜디오

VS Code



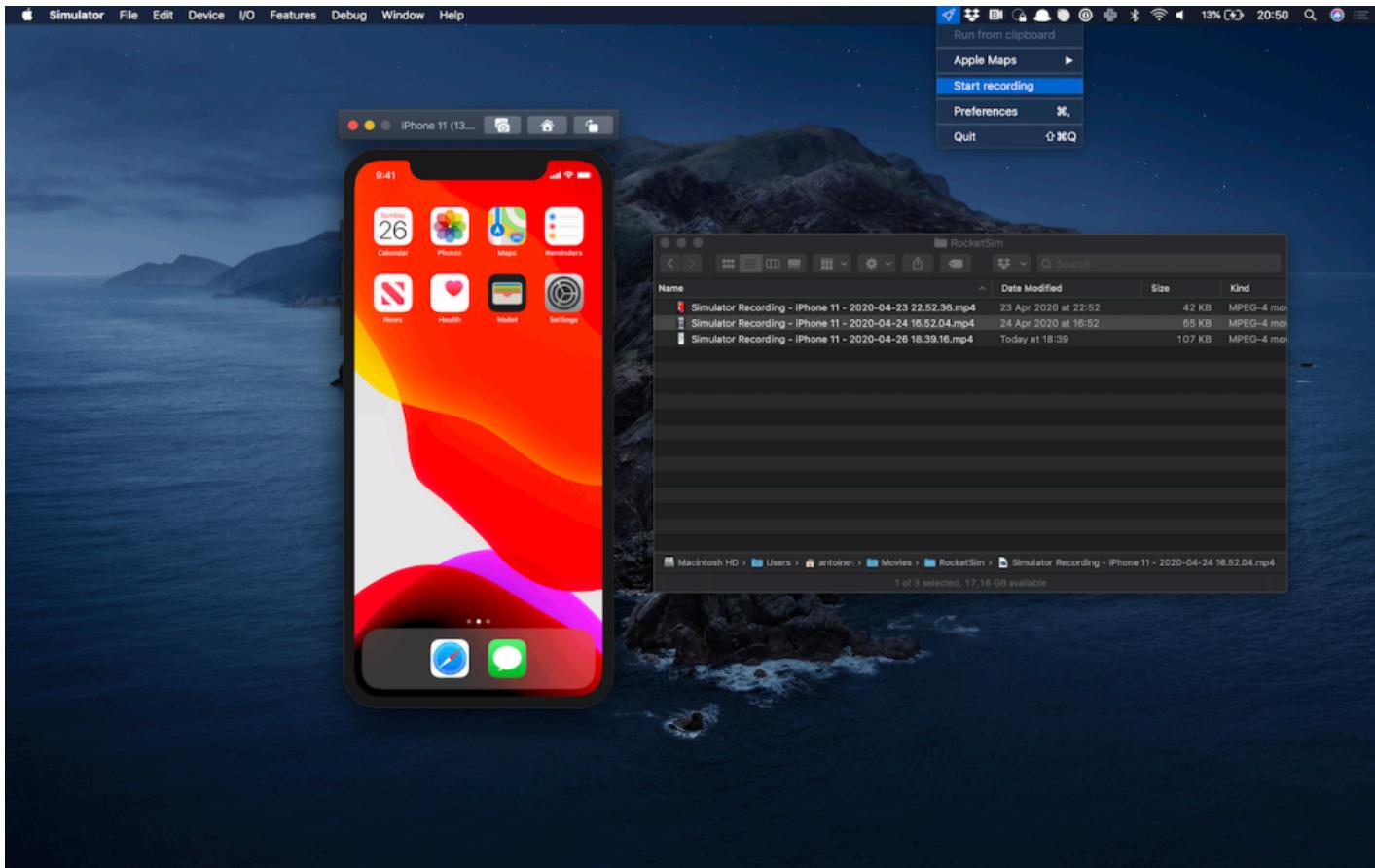
안드로이드 테스트

● 안드로이드 디바이스 or Android Emulator



iOS 테스트

- iPhone 디바이스 or iOS Simulator



Android & iOS

Android Apps

Mac or PC

Android Studio

Android Emulator
or physical device

iOS Apps

Mac

Android Studio

iOS Simulator
or physical device

PC에서 Flutter 설정

- Install the Flutter SDK
- Install VSCode (Android Studio)
- Install the Android Emulator

PC에서 Flutter 설정

● Install the Flutter SDK

<https://docs.flutter.dev/get-started/install/windows>

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

[flutter_windows_3.10.6-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, `%USERPROFILE%\flutter`, `D:\dev\flutter`).

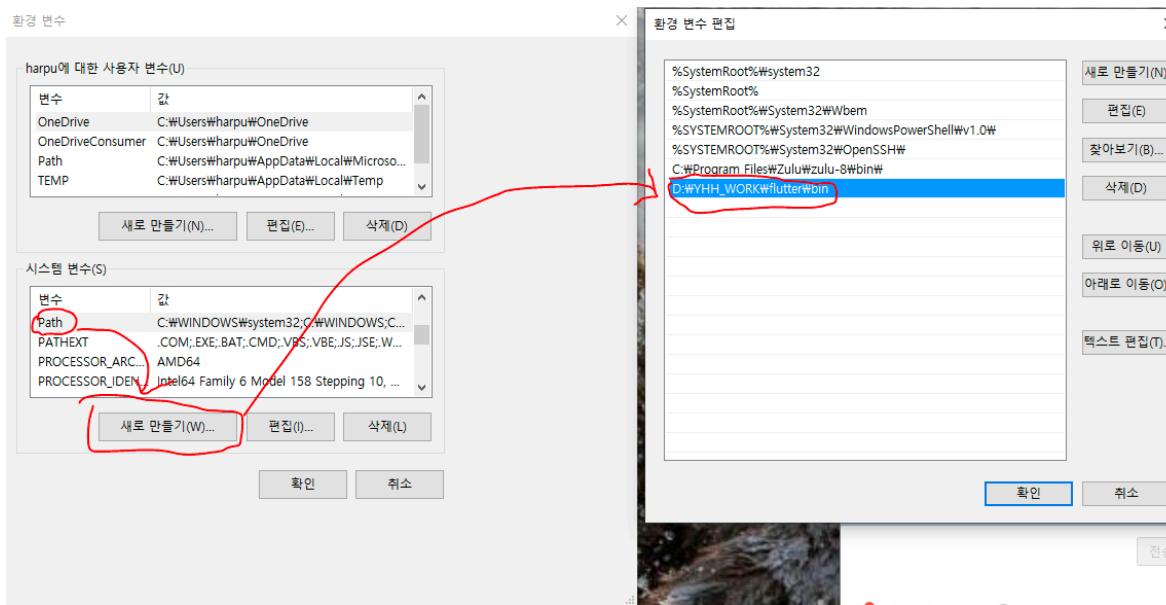
⚠ Warning: Do not install Flutter to a path that contains special characters or spaces.

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

PC에서 Flutter 설정

- PATH 환경변수에 Flutter 추가
- 설치 후 환경변수 추가

```
C:\>where flutter dart
C:\path-to-flutter-sdk\bin\flutter
C:\path-to-flutter-sdk\bin\flutter.bat
C:\path-to-dart-sdk\bin\dart.exe          :: this should go after `C:\path-to-flutter-sdk\bin\
C:\path-to-flutter-sdk\bin\dart
C:\path-to-flutter-sdk\bin\dart.bat
```



PC에서 Flutter 설정

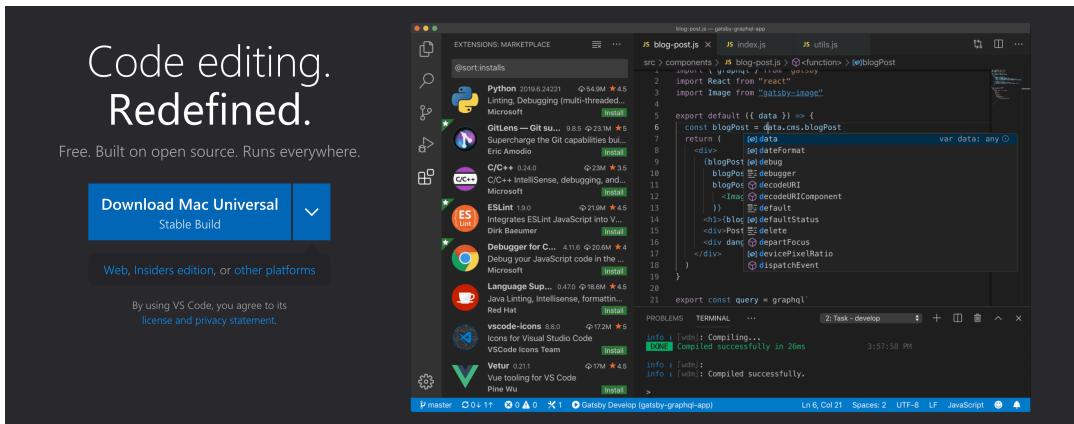
- flutter doctor 실행
 - 명령프롬프트(cmd)에서 flutter doctor 명령 실행

```
C:\src\flutter>flutter doctor
```

```
[-] Android toolchain - develop for Android devices
  • Android SDK at D:\Android\sdk
× Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
  • Try re-installing or updating your Android SDK,
    visit https://docs.flutter.dev/setup/#android-setup for detailed instructions.
```

VSCODE 설치

● VSCODE 에디터 설치

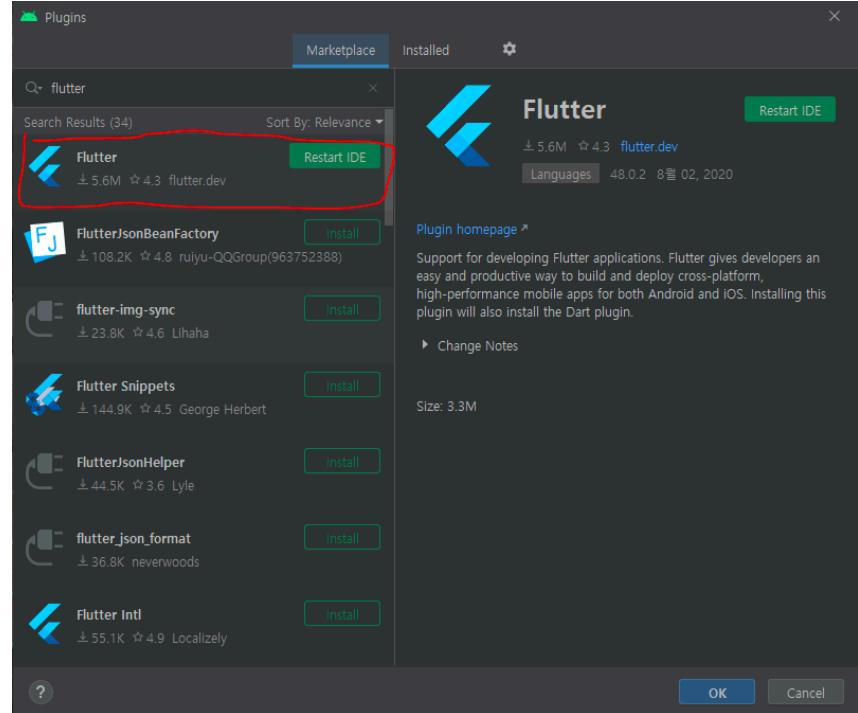
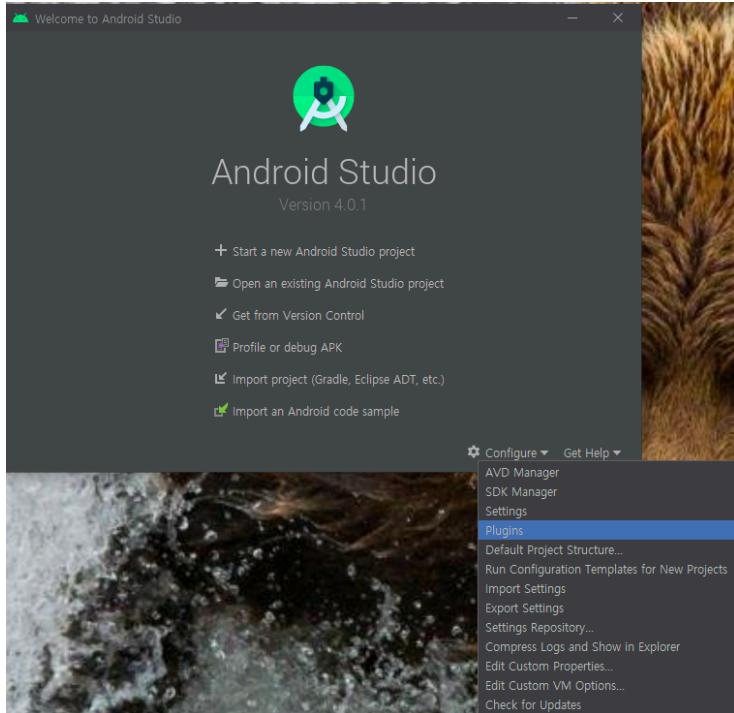


