

The Basics

Widgets & How Apps Are Built

What's In This Section?

How a Flutter App Starts & Works

Working with Widgets & Building Custom
Widgets

Reacting to User Events

Stateless & Stateful Widgets

Dart Fundamentals

CREATING A NEW PROJECT

```
Q:\myworkspaces\flutter_ws>flutter create belajar_flutter_02
Creating project belajar_flutter_02...
belajar_flutter_02\lib\main.dart (created)
belajar_flutter_02\pubspec.yaml (created)
belajar_flutter_02\README.md (created)
belajar_flutter_02\test\widget_test.dart (created)
belajar_flutter_02\.gitignore (created)
belajar_flutter_02\.idea\libraries\Dart_SDK.xml (created)
belajar_flutter_02\.idea\libraries\KotlinJavaRuntime.xml (created)
belajar_flutter_02\.idea\modules.xml (created)
belajar_flutter_02\.idea\runConfigurations\main_dart.xml (created)
belajar_flutter_02\.idea\workspace.xml (created)
belajar_flutter_02\.metadata (created)
belajar_flutter_02\analysis_options.yaml (created)
belajar_flutter_02\android\app\build.gradle (created)
belajar_flutter_02\android\app\src\main\kotlin\com\example\belajar_flutter_02>MainActivity.kt (created)
belajar_flutter_02\android\build.gradle (created)
belajar_flutter_02\android\belajar_flutter_02_android.iml (created)
belajar_flutter_02\android\.gitignore (created)
belajar_flutter_02\android\app\src\debug\AndroidManifest.xml (created)
belajar_flutter_02\android\app\src\main\AndroidManifest.xml (created)
belajar_flutter_02\android\app\src\main\res\drawable\launch_background.xml (created)
belajar_flutter_02\android\app\src\main\res\drawable-v21\launch_background.xml (created)
belajar_flutter_02\android\app\src\main\res\mipmap-hdpi\ic_launcher.png (created)
belajar_flutter_02\android\app\src\main\res\mipmap-mdpi\ic_launcher.png (created)
belajar_flutter_02\android\app\src\main\res\mipmap-xhdpi\ic_launcher.png (created)
belajar_flutter_02\android\app\src\main\res\mipmap-xxhdpi\ic_launcher.png (created)
```

```
Command Prompt — Command Prompt
```

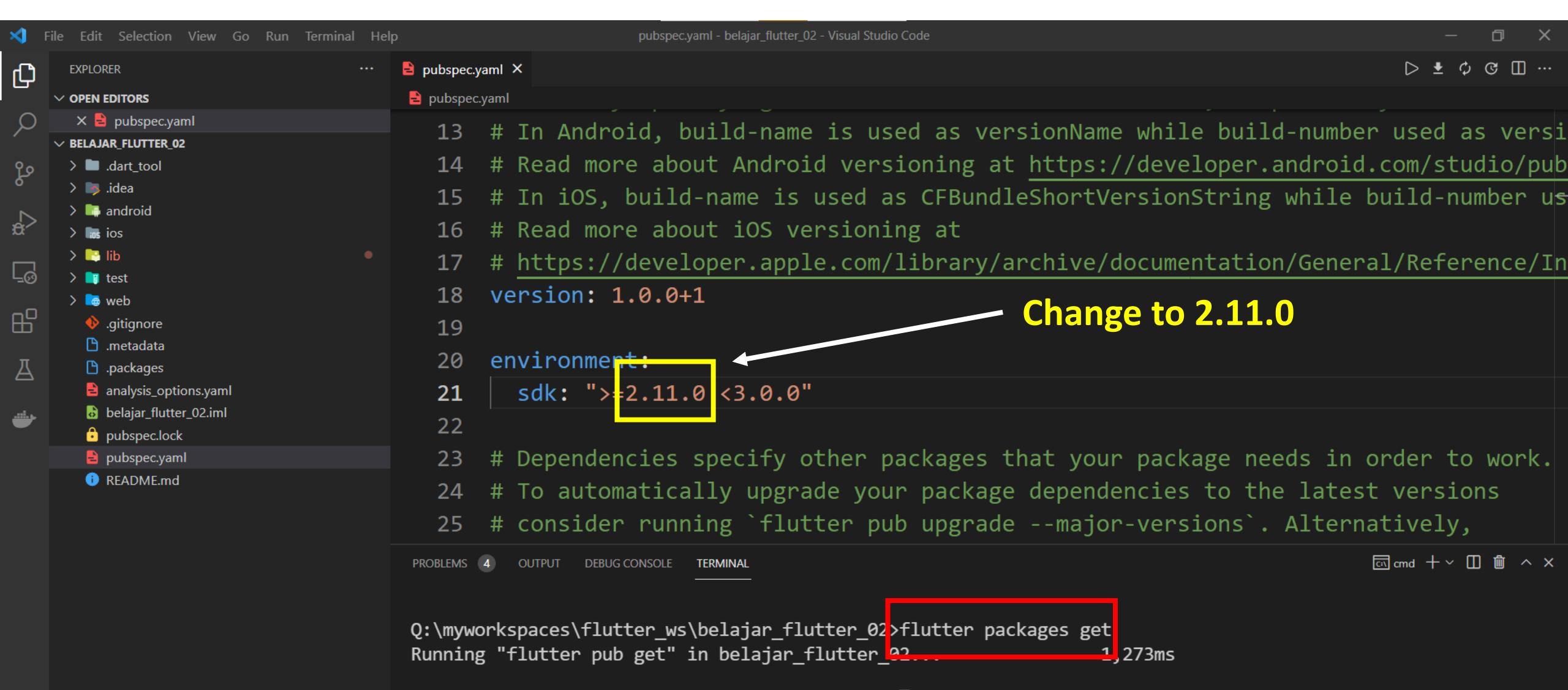
belajar_flutter_02\ios\Runner.xcworkspace\contents.xcworkspacedata (created)
belajar_flutter_02\ios\Runner.xcworkspace\xcshareddata\IDEWorkspaceChecks.plist (created)
belajar_flutter_02\ios\Runner.xcworkspace\xcshareddata\WorkspaceSettings.xcsettings (created)
belajar_flutter_02\belajar_flutter_02.iml (created)
belajar_flutter_02\web\favicon.png (created)
belajar_flutter_02\web\icons\Icon-192.png (created)
belajar_flutter_02\web\icons\Icon-512.png (created)
belajar_flutter_02\web\icons\Icon-maskable-192.png (created)
belajar_flutter_02\web\icons\Icon-maskable-512.png (created)
belajar_flutter_02\web\index.html (created)
belajar_flutter_02\web\manifest.json (created)
Running "flutter pub get" in belajar