

```
..._mod = modifier_ob.  
...set mirror object to mirror  
...mirror_mod.mirror_object
```

```
...operation == "MIRROR_X":  
...mirror_mod.use_x = True  
...mirror_mod.use_y = False  
...mirror_mod.use_z = False  
...operation == "MIRROR_Y":  
...mirror_mod.use_x = False  
...mirror_mod.use_y = True  
...mirror_mod.use_z = False  
...operation == "MIRROR_Z":  
...mirror_mod.use_x = False  
...mirror_mod.use_y = False  
...mirror_mod.use_z = True
```

```
...selection at the end -add  
...mirror_ob.select= 1  
...modifier_ob.select=1  
...context.scene.objects.active  
...("Selected" + str(modifier_ob.  
...mirror_ob.select = 0  
...= bpy.context.selected_object  
...data.objects[one.name].select  
...print("please select exactly
```

```
...-- OPERATOR CLASSES --
```

```
...types.Operator):  
...X mirror to the selected  
...object.mirror_mirror_x"  
...mirror X"
```

```
...context):  
...context.active_object is not
```



# FLUTTR FRAMEWORK

ICT UPSKILLING PROGRAMME  
(ICTUPSKILL)

UNTUK PPTM GRED FA29 (KUP)  
SEKTOR AWAM

15 - 17 November 2021

# What is Flutter

- Flutter is an open-source UI Framework development kit created by Google
- Flutter is Google UI toolkit for building beautiful, natively compiled applications for mobile, web and desktop from a single codebase.
- Flutter uses hybrid approach
- It is use to develop application for Android, iOS, Windows, Mac, Linux, Google Fushsia and the web
- Flutter used Dart Language

# Mobile Platforms





## Mobile Platforms

- Android
- iOS



## Mobile Platforms

- Mobile development approaches
  - Native
  - Hybrid
  - Web



## Mobile Platforms

# Native app approaches

## iOS

- ObjC
- Swift

## Android

- Kotlin
- Java



## Mobile Platforms

- Web app approaches
  - HTML
  - CSS
  - JS



## Mobile Platforms

- Hybrid approaches
  - PhoneGap
  - Cordova
  - IONIC





## Mobile Platforms

- Other popular approaches
  - Xamarin
  - React Native
  - Appclerator

So  
what's is  
Flutter

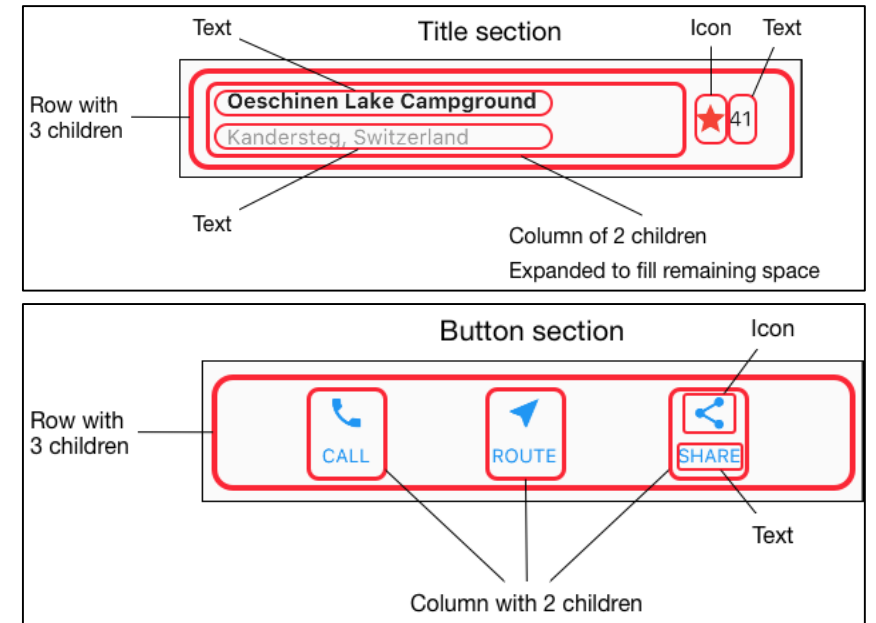


# So what's is Flutter

- So what's is Flutter
  - Flutter uses hybrid approach
  - Flutter is Google UI toolkit for building beautiful, natively compiled applications for mobile, web and desktop form a single codebase.



# What language is Flutter build with?



```
mirror_mod = modifier_ob.  
set mirror object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```


```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES --
```

```
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

# Widgets in Flutter



## What are Widgets in Flutter

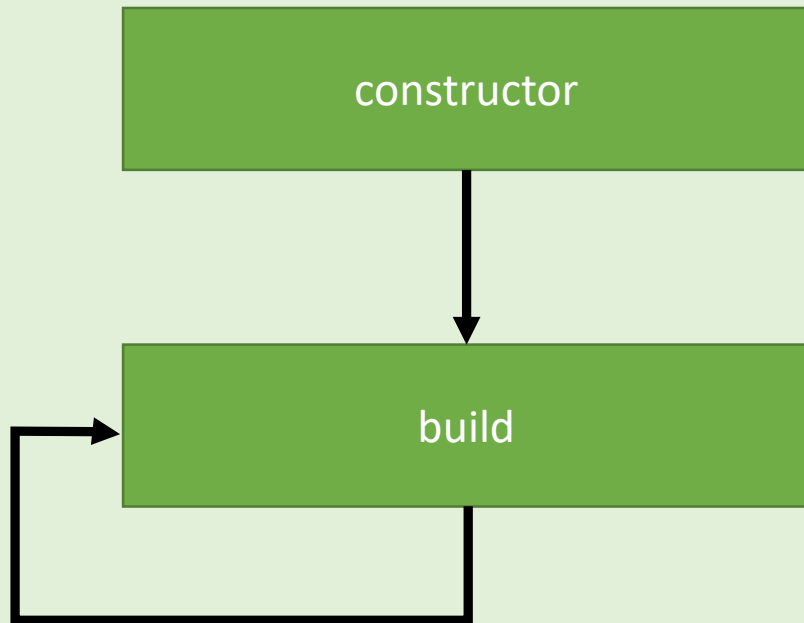
- Everything in flutter consist of Widgets including but not limited to visible Screen, Text, Button, Material Design, Application Bar
  - OOP - Object Oriented Programming (*Class, Constructors, Properties and methods, Methods: static, private, etc*)
  - There are 2 types of Widgets:
    - i. Stateless Widgets
    - ii. Stateful Widgets
- 