● 플러터 플러그인 설치



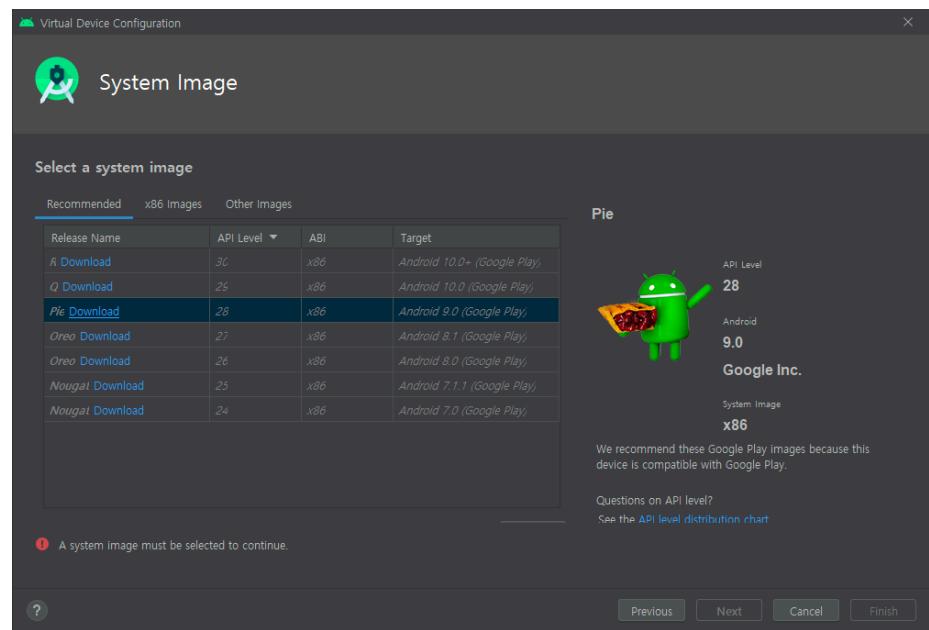
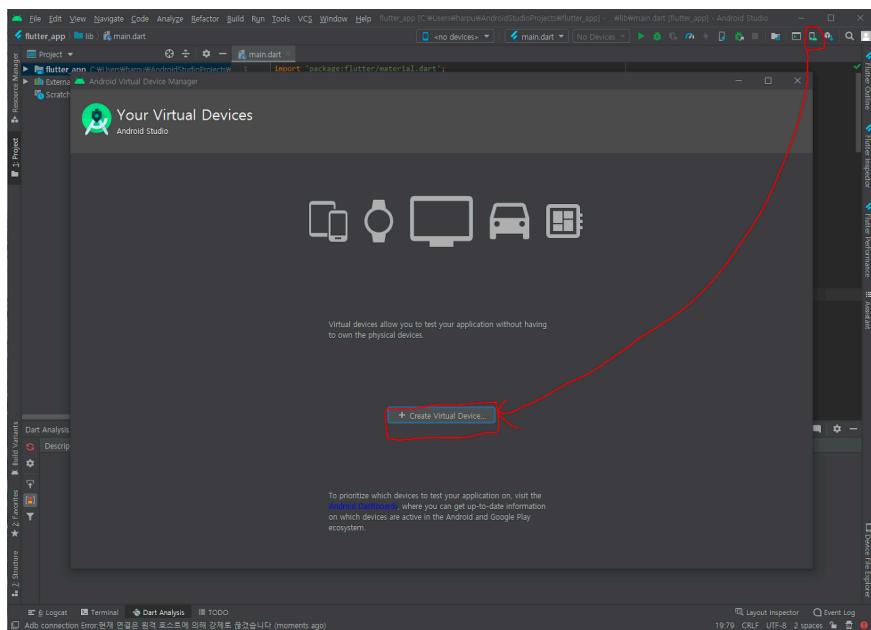
안드로이드 스튜디오

- Install Android Studio
 - Flutter 플러그인 설치



PC에서 Flutter 설정

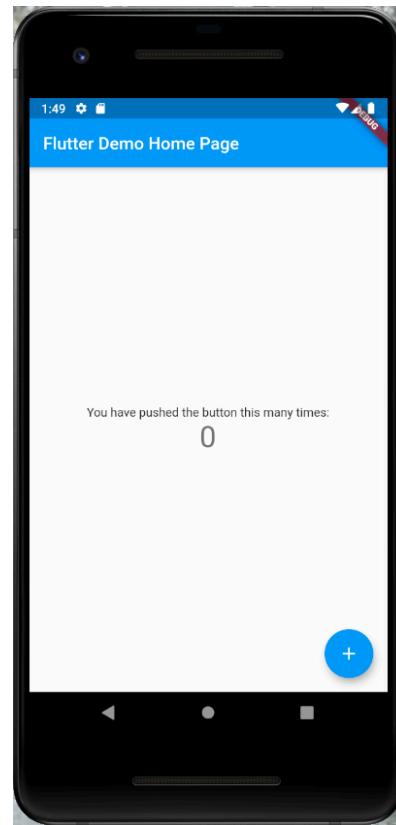
- Install the Android Emulator
 - AVD (Android Virtual Device)



PC에서 Flutter 설정

- 마지막으로 Flutter doctor 실행

- flutter doctor --android-licenses 실행
- 약관에 모두 y 를 입력



Mac에서 Flutter 설정

- Install the Flutter SDK
- Install Android Studio
- Install the Android Emulator
- Install Xcode and command-line tools
- Test the iOS Simulator

- 변수 선언 - var , final , const
- 문자열, String
- 함수 작성법
- 클래스 작성과 생성자
- 제네릭 구현
- 컬렉션 - 데이터 다루기
- 고차함수 - 함수형 프로그래밍
- Null 안전 코드 이해하기

Dart. 다트 언어 살펴보기

DartPad

- Dart 코드를 실행할 수 있는 간단한 웹 앱
- <https://dartpad.dev>

The screenshot shows the DartPad interface. At the top, there's a navigation bar with the DartPad logo, 'New Pad', 'Reset', 'Format', 'Install SDK', and a session ID 'glistening-villa-4646'. Below the bar is a toolbar with a 'Run' button. The main area is divided into two sections: 'Console' on the right and 'Documentation' on the bottom right. The 'Console' section is currently empty. The 'Documentation' section also contains no content. On the left, the code editor displays the following Dart code:

```
1▼ void main() {  
2▼   for (int i = 0; i < 5; i++) {  
3    print('hello ${i + 1}');  
4  }  
5 }  
6
```

At the bottom of the page, there are links for 'Privacy notice', 'Send feedback', 'stable channel', 'no issues', 'Based on Flutter 3.0.5 Dart SDK 2.17.6', and a help icon.

변수 선언

- **var vs. final vs. const**

- var - 모든 변수 타입을 지정가능
- final - 한번 지정된 값을 변경 할 수 없음
 - ❖ 권장되는 방식
 - ❖ 타입 지정은 옵션임
- const - final과 동일하나 ~~타입지정을 할 수 없음~~
 - ❖ 컴파일 타임에 값이 지정됨

문자열, String

- 텍스트를 포함하는 변수
 - 주로 ‘ ‘ 싱글 따옴표 사용
 - “ “ 더블 따옴표도 사용 가능
- 문자열 보간, String interpolation
 - \${}

함수, Functions

- Dart 의 추가적인 기능
 - Optional Parameters
 - ❖ [] 사용
 - Named Optional Parameters
 - ❖ { } 사용

클래스, Class

- 클래스 작성
 - 전통적인 OOP 클래스 작성 방식
- 생성자
 - 빠른 생성자 작성 문법을 지원
- Mixin
 - 클래스를 상속받지 않고 사용 가능하게 해줌

컬렉션, Collections

- Dart 에서 가장 많이 사용하는 컬렉션
 - List
 - ❖ 목록 요소의 순서가 유지되는 선형 컬렉션
 - Map
 - ❖ 키 : 값 의 쌍으로 구성된 비선형 컬렉션
 - Set
 - ❖ 집합은 순서가 유지되지 않은 고유한 값의 비선형 컬렉션

고차함수 - 함수형 프로그래밍

- 데이터 CRUD - 함수형 프로그래밍으로 간편하게..
 - 매핑, Mapping
 - 필터링, Filtering
 - 정렬, Sorting
 - 집계, Reducing

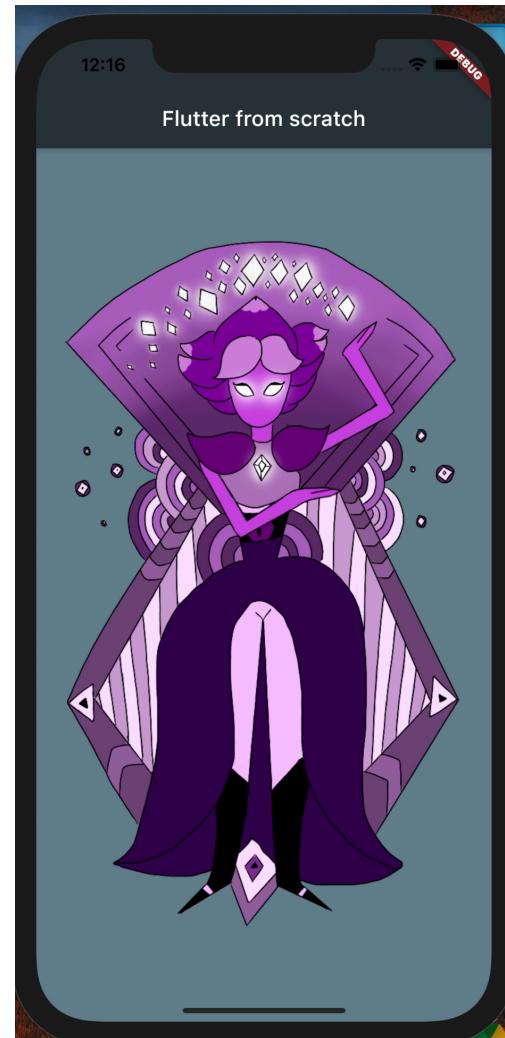
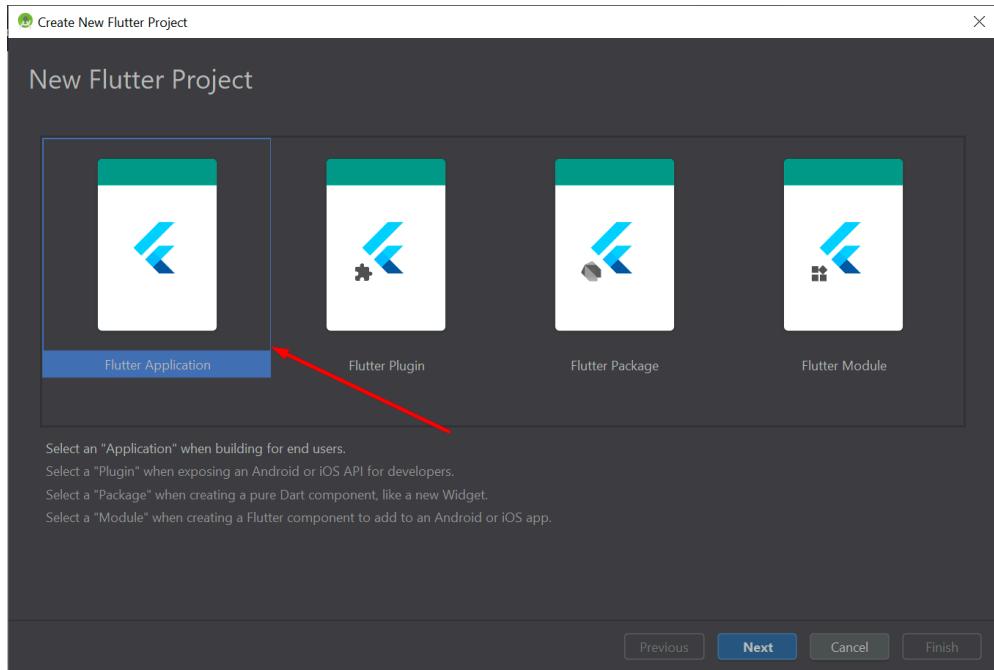
캐스케이드 연산자