# Everything

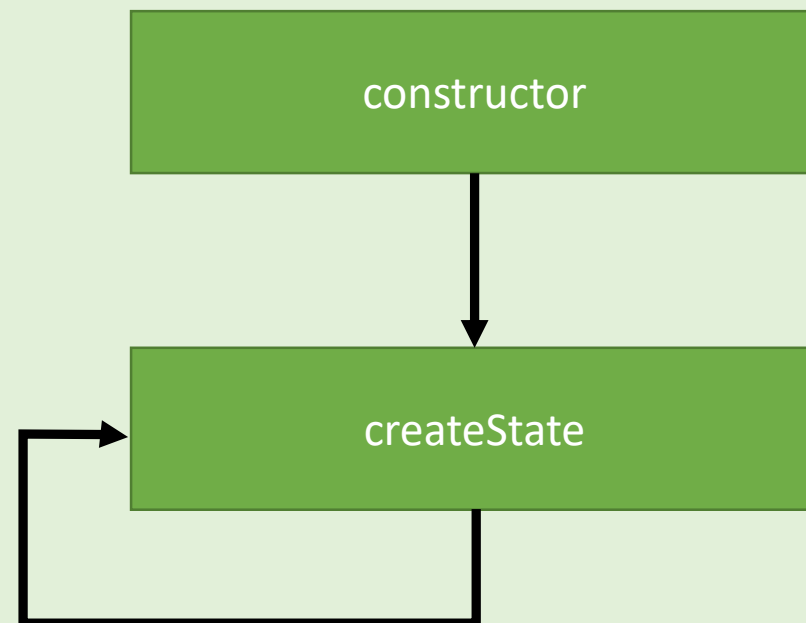
# Stateful Widget vs Stateless Widget

## Stateless Widget



A single StatelessWidget can build in many different BuildContexts

## Stateful Widget



A single StatefulWidget creates a new State object for each BuildContexts



# Stateless Widget

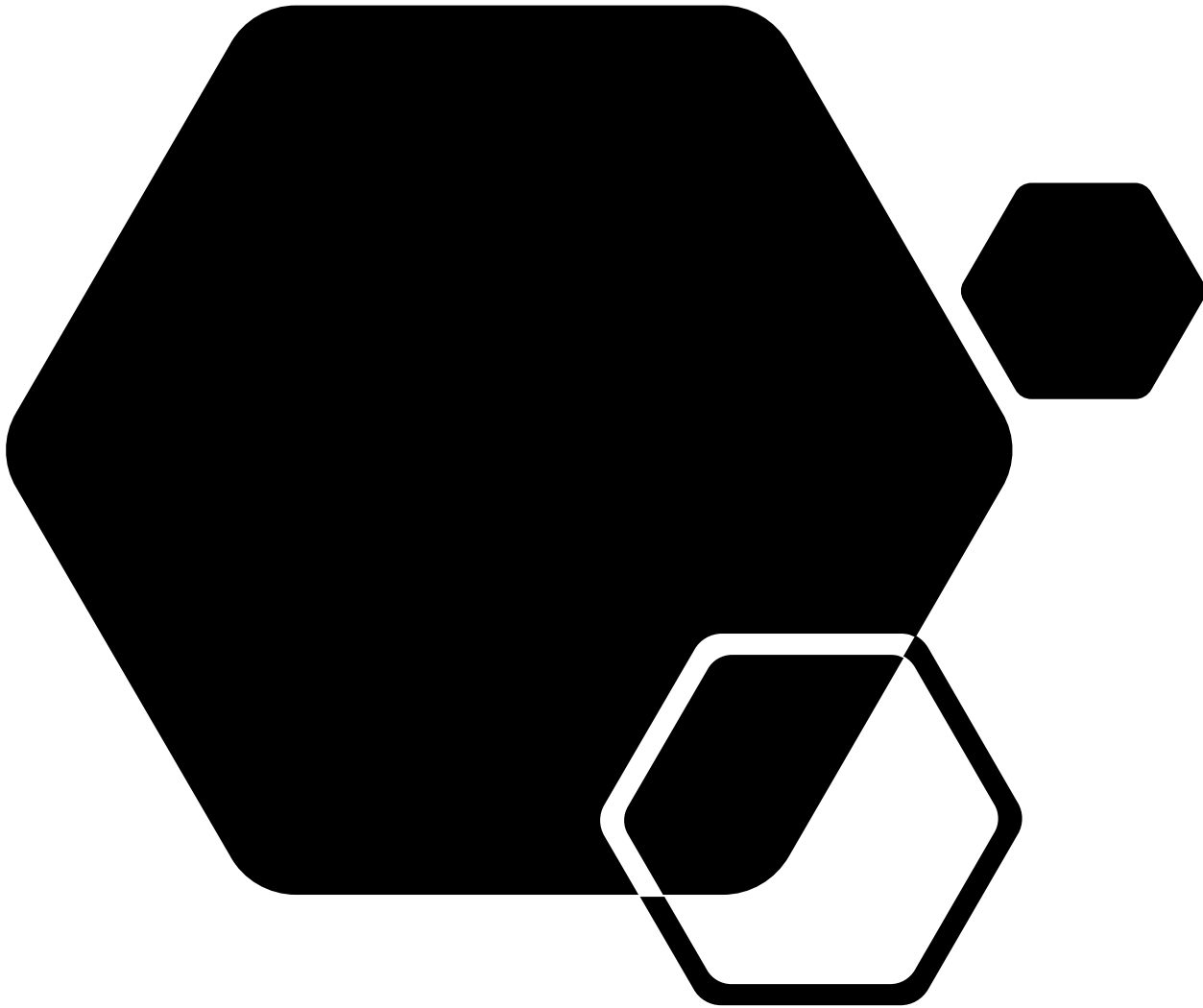
- We create Stateless widget by extending our class from StatelessWidget
- Single Class
- No State

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

# Stateful Widget

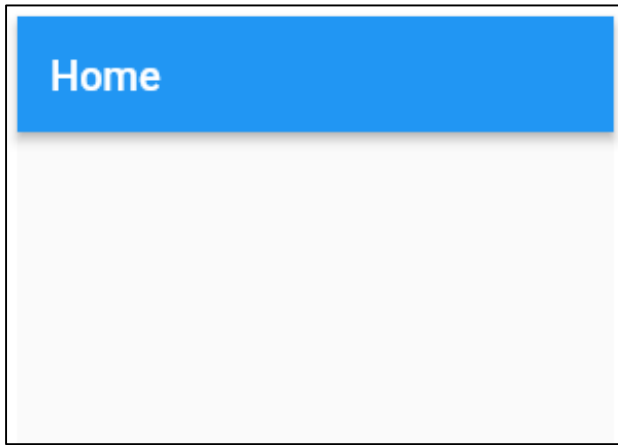
- Stateful Widgets are mutable and can be drawn multiple time within its lifetime
- 2 Classes: State & UI
- Has State -> createState()

```
class MyApp extends StatefulWidget {  
  @override  
  _MyAppPageState createState() => _MyAppPageState();  
}  
  
class _MyAppPageState extends State<MyApp> {  
  @override  
  Widget build(BuildContext context) {  
    return Container();  
  }  
}
```