- Cascade 연산자 (..)
 - 빌더 패턴을 쉽게 구현할 수 있게 해줌

- Flutter 프로젝트 시작
- StatelessWidget Widget
- MaterialApp, Scaffold
- Asset 작업, 앱에 이미지 추가

3장. Flutter 앱 처음부터 작성하기

Android Studio 프로젝트 생성



Layout System

- 간단하면서 유연하고 아름다운 레이아웃 시스템
 - cf. react, angular, vue.js (HTML + CSS)
 - cf. android native app (XML)
 - Layout + widget

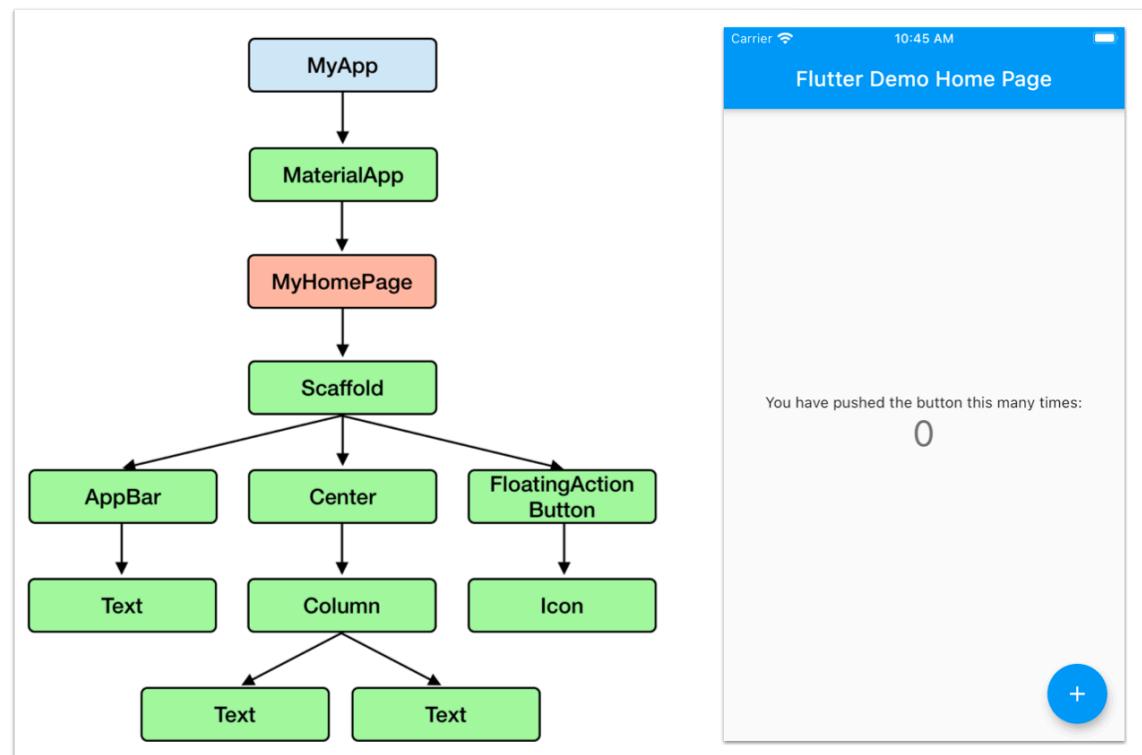
Dart 언어만 사용



모든것이 위젯

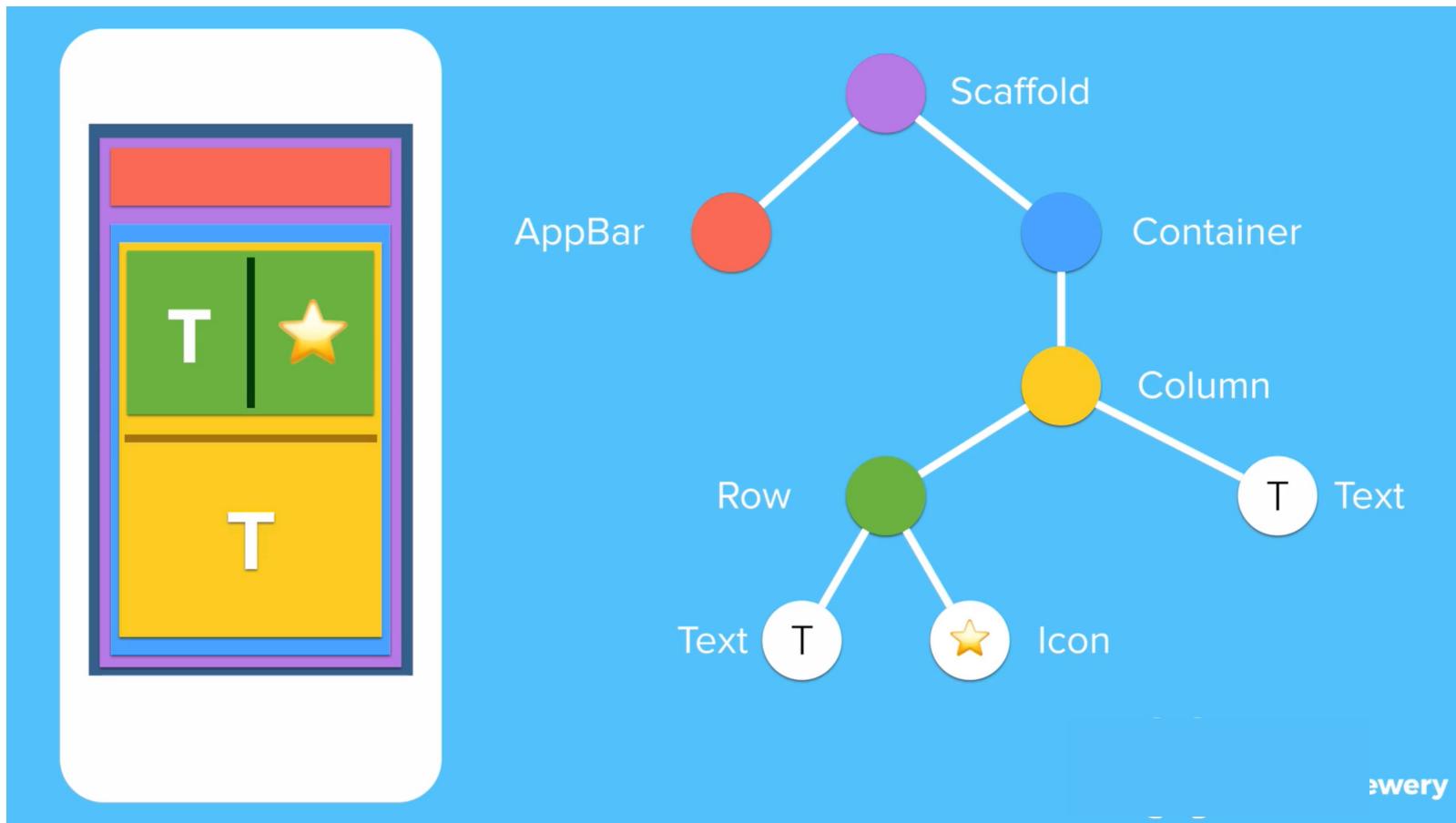
위젯

- 컴포넌트 라고도 함
- 위젯 트리를 구성
- 상속 보다는 합성



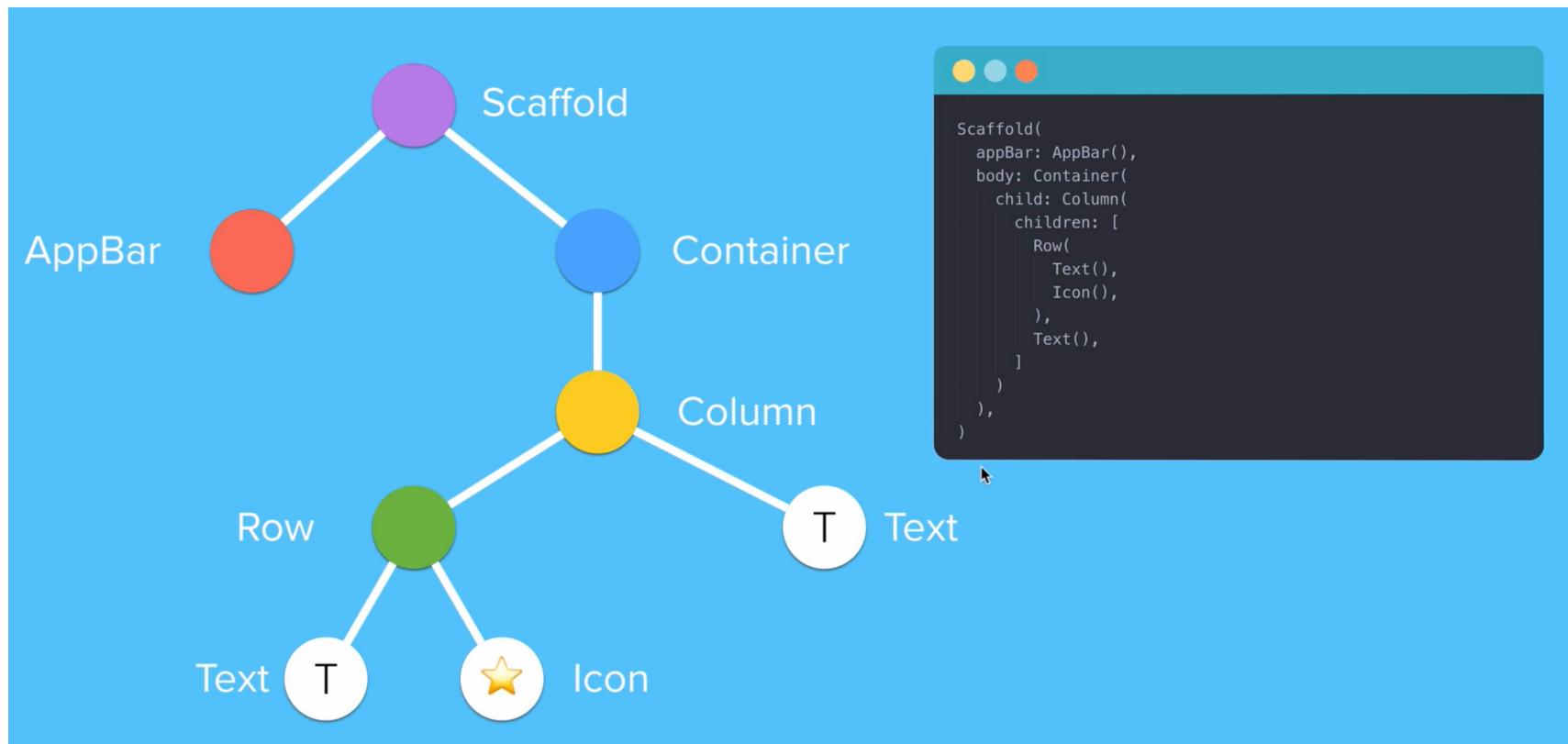
Flutter 앱 해부

- 모든 것이 위젯



Flutter 앱 해부

- 모든 것이 위젯



Material Design 기본

- Material Design

<https://material.io/>



Build beautiful products, faster.

Material is a design system – backed by open-source code – that helps teams build high-quality digital experiences.

GET STARTED

▶ WATCH VIDEO

Flat Design vs. Skeuomorphic vs Material

Scaffold

- Scaffold

Scaffold class

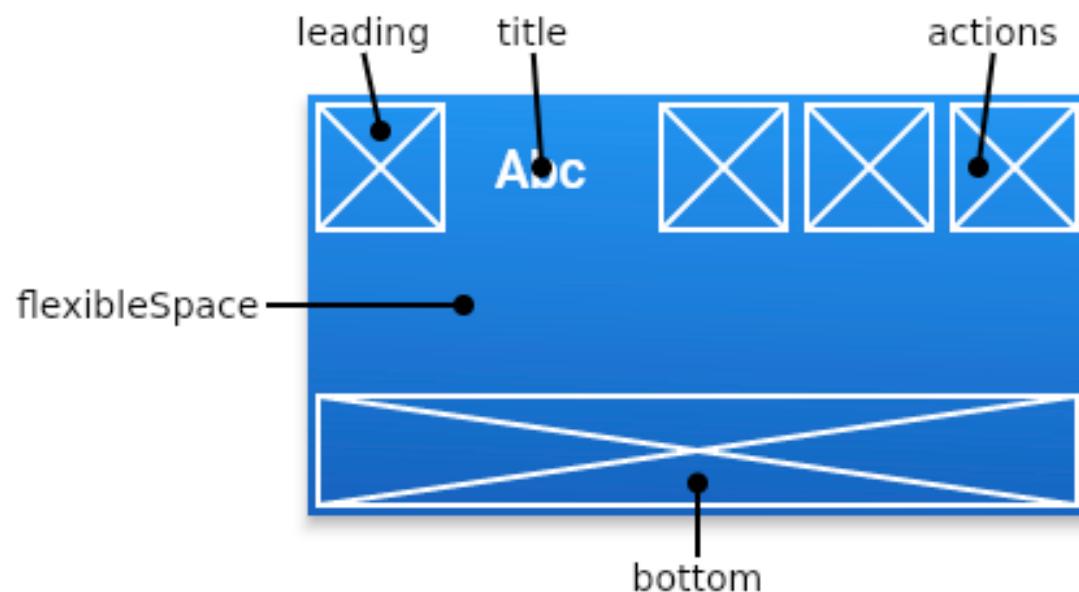
(Null safety)

Implements the basic material design visual layout structure.

This class provides APIs for showing drawers and bottom sheets.

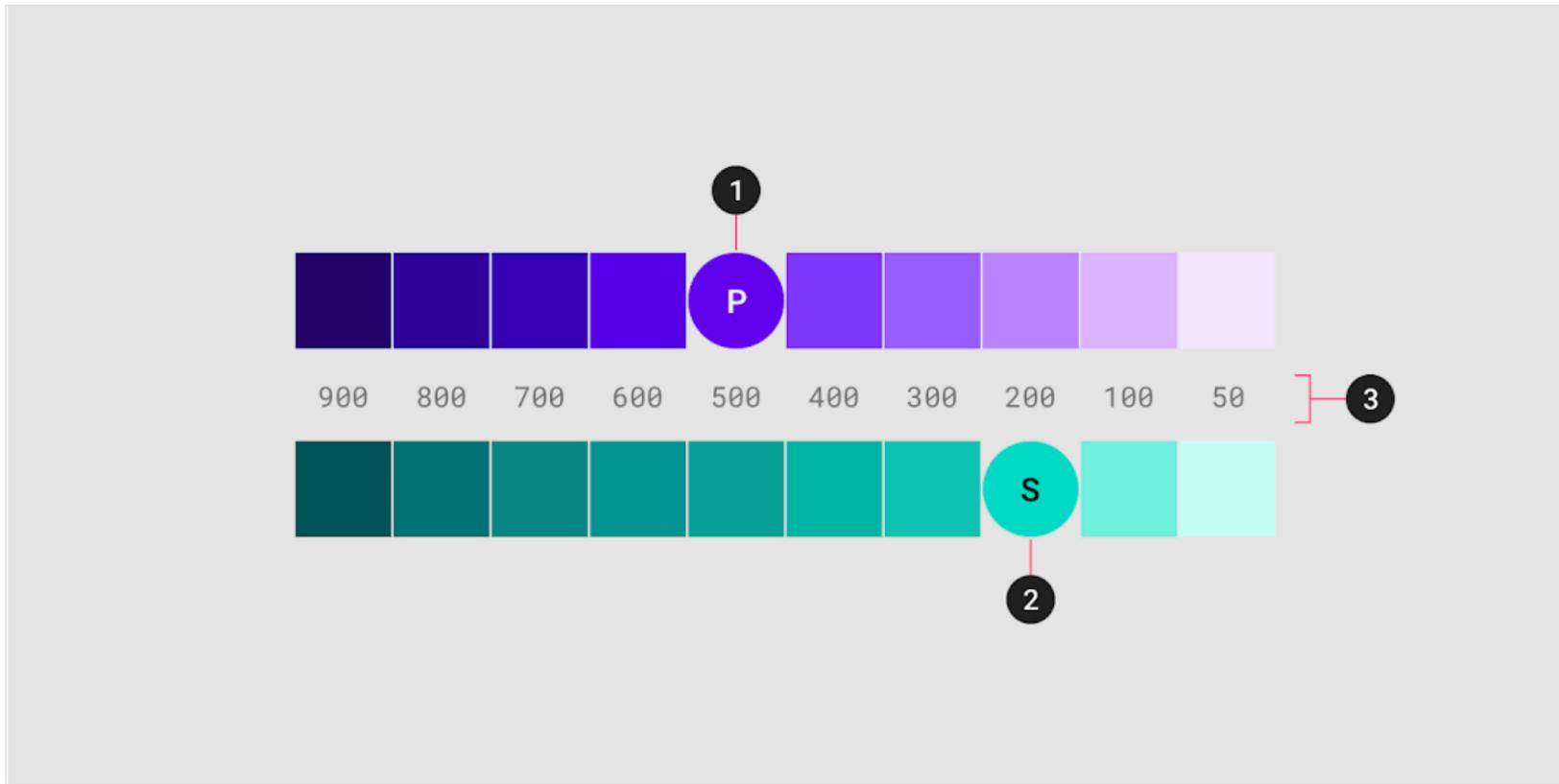
To display a persistent bottom sheet, obtain the `ScaffoldState` for the current `BuildContext` via `Scaffold.of` and use the `ScaffoldState.showBottomSheet` function.

- appBar



Material Color System

- The color system



<https://material.io/design/color/the-color-system.html#color-usage-and-palettes>

Image 위젯

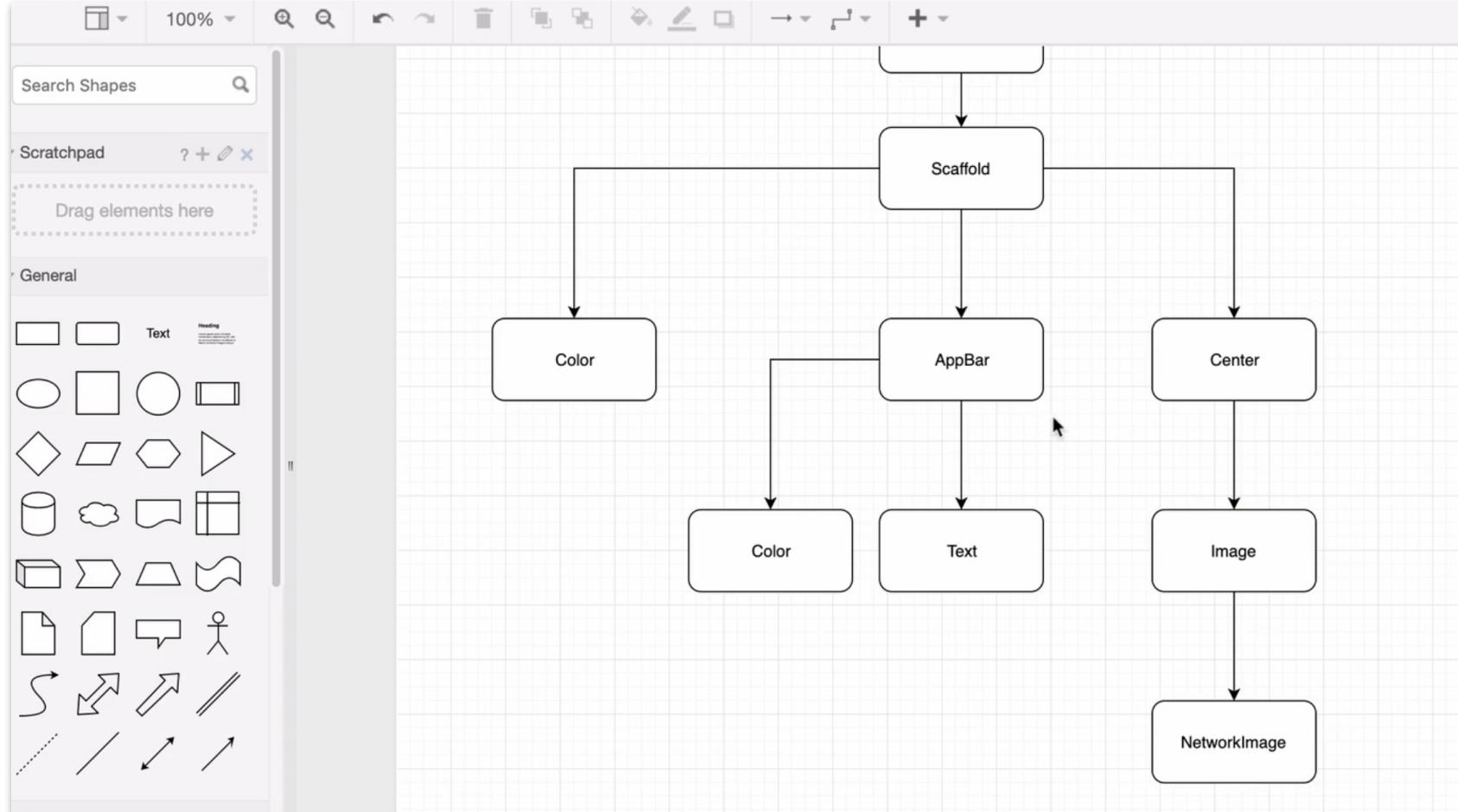
Image class

Null safety

- [new Image](#), for obtaining an image from an [ImageProvider](#).
- [new Image.asset](#), for obtaining an image from an [AssetBundle](#) using a key.
- [new Image.network](#), for obtaining an image from a URL.
- [new Image.file](#), for obtaining an image from a [File](#).
- [new Image.memory](#), for obtaining an image from a [Uint8List](#).