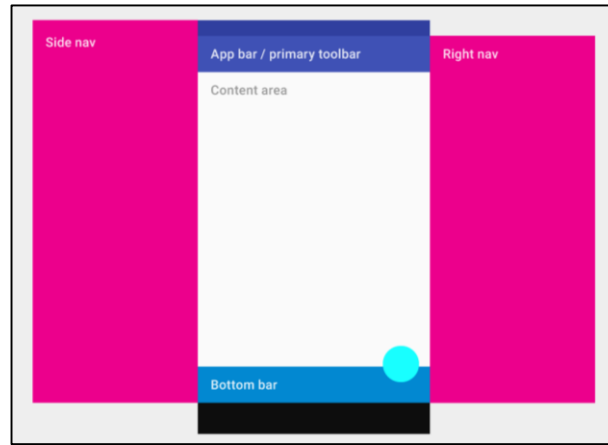


# Flutter Widget Catalog

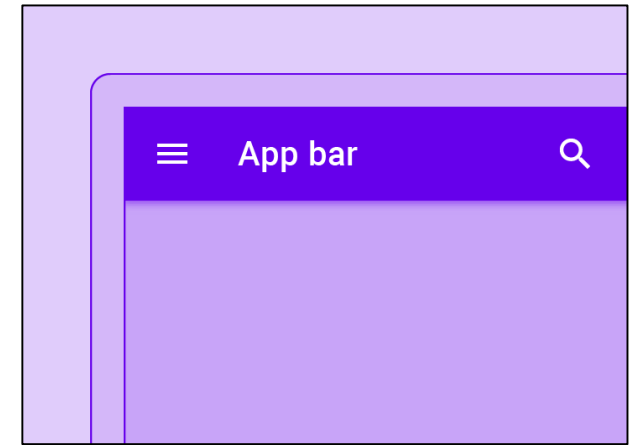
# Flutter Widget



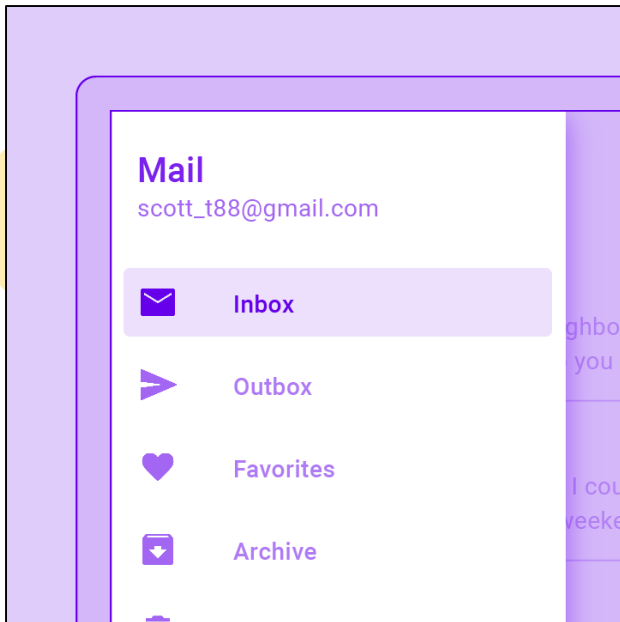
MaterialApp



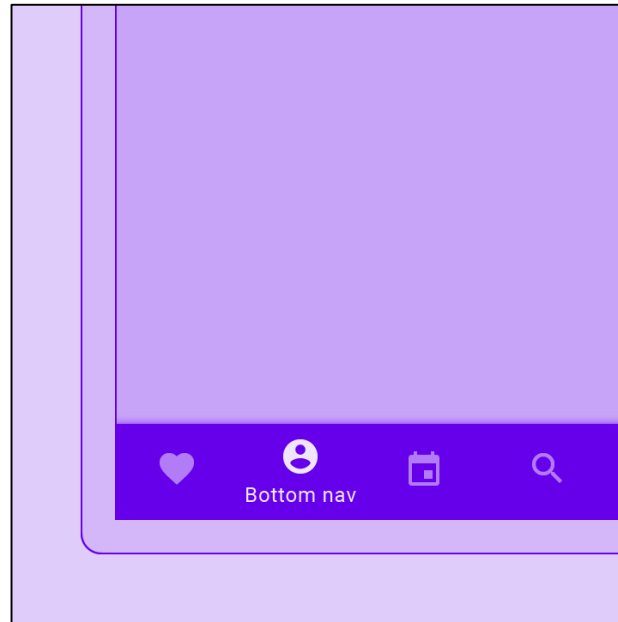
Scaffold



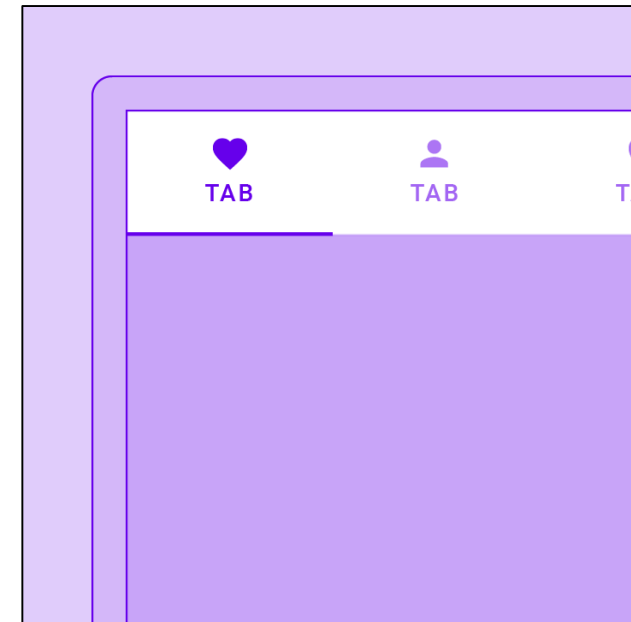
App Bar



Sidebar

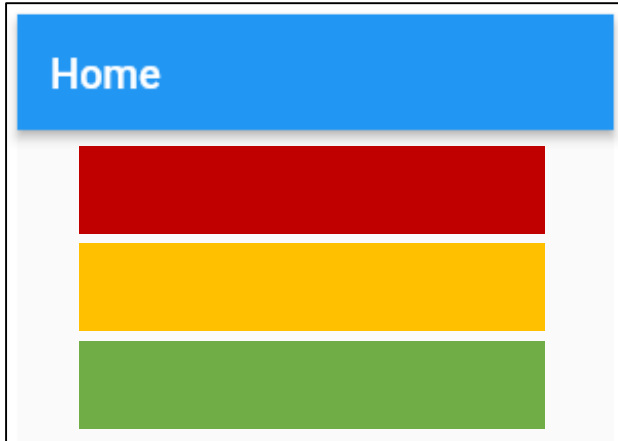


Bottom Navigation Bar

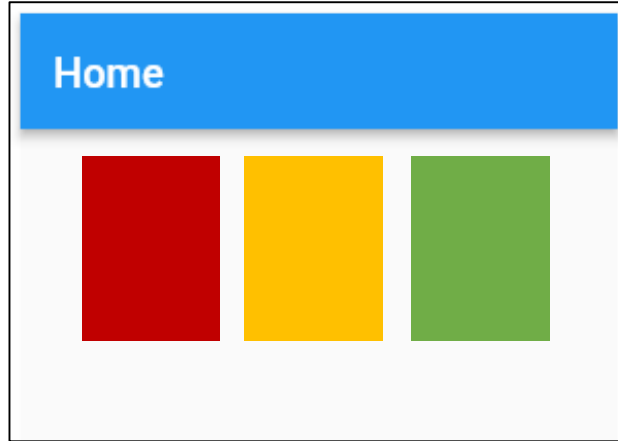


Tap Bar

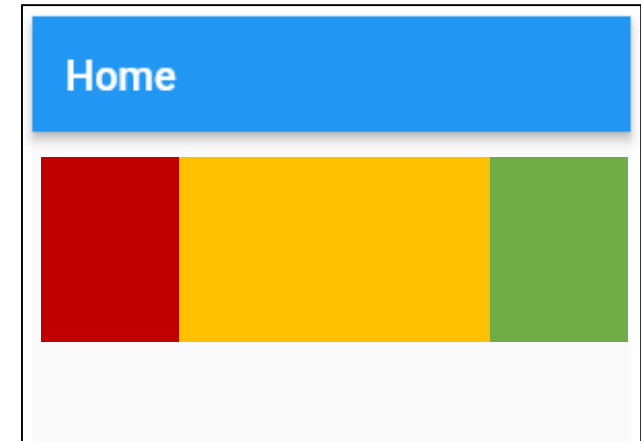
# Flutter Widget



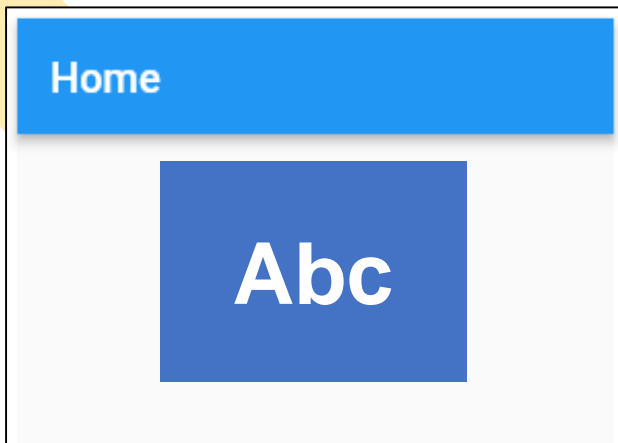
Column



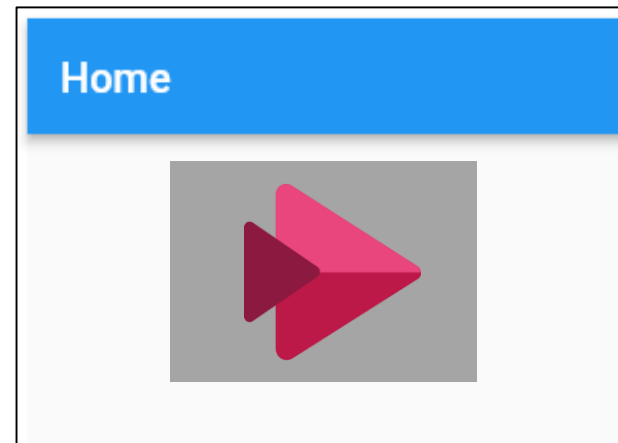
Row



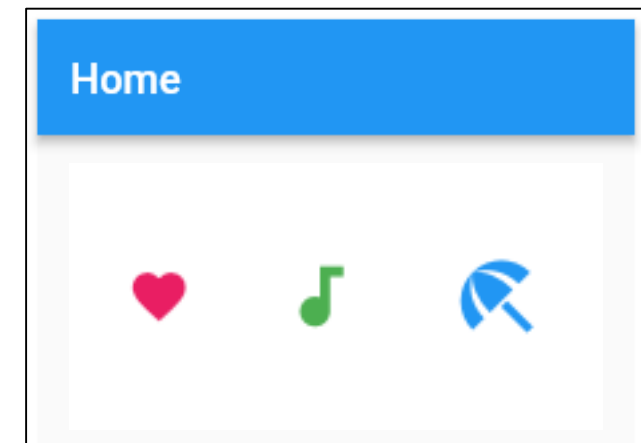
Expanded



Text



Images



Icon



Development Software

# Setup & Software

# Setup & Softwares



Flutter | Dart Plugins

- <https://flutter.dev/docs>



Android Studio + AVD (Emulator)

- <https://developer.android.com/studio>



Visual Studio Code

- <https://code.visualstudio.com>

# New Project

- Visual Studio Code
  - View > Command Palette... > Flutter: New Application Project
- Terminal or CMD
  - flutter create yourProjectName

# Flutter Project Structure

Related dart Tools

Android Project

Apk, AAB or IPA

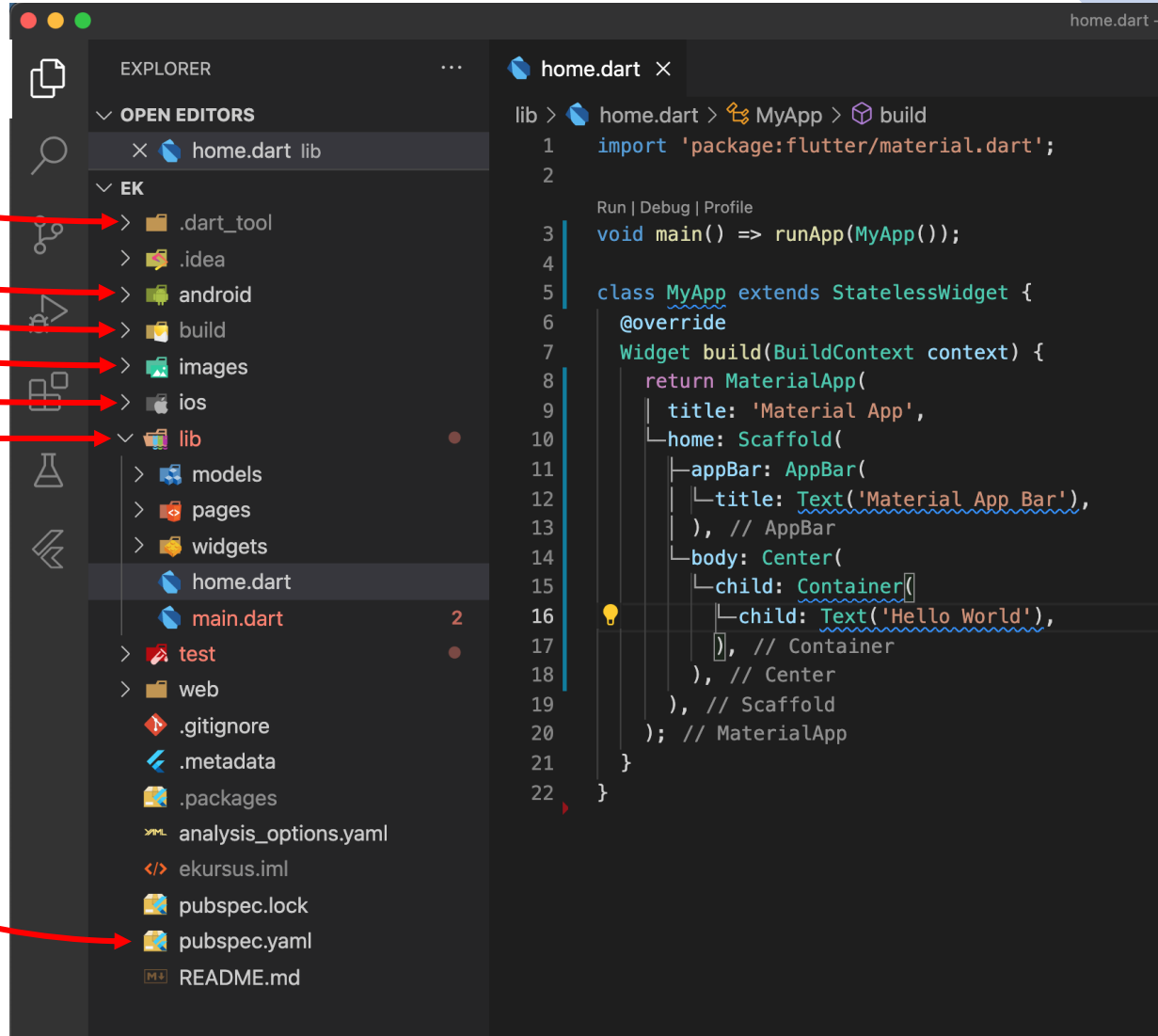
Images or fonts, etc folder

iOS Project

All dart code Files

Dependencies Requirements

- assets
- fonts





# Awesome Flutter Snippets

importM	Material Package	Import Material package.
mateapp	Material App	Create a new Material App.

statelessW	Stateless Widget	Creates a Stateless widget
statefulW	Stateful Widget	Creates a Stateful widget
initS	InitState	Called when this object is inserted into the tree. The framework will call this method exactly once for each State object it creates.

<https://marketplace.visualstudio.com/items?itemName=Nash.awesome-flutter-snippets>

# Useful or Common Layout and Widgets



# Flutter Layout & UI Elements

## Layout Element

- Container
- Row
- Column
- Stack

## UI Element

- Text
- Image
- Buttons
- Icon

# Common FFlutter Widgets

## Widget with single root, child

- Container
- Buttons - ElevatedButton, OutlinedButton, TextButton
- Card
- SingleChildScrollView