<https://api.flutter.dev/flutter/widgets/Image-class.html>

draw.io



Asset 추가하기

- pubspec.yaml 설정파일
 - 이미지 파일 추가 가능

The screenshot shows the Android Studio interface with the project 'i_am_rich' open. The left sidebar displays the project structure, including 'images' (containing 'diamond.png'), 'lib', 'test', and 'pubspec.yaml'. The main editor window shows the content of the 'pubspec.yaml' file:

```
31  # The following section is specific to Flutter.
32  flutter:
33    # The following line ensures that the Material Icons font is
34    # included with your application, so that you can use the icons in
35    # the material Icons class.
36    uses-material-design: true
37
38    # To add assets to your application, add an assets section, like this:
39    assets:
40      - images/diamond.png
41
42      # An image asset can refer to one or more resolution-specific "variants", see
43      # https://flutter.io/assets-and-images/#resolution-aware.
44
45      # For details regarding adding assets from package dependencies, see
46      # https://flutter.io/assets-and-images/#from-packages
47
48      # To add custom fonts to your application, add a fonts section here,
49      # in this "flutter" section. Each entry in this list should have a
50      # "family" key with the font family name, and a "fonts" key with a
51      # list giving the asset and other descriptors for the font. For
52      # example:
53      # fonts:
```

앱 아이콘 추가

<https://appicon.co/>

 App Icon Generator

 App Icon

 Image Sets

 Desktop App

NEW

 Donate



Click or drag image file (1024 x 1024)

App icon Generator

Drag or select an app icon image (1024x1024) to generate different app icon sizes for all platforms

iOS and macOS

- iPhone - 11 different sizes and files
- iPad - 13 different sizes and files
- watchOS - 8 different sizes and files
- macOS - 11 different sizes and files

Android

- Android - 4 different sizes and files

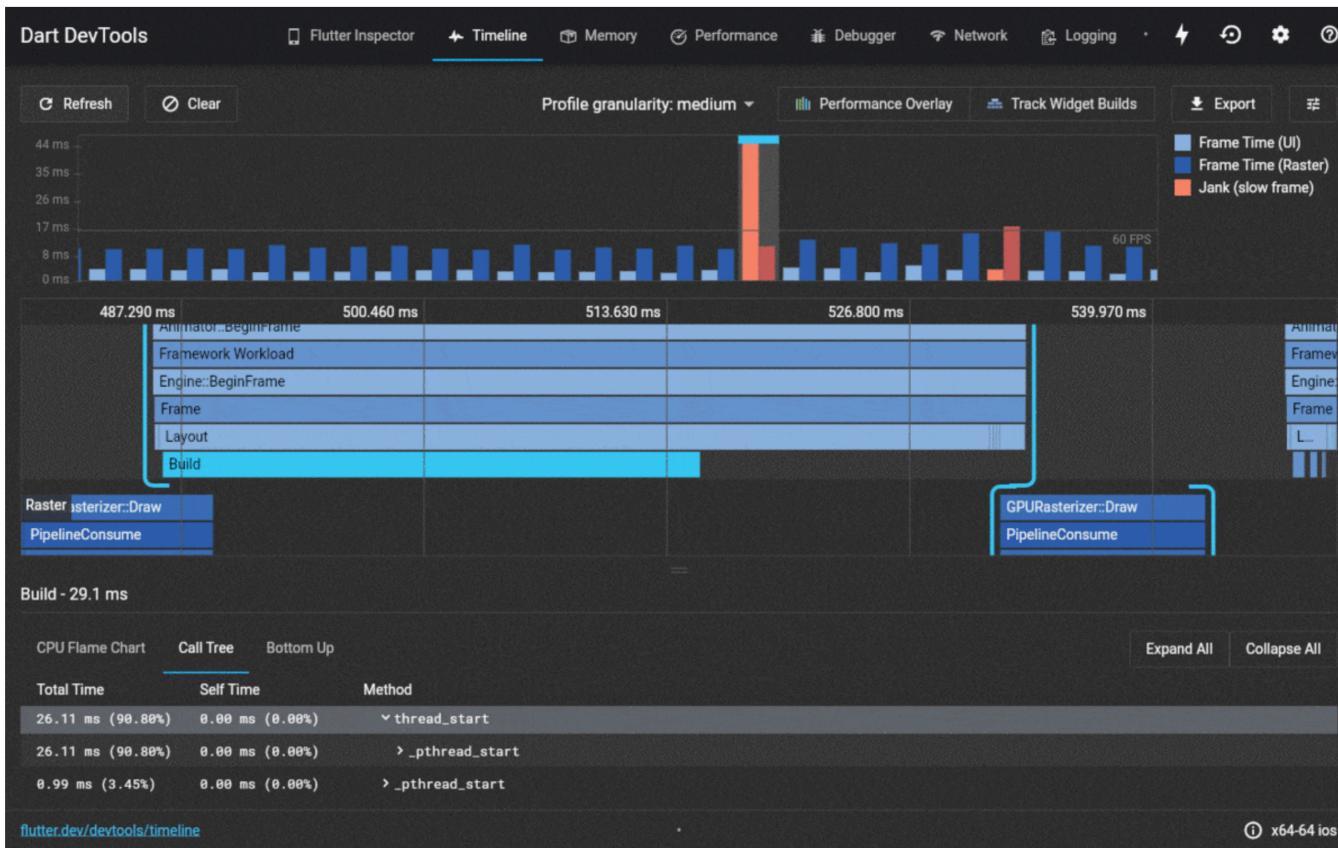
File name

Change file name for all generated Android images

 Generate

플러터 DevTools

- Dart 와 Flutter 의 성능과 디버깅을 위한 툴
 - <https://docs.flutter.dev/development/tools/devtools/overview>

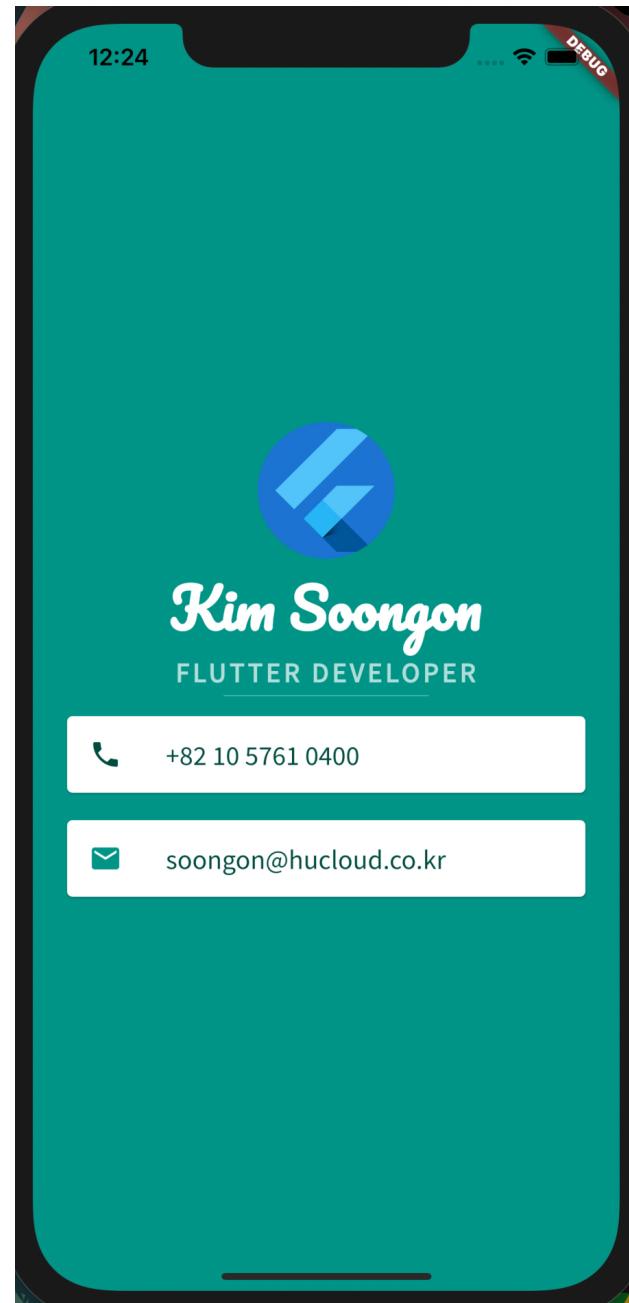


- Single Screen MiCard App
- 레이아웃 위젯 - Container, Row, Column
- Custom Font, Icon 추가
- Card & ListTile 위젯 사용

4장. 위젯 사용하여 멋진 UI 만들기

<https://flutter.dev/docs/development/ui/layout/tutorial>

단일 화면 레이아웃



프로젝트 시작

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

void main() {
    runApp(
        MaterialApp(
            home: Scaffold(
                backgroundColor: Colors.teal,
                body: Container(),
            ),
        ),
    );
}
```

위젯과 Layout

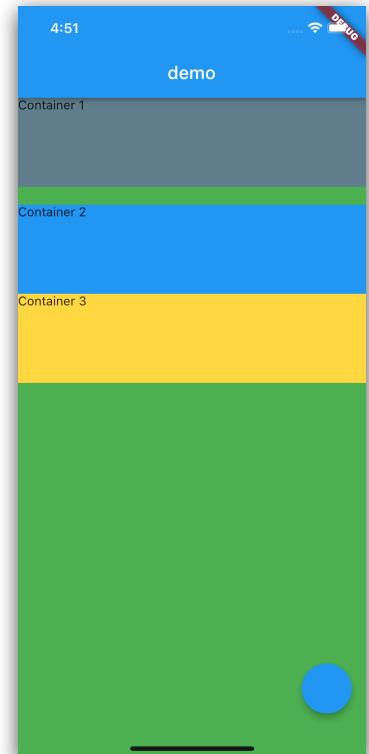
- Single-child 레이아웃 위젯

- Container



- Multi-child 레이아웃 위젯

- Column
 - Row



<https://flutter-ko.dev/docs/development/ui/widgets/layout>

Layout Cheat Sheet

Flutter Layout Cheat Sheet



Tomek Polański [Follow](#)

May 3, 2018 · 13 min read



<https://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037e>

구글 폰트 사용

Google Fonts

Browse fonts Icons More ⚙️

Search Sentence Type something 40px

Categories Language Font properties Show only variable fonts

1052 of 1052 families Sort by: Trending

<p>Roboto Christian Robertson</p> <p>12 styles</p> <p>Almost before we knew it, we had left the ground.</p>	<p>Noto Sans KR Google</p> <p>6 styles</p> <p>그들의 장비와 기구는 모두 살아 있다.</p>	<p>Antonio Vernon Adams</p> <p>Variable</p> <p>Almost before we knew it, we had left the ground.</p>	<p>Orelega One Haruki Wakamatsu</p> <p>1 style</p> <p>Almost before we knew it, we had left the ground.</p>
---	---	--	---

<https://fonts.google.com/>

사용자 폰트 사용

- 다운로드한 폰트 파일 추가

- 주로 assets/fonts 폴더

<https://flutter.dev/docs/cookbook/design/package-fonts>

- pubspec.yaml 파일에 폰트 선언 추가

```
flutter:  
  fonts:  
    - family: Raleway  
      fonts:  
        - asset: packages/awesome_package/fonts/Raleway-Regular.ttf  
        - asset: packages/awesome_package/fonts/Raleway-Italic.ttf
```

- 폰트 사용

```
Text(  
  'Using the Raleway font from the awesome_package',  
  style: TextStyle(  
    fontFamily: 'Raleway',  
    package: 'awesome_package',  
  ),  
) ;
```

Icon 사용하기

Icons class Null safety

<https://api.flutter.dev/flutter/material/Icons-class.html>

<https://fonts.google.com/icons?selected=Material+Icons>

material design palette

<https://www.materialpalette.com/pink/light-blue>

Card 위젯

Card class

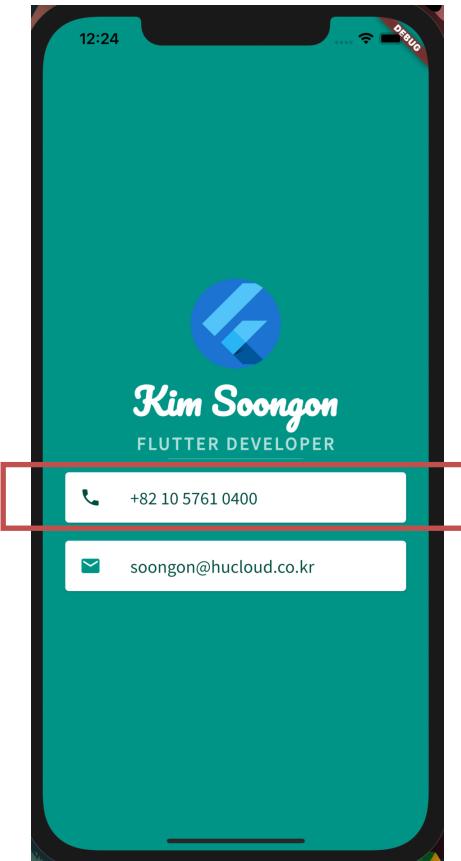
Null safety



The Enchanted Nightingale

Music by Julie Gable. Lyrics by Sidney Stein.

BUY TICKETS LISTEN



<https://api.flutter.dev/flutter/material/Card-class.html>

Padding 위젯