## Widgets with multiple, children:

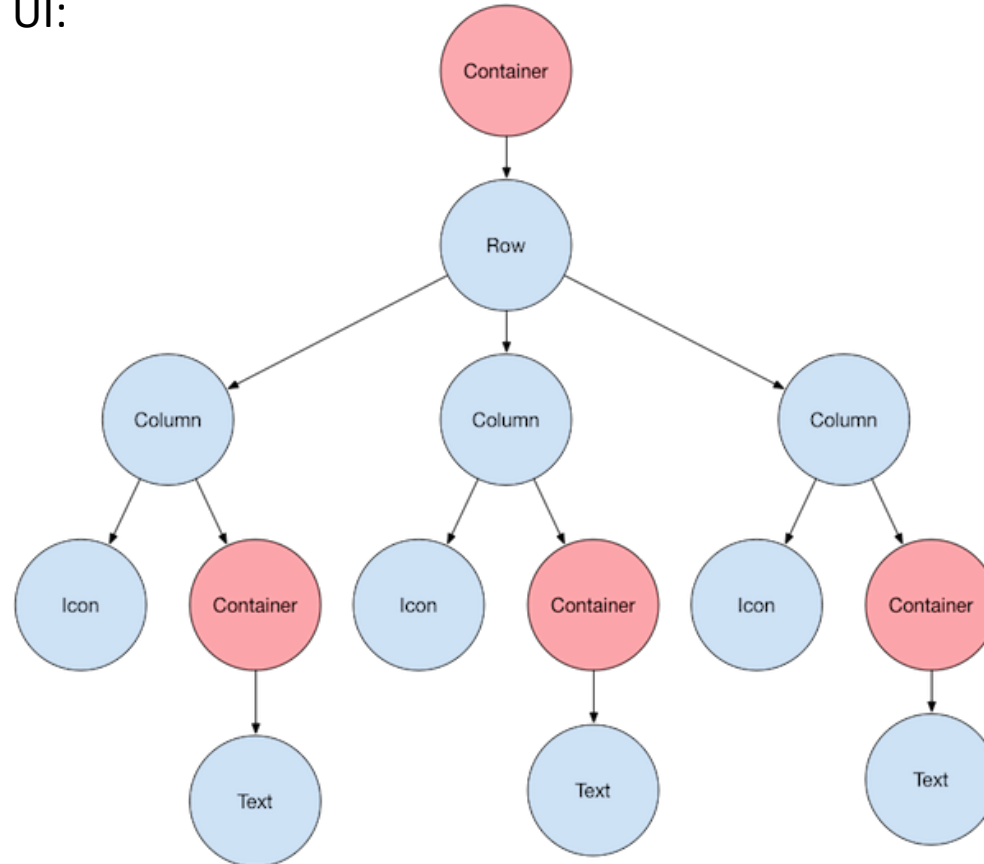
- Column
- Row
- ListView
- Stack
- Wrap

# Flutter Layout

For example, the first screenshot below shows 3 icons with a label under each one:



Here's a diagram of the widget tree for this UI:



# Flutter Layout

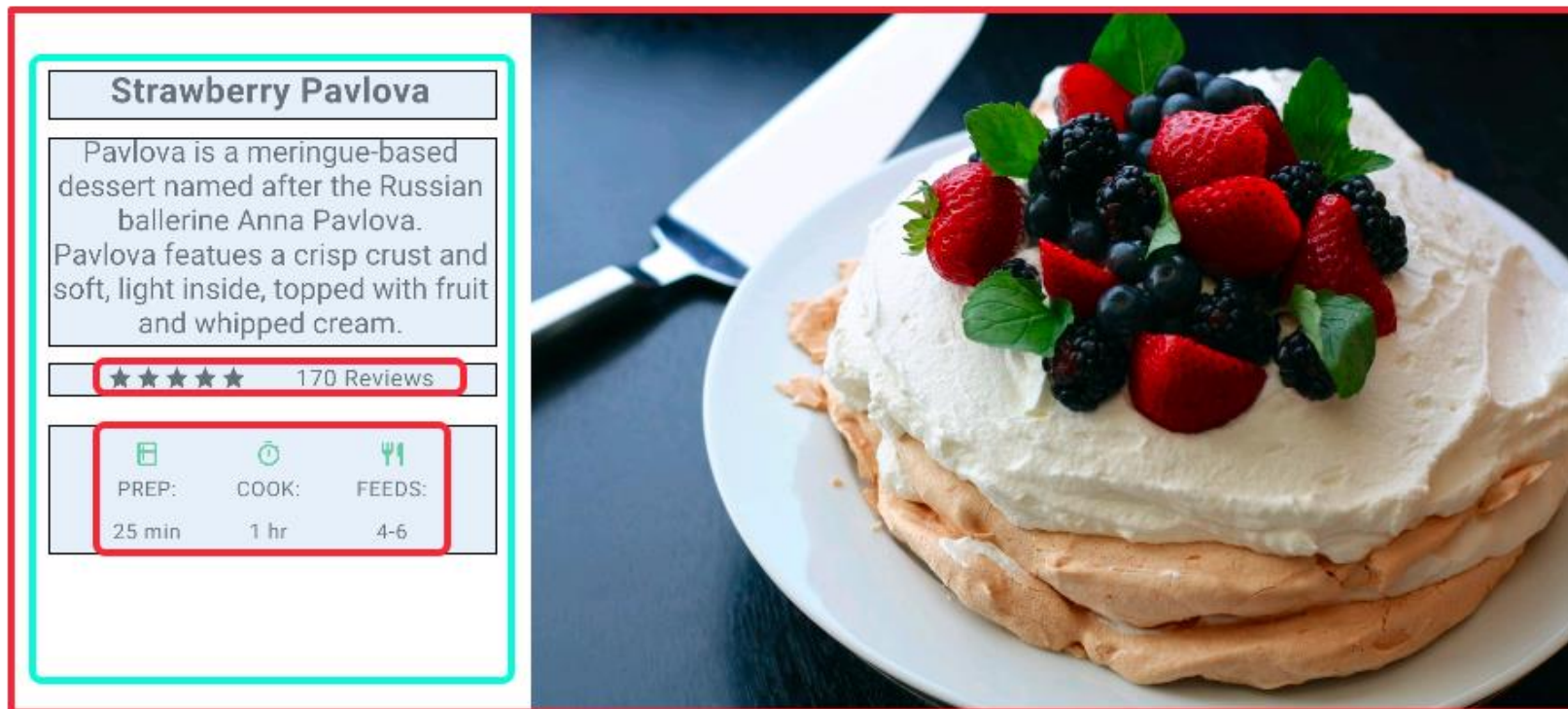
Layout multiple widgets vertically and horizontally

Row  
2 children

child: new Column

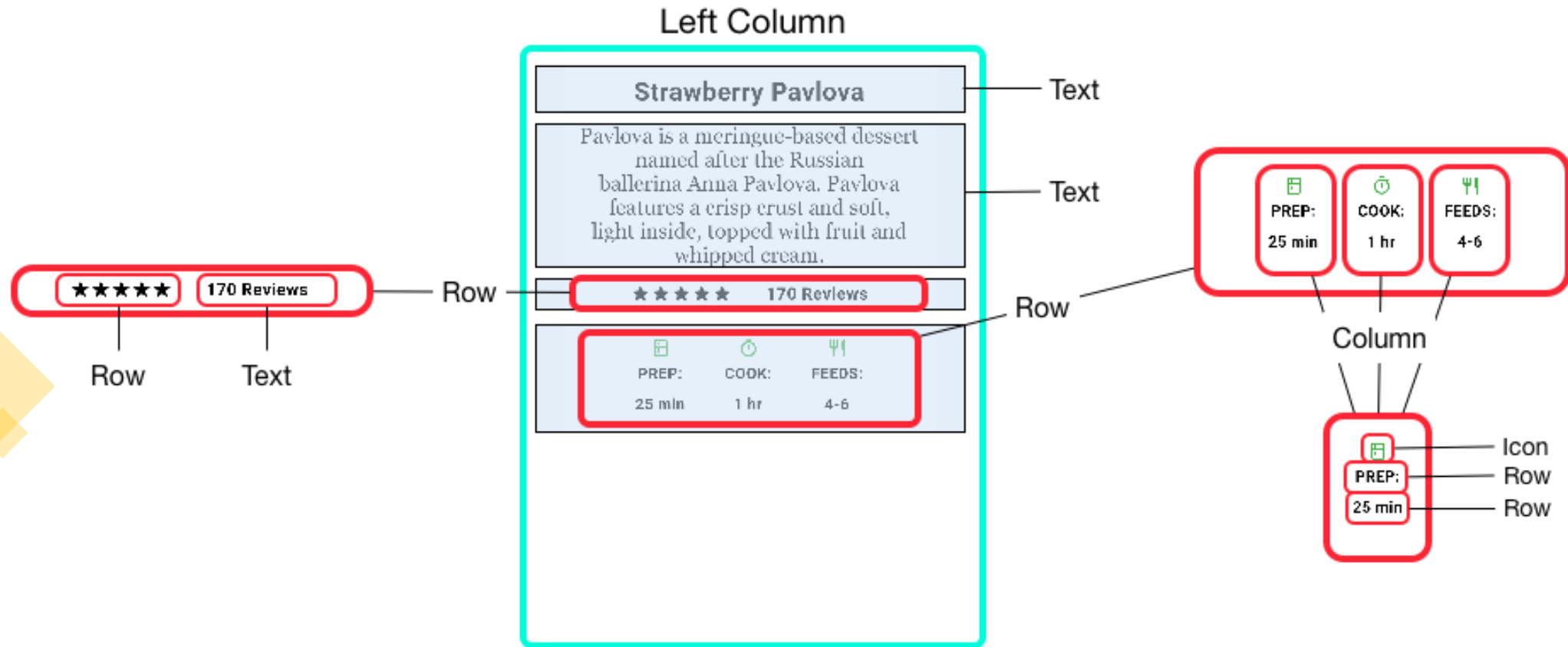
child: new Image

Column  
4 children



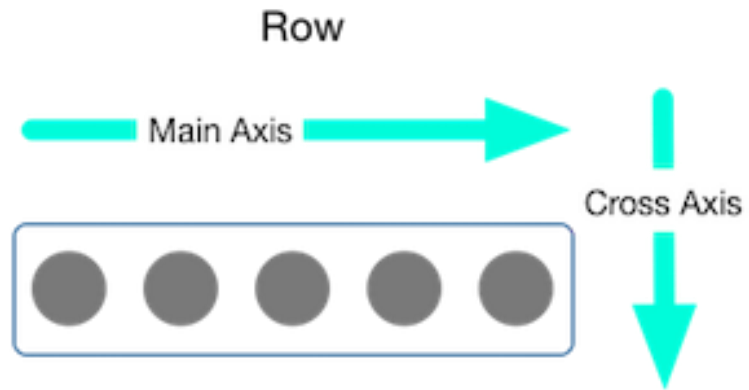
# Flutter Layout

Layout multiple widgets vertically and horizontally

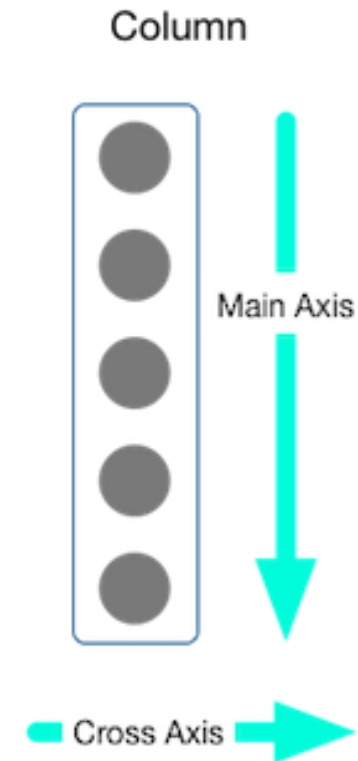


# Flutter Layout

Row Aligning widgets



Column Aligning widgets





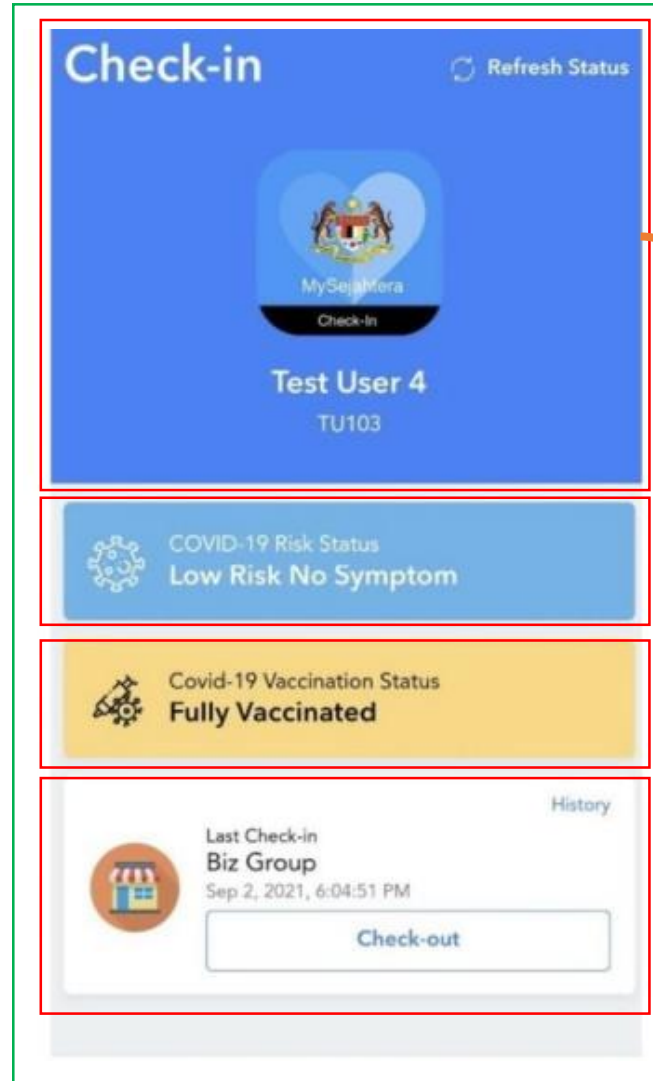
# Flutter Layout

## MySejahtera

How many column and row?

1 Column

4 Row



How many column and row?

1 Column

4 Row

The image features a dark blue background with blurred, glowing lines of code in white and yellow. A central white rectangular box with a thin black border contains the title and introductory text. The title 'Flutter Cookbook' is prominently displayed in a large, black, sans-serif font. Below it, a paragraph of text explains the purpose of the cookbook. At the bottom of the box, the URL 'https://flutter.dev/docs/cookbook' is written in a large, black, sans-serif font. The background code snippets are partially legible, showing Dart code related to mirror objects and operations.

# Flutter Cookbook

This cookbook contains recipes that demonstrate how to solve common problems while writing Flutter apps. Each recipe is self-contained and can be used as a reference to help you build up an application.