```
EdgeInsets.all(24)  
EdgeInsets.only(right:80)  
EdgeInsets.symmetric(vertical:48.5)
```

<https://api.flutter.dev/flutter/widgets/Padding-class.html>

ListTile 클래스

ListTile class

One-line ListTile



One-line with leading widget

One-line with trailing widget

⋮



One-line with both widgets

⋮

One-line dense ListTile



Two-line ListTile

Here is a second line

⋮



Three-line ListTile

A sufficiently long subtitle warrants
three lines.

⋮

<https://api.flutter.dev/flutter/material/ListTile-class.html>

Divider 위젯

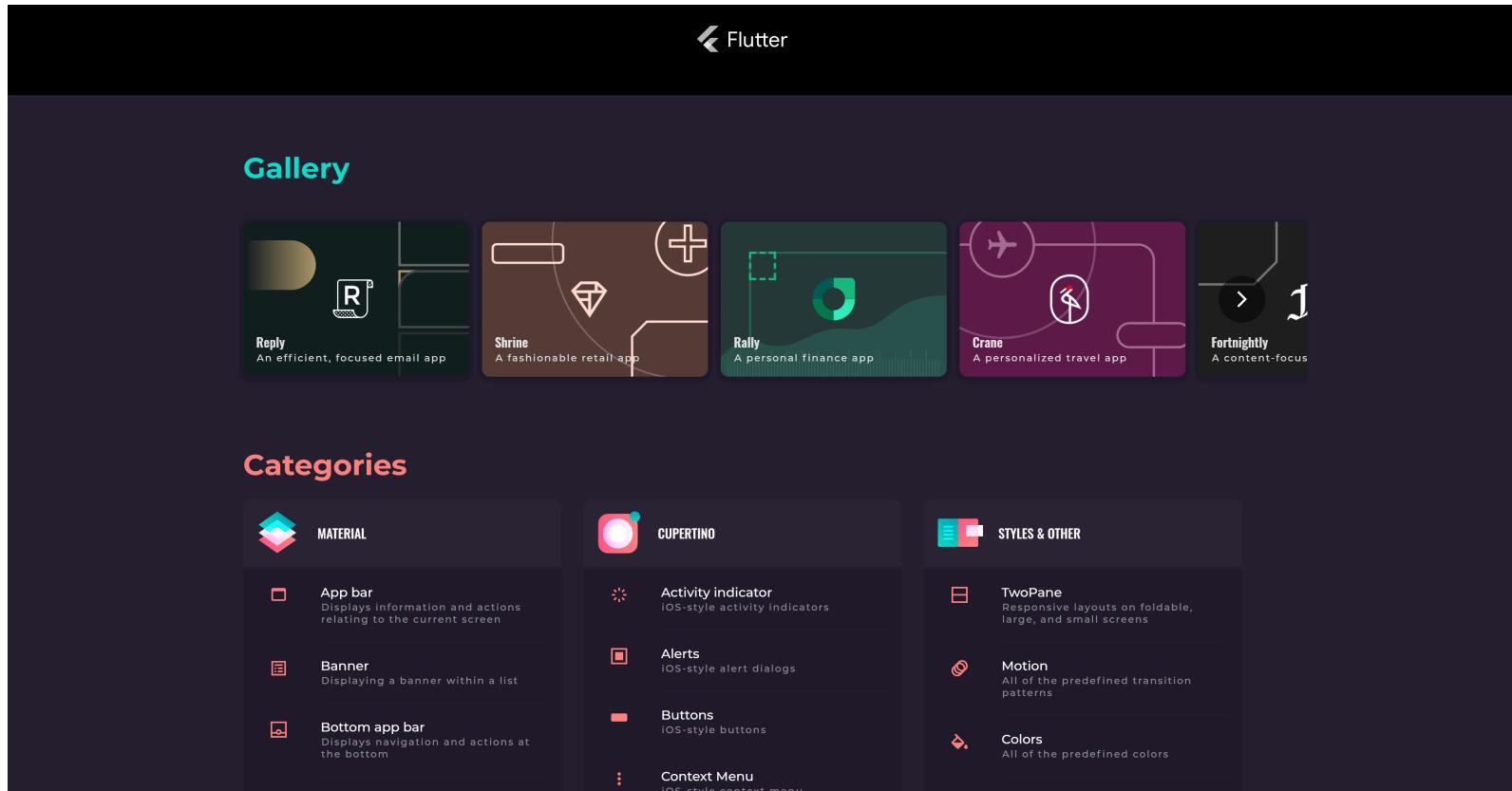
Divider class Null safety



<https://api.flutter.dev/flutter/material/Divider-class.html>

참고

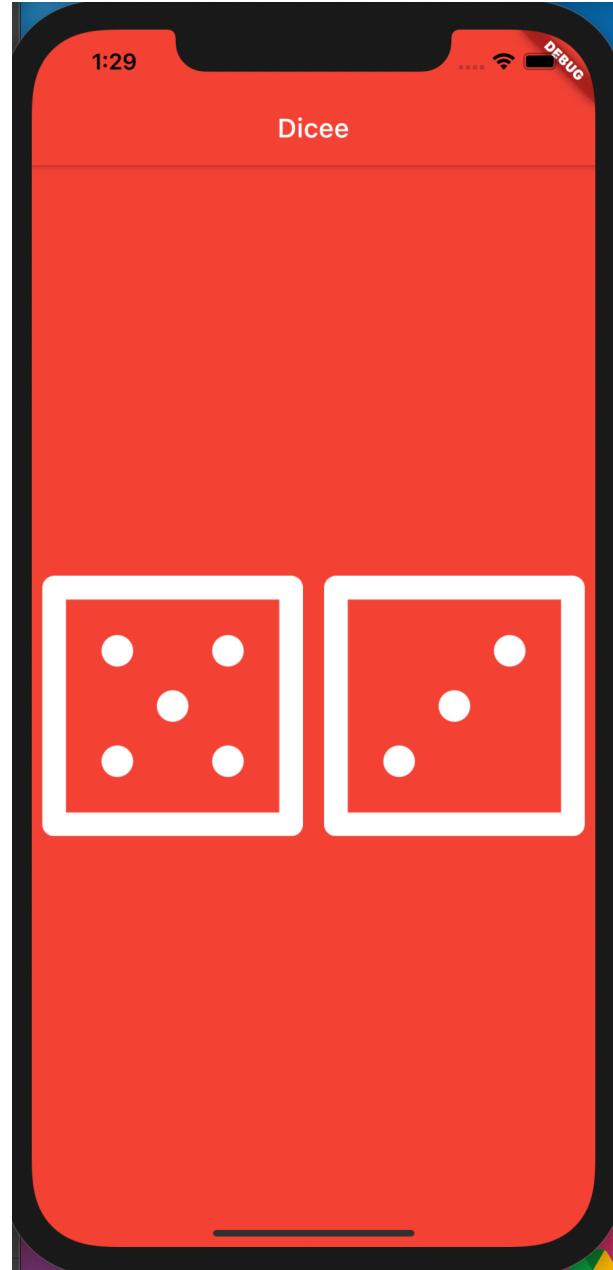
- Flutter Gallery - <https://gallery.flutter.dev>



- 사용자 상호작용 - 이벤트 처리
- Dart 문법 - 변수, 데이터 타입, 함수
- State가 바뀌면 위젯이 리빌드 된다!
- Random 클래스

5장. 상태있는 위젯. Stateful Widget

상태(state) 있는 위젯
Stateful widget



프로젝트 시작

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

void main() {
    return runApp(
        MaterialApp(
            home: Scaffold(
                backgroundColor: Colors.red,
                appBar: AppBar(
                    title: Text('Dicee'),
                    backgroundColor: Colors.red,
                ),
                body: DicePage(),
            ),
        ),
    );
}

class DicePage extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Container();
    }
}
```

Expanded 위젯

- Row, Column 의 자식으로 사용

- 전체를 꽉 채울 때 사용
- Flex로 비율 설정

Expanded Row Sample



Expanded Column Sample

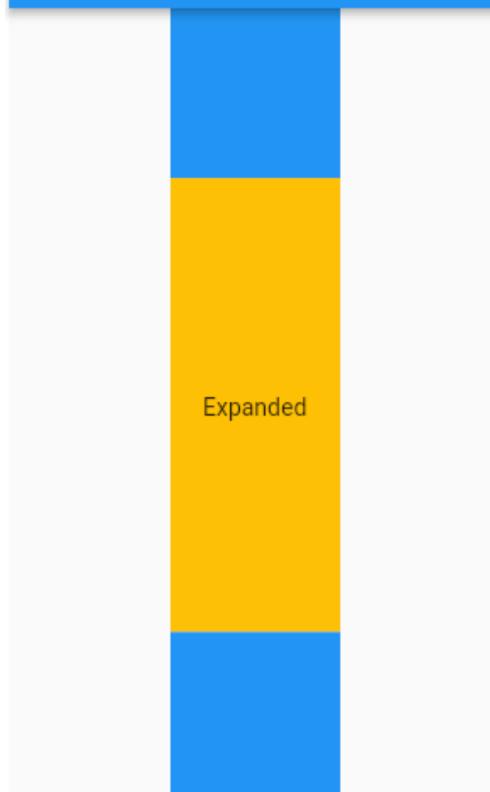


Image 위젯

- **Image**

```
const Image(  
    image: NetworkImage('https://flutter.github.io...owl.jpg'),  
)
```

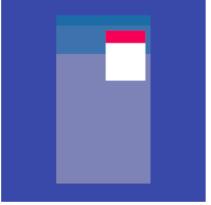
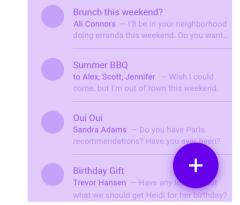
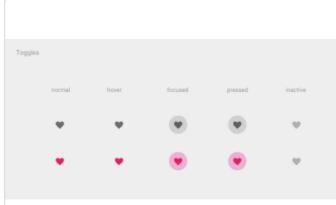
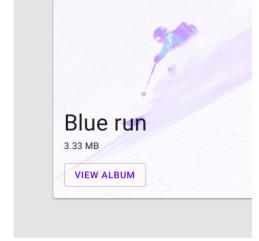
- **Image.asset**

- **Image.network**

- **Image.file**

Button 위젯

Buttons

 <p>DropdownButton Shows the currently selected item and an arrow that opens a menu for selecting another item.</p> <p>Documentation</p>	 <p>ElevatedButton A Material Design elevated button. A filled button whose material elevates when pressed.</p> <p>Documentation</p>	 <p>FloatingActionButton A floating action button is a circular icon button that hovers over content to promote a primary action in the application. Floating action buttons are most commonly used in the Scaffold.floatingActionButton field.</p> <p>Documentation</p>	 <p>Toggles</p> <table border="1"><thead><tr><th>normal</th><th>hover</th><th>focused</th><th>pressed</th><th>inactive</th></tr></thead><tbody><tr><td>♥</td><td>♥</td><td>●</td><td>●</td><td>♥</td></tr><tr><td>♥</td><td>♥</td><td>●</td><td>●</td><td>♥</td></tr></tbody></table>	normal	hover	focused	pressed	inactive	♥	♥	●	●	♥	♥	♥	●	●	♥	 <p>IconButton An icon button is a picture printed on a Material widget that reacts to touches by filling with color (ink).</p> <p>Documentation</p>	 <p>OutlinedButton A Material Design outlined button, essentially a TextButton with an outlined border.</p> <p>Documentation</p>
normal	hover	focused	pressed	inactive																
♥	♥	●	●	♥																
♥	♥	●	●	♥																

<https://flutter.dev/docs/development/ui/widgets/material>

이벤트 처리

● Gesture detector

```
// The GestureDetector wraps the button.  
GestureDetector(  
    // When the child is tapped, show a snackbar.  
    onTap: () {  
        final snackBar = SnackBar(content: Text("Tap"));  
  
        ScaffoldMessenger.of(context).showSnackBar(snackBar);  
    },  
    // The custom button  
    child: Container(  
        padding: EdgeInsets.all(12.0),  
        decoration: BoxDecoration(  
            color: Theme.of(context).buttonColor,  
            borderRadius: BorderRadius.circular(8.0),  
        ),  
        child: Text('My Button'),  
    ),  
);
```

<https://flutter.dev/docs/cookbook/gestures/handling-taps>

Dart pad

- Dart 코드를 온라인 상에서 실행 가능

The screenshot shows the DartPad interface. On the left, there is a code editor with the following Dart code:

```
1 void main() {  
2   for (int i = 0; i < 5; i++) {  
3     print('hello ${i + 1}');  
4   }  
5 }  
6
```

Next to the code editor is a blue "RUN" button. To the right of the code editor is a "Console" panel which displays the output of the executed code:

```
hello 1  
hello 2  
hello 3  
hello 4  
hello 5
```

Below the code editor and console are two smaller panels: "Documentation" and "Samples". At the bottom of the interface, there are links for "Privacy notice", "Send feedback", and "Null Safety" (which is currently turned off). On the far right, it says "no issues Based on Flutter 2.0.5 Dart SDK 2.12.3".

<https://dartpad.dartlang.org/>

Dart 문법

- Function 사용
- 변수와 데이터 타입

Stateful vs. Stateless Widget

- Stateful 위젯

- 상태(데이터)를 포함하고 있음.
- 상태가 변경되면 위젯이 새롭게 빌드
 - ❖ `setState` 를 사용하여 변경해야 함

<https://flutter.dev/docs/development/ui/interactive>

dart:math 라이브러리

- Random 클래스

- 랜덤한 값을 생성하기 위한 클래스

Methods

`nextBool() → bool`

Generates a random boolean value.

`nextDouble() → double`

Generates a non-negative random floating point value uniformly distributed in the range from 0.0, inclusive, to 1.0, exclusive.

`nextInt(int max) → int`

Generates a non-negative random integer uniformly distributed in the range from 0, inclusive, to `max`, exclusive. [\[...\]](#)

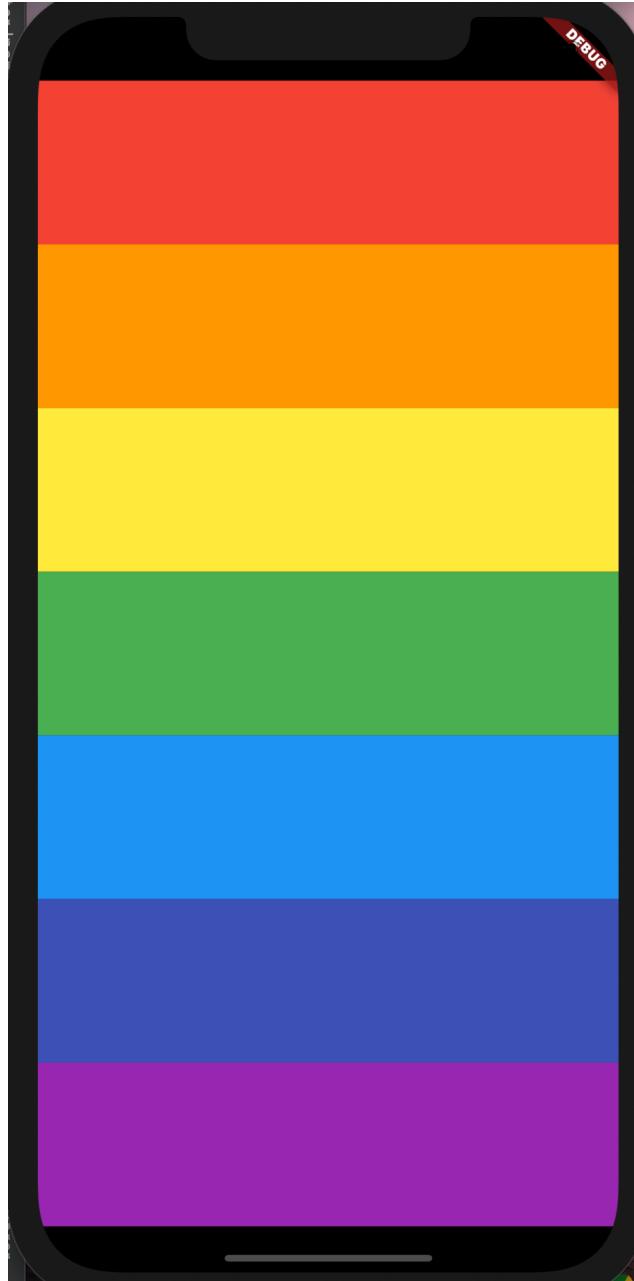
<https://api.dart.dev/stable/2.2.0/dart-math/dart-math-library.html>

- Dart 패키지를 사용하여 개발 속도 높이기
- Audioplayers 패키지
- Dart 문법 :: 함수 사용하기

6장. Xylophone - 패키지 사용

Dart 패키지 사용

- Audioplayers



프로젝트 시작

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

void main() => runApp(XylophoneApp());

class XylophoneApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            home: Scaffold(
                body: SafeArea(
                    child: Container(),
                ),
            ),
        );
    }
}
```

패키지 사용하기

- pub.dev :
 - dart 또는 Flutter 패키지 레파지토리
- 패키지 vs. 플러그인
 - 플러그인은 네이티브 시스템에 접근하는 패키지

<https://flutter.dev/docs/development/packages-and-plugins/using-packages>

Audioplayers 패키지

- Installing

```
dependencies:
```

```
  audioplayers: ^0.18.3
```

```
import 'package:audioplayers/audioplayers.dart';
```

- 로컬 assets 파일을 사용할 때

- AudioCache 클래스 사용

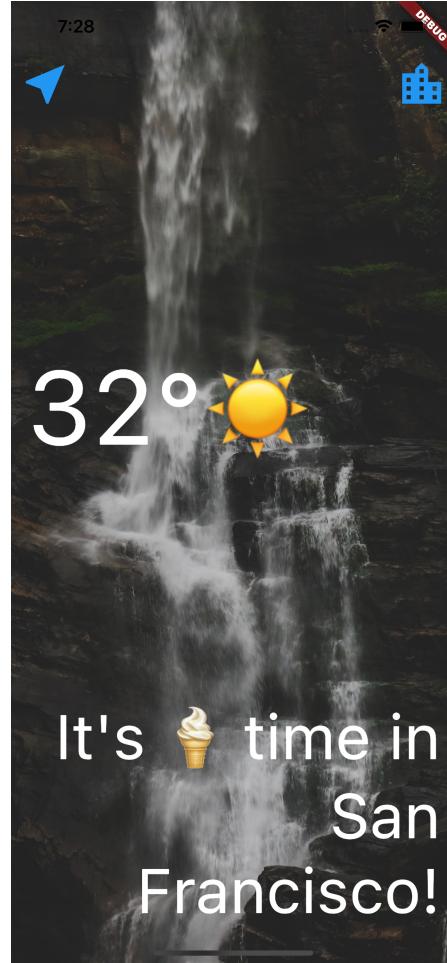
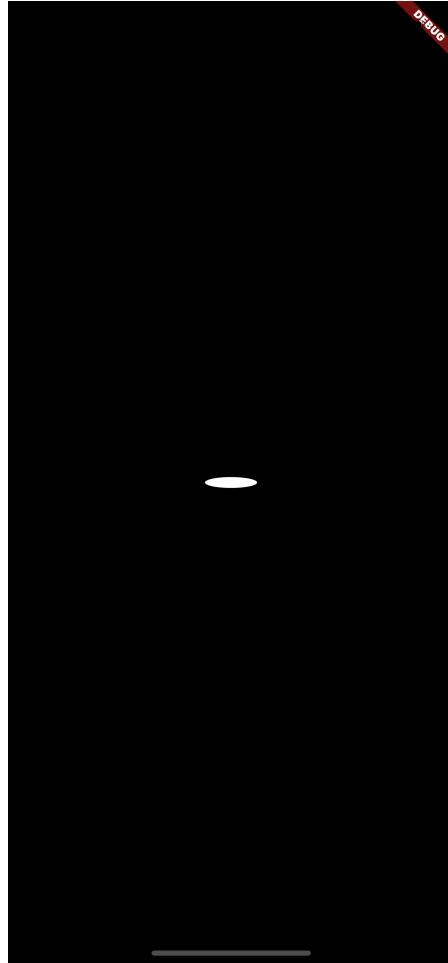
https://pub.dev/documentation/audioplayers/latest/audio_cache/AudioCache-class.html

<https://pub.dev/packages/audioplayers>

- Location
- Dart 문법 - Future, Async&Await, 예외처리, Null safety
- Stateful 위젯 라이프 사이클 메소드
- Network : HTTP 패키지 사용 - API 호출
- JSON 파싱과 다이나믹 타입
- Routes & 네비게이션
- 네비게이션 스택

7장. HTTP로 데이터 가져오기 - 날씨 앱 작성

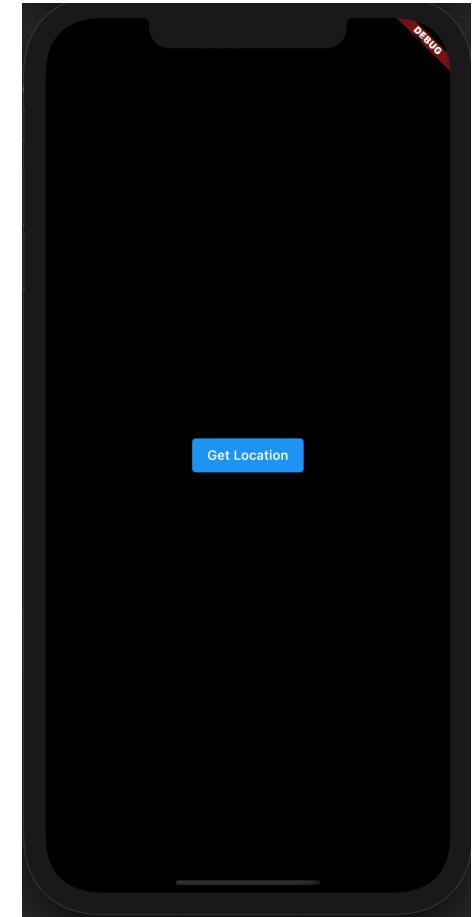
날씨 앱 작성



시작하기

```
class _LoadingScreenState extends State<LoadingScreen> {  
    @override  
    Widget build(BuildContext context) {  
        return Center(  
            child: ElevatedButton(  
                onPressed: () {},  
                child: Text('Get Location'),  
            ),  
        );  
    }  
}  
  
screens/loading_screen.dart
```

```
void main() {  
    runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            theme: ThemeData.dark(),  
            home: LoadingScreen(),  
        );  
    }  
}  
  
main.dart
```



위치 데이터 가져오기

geolocator 7.0.3

Published Apr 14, 2021 •  [baseflow.com](#) Null safety

 [FLUTTER](#)  [ANDROID](#)  [IOS](#)  [WEB](#)

 1.62K

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

Flutter Geolocator Plugin

pub v7.0.3  style effective dart 

A Flutter geolocation plugin which provides easy access to platform specific location services ([FusedLocationProviderClient](#) or if not available the [LocationManager](#) on Android and [CLLocationManager](#) on iOS).

<https://pub.dev/packages/geolocator>

Geolocator 사용

```
dependencies:  
geolocator: ^7.0.3
```

```
import 'package:geolocator/geolocator.dart';
```

```
void getCurrentLocation() async {  
    LocationPermission permission = await Geolocator.checkPermission();  
  
    if (permission == LocationPermission.denied) {  
        Geolocator.requestPermission();  
    } else {  
        Position position = await Geolocator.getCurrentPosition(  
            desiredAccuracy: LocationAccuracy.low);  
        print(position);  
    }  
}
```

위치정보 권한 부여 : iOS

ios/Runner/Info.plist

```
<dict>
  <!-- for iOS 11 + -->
  <key>NSLocationWhenInUseUsageDescription</key>
  <string>Reason why app needs location</string>
  <key>NSLocationAlwaysAndWhenInUseUsageDescription</key>
  <string>Reason why app needs location</string>

  <!-- additionally for iOS 9/10, if you need always permission -->
  <key>NSLocationAlwaysUsageDescription</key>
  <string>Reason why app needs location</string>
  ...
</dict>
```

위치정보 권한 부여 : Android

android/app/src/main/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <!-- or -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

Stateful Widget Lifecycle

위젯이 처음 생성될 때 호출됨

```
@override  
void initState() {  
    super.initState();  
}
```

위젯의 state 가 바뀔 때마다 호출됨

```
@override  
Widget build(BuildContext context) {  
    return null;  
}
```

위젯이 화면에서 사라질 때 호출됨

```
@override  
void dispose() {  
    super.dispose();  
}
```

과제 : 위젯이 시작될 때 자동으로 위치를 가져오게 코드 변경

Dart Exception handling & Null aware 연산자

- 예외가 발생되면 처리를 해줘야 함
 - 처리되지 않은 예외는 프로그램을 비정상 종료 시킴

```
void main() {  
  
    String myString = 'abc';  
    double? convertedDouble;  
    try {  
        convertedDouble = double.parse(myString);  
    }  
    catch (e) {  
        print(e);  
    }  
    print(convertedDouble ?? 15);  
}
```

Location 리팩토링

- /services/location.dart 리팩토링

```
class LocationService {  
    double latitude;  
    double longitude;  
  
    Future<void> getCurrentLocation() {}  
}
```

API

- HTTP 를 통해 데이터를 요청하고 응답
 - 교환 데이터 포맷은 주로 JSON 을 사용
- Flutter 라이브러리 http 사용

http 0.13.1 

Published Mar 19, 2021 ·  dart.dev 

DART

NATIVE

JS

FLUTTER

ANDROID

IOS

LINUX

MACOS

WEB

WINDOWS

 2.46K

날씨 데이터 API



Weather API

[Home](#) / Weather API

Please [sign up](#) and use our fast and easy-to-work weather APIs for free. Look at our [monthly subscriptions](#) for more options rather than the Free account that we provide you. Read [How to start](#) first and enjoy using our powerful weather APIs.

Current & Forecast weather data collection

Current Weather Data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and vast network of weather stations
- JSON, XML, and HTML formats

Hourly Forecast 4 days

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- JSON and XML formats
- Available for Developer, Professional and Enterprise accounts

One Call API

[API doc](#) [Subscribe](#)

- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **National weather alerts**
- JSON format

<https://openweathermap.org/>

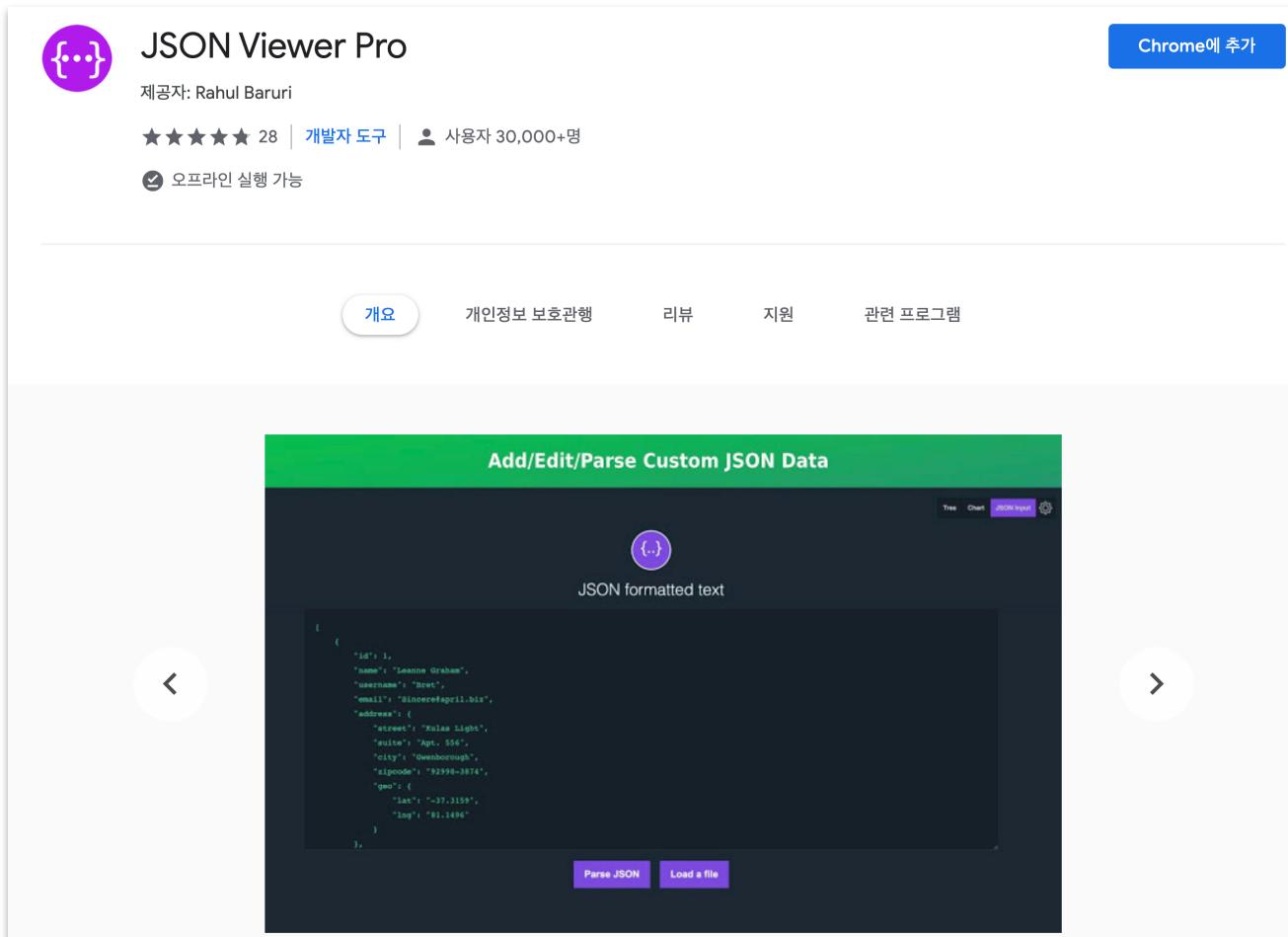
날씨 데이터 가져오기

```
Future<void> getWeather() async {
  Uri uri = Uri.https('api.openweathermap.org', '/data/2.5/weather', {
    'lat': '37.785834',
    'lon': '-122.406417',
    'appid': '5df3f1b99c9227bb6bb15d61c2b5bf9b',
    'units': 'metric',
  });
  Response response = await get(uri);
  print(response.body);
}
```

/services/networking.dart 파일에 NetworkHelper 클래스로 리팩토링

JSON 데이터 파싱

● 웹 브라우저용 JSON viewer



Navigation



Navigate to a new screen and back

[Docs](#) > [Cookbook](#) > [Navigation](#) > Navigate to a new screen and back

Most apps contain several screens for displaying different types of information. For example, an app might have a screen that displays products. When the user taps the image of a product, a new screen displays details about the product.

Terminology: In Flutter, *screens* and *pages* are called *routes*. The remainder of this recipe refers to routes.

In Android, a route is equivalent to an Activity. In iOS, a route is equivalent to a ViewController. In Flutter, a route is just a widget.

This recipe uses the [Navigator](#) to navigate to a new route.

The next few sections show how to navigate between two routes, using these steps:

1. Create two routes.
2. Navigate to the second route using `Navigator.push()`.
3. Return to the first route using `Navigator.pop()`.

<https://flutter.dev/docs/cookbook/navigation/navigation-basics>

대기 페이지에 스피너 넣기

flutter_spinkit 5.0.0

Published Mar 4, 2021 

FLUTTER | ANDROID IOS LINUX MACOS WEB WINDOWS

 1.34K

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

✨ Flutter Spinkit

A collection of loading indicators animated with flutter. Heavily inspired by [@tobiasahlin's SpinKit](#).

페이지 간 데이터 전달

- 네비게이션으로 페이지 이동 시 데이터 전달 가능
 - 대상 화면 생성자를 통해 전달

```
Navigator.push(context, MaterialPageRoute(builder: (context) {  
  return LocationScreen(weatherData: jsonWeatherData);  
}));
```

widget 속성

- **Stateful** 위젯
 - 화면과 상태(데이터)를 분리
 - StatefulWidget 과 State<Screen> 을 분리
- widget 속성을 통해 데이터 접근 가능

StatefulWidget class

Null safety

<https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html>

WeatherModel

- Model in MVC 패턴

- 데이터를 제공해주는 역할
- 비즈니스 모델 즉, 기능을 제공

```
Future<dynamic> getWeatherData() async {
    Location location = Location();
    await location.getCurrentLocation();

    NetworkHelper networkHelper = NetworkHelper(
        lat: location.latitude.toString(),
        lon: location.longitude.toString(),
    );
    return await networkHelper.getData();
}
```

도시 선택 화면 구성

● TextFiled 위젯 스타일링

onChanged 이벤트

TextField

`TextField` is the most commonly used text input widget.

By default, a `TextField` is decorated with an underline. You can add a label, icon, inline hint text, and error text by supplying an `InputDecoration` as the `decoration` property of the `TextField`. To remove the decoration entirely (including the underline and the space reserved for the label), set the `decoration` to null.

```
TextField(  
    decoration: InputDecoration(  
        border: OutlineInputBorder(),  
        hintText: 'Enter a search term'  
    ),  
);
```



<https://flutter.dev/docs/cookbook/forms/text-input>

constants.dart 사용

- 스타일이나 데코레이션 등
 - 중복 사용되는 부분은 상수로 별도로 관리

```
const kTempTextStyle = TextStyle(  
    fontFamily: 'Spartan MB',  
    fontSize: 100.0,  
);  
  
const kMessageTextStyle = TextStyle(  
    fontFamily: 'Spartan MB',  
    fontSize: 60.0,  
);  
  
const kButtonTextStyle = TextStyle(  
    fontSize: 30.0,  
    fontFamily: 'Spartan MB',  
);
```

Navigator.pop()에서 데이터 전달

- 화면으로 부터 반환되는 데이터 받기

push<T extends Object?> method Null safety

```
@optionalTypeArgs  
Future<T?> push <T extends Object?>(  
    BuildContext context,  
    Route<T> route  
)
```

pop<T extends Object?> method Null safety

```
@optionalTypeArgs  
void pop <T extends Object?>(  
    BuildContext context,  
    [T? result]  
)
```

<https://flutter.dev/docs/cookbook/navigation/returning-data>

- 프론트엔드 개발자 로드맵
- 글로벌 상태관리
- Dart 언어 심화
- 안드로이드, iOS 네이티브 개발 개요
- 고급 UI 작성
- UI 템플릿 사이트 활용

Flutter 기본 다음은..

플러터 개발 로드맵

- **스텝 0: 사전지식**

- OOP 컨셉 : Dart 언어
- 안드로이드 개발 지식이 있으면 굿!
- GIT & GitHub

플러터 개발 로드맵

- **스텝 1: 기본 사항**

- Dart 언어 기본
 - ❖ Lists
 - ❖ Map
 - ❖ Lambda Function
 - ❖ mixins
 - ❖ await and async

플러터 개발 로드맵

- **스텝 2: UI 개발**

- Flutter로 화면 개발 연습
- 클론 앱 만들기
- flutter way

- **스텝 3: API 호출 배우기**

- http, dio, retrofit, chopper
- 모델 클래스
- fake API 서버에서 데이터 받아오기로 시작

플러터 개발 로드맵

- **스텝 4: 데이터베이스 연동**
 - Firebase
 - Supabase
 - SQFLite (로컬 데이터베이스)
 - Hive (로컬 데이터베이스)
 - Moor (로컬 데이터베이스)

플러터 개발 로드맵

● 스텝 5: 상태관리 배우기

- 앱의 사이즈와 복잡성이 증가 될수록..
- 상태 (State) 관리의 중요성 증가
- 다양한 상태관리 기법과 패키지가 존재
 - ❖ Provider로 시작
 - ❖ BLoC
 - ❖ GetX

플러터 개발 로드맵

- 스텝 6(마지막): 프로젝트 구조 및 아키텍처
 - MVVM, MVC, TDD, etc
 - KISS
 - Scalable
 - 코드 재사용
 - 기능 간 의존성 최소화

수고하셨습니다.

김순곤

soongon@hucloud.co.kr