<https://flutter.dev/docs/cookbook>

# Let's Start a New Project

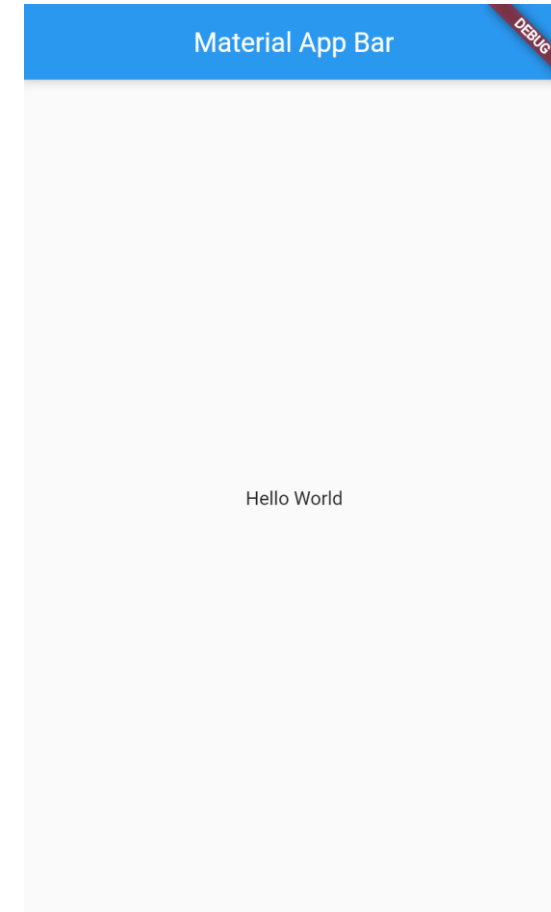
---



# Visual Studio Code

View > Command Palette... > Flutter: New Application Project

- Name App : MyApp



# Visual Studio Code

View > Command Palette... > Flutter: New Application Project

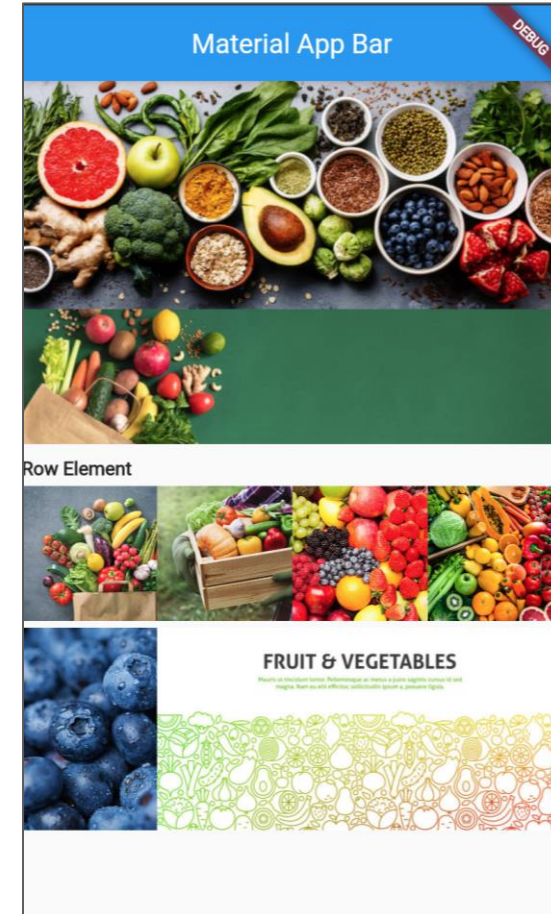
- Name App : AppLayout



# Visual Studio Code

View > Command Palette... > Flutter: New Application Project

- Name App : AppFruit





# How to build APK

## Build Release APK

By default, if not specified, the generated APK files use release mode.

```
flutter build apk
```

By default, if not specified, the generated APK files use release mode.

```
flutter build apk --release
```

## Build Debug APK

For debug mode, you need to add --debug flag.

```
flutter build apk --debug
```

## Set Build Name and Version

For example, version: 1.0.1+2 means the build name is 1.0.1, while the build number is 2.

There's an alternative way if you want to build APKs with different version name and number

```
flutter build apk --build-name=1.0.2 --build-number=3
```

<https://www.woolha.com/tutorials/flutter-build-android-apk>

<https://github.com/mzm-dev/upskill-flutter>

Source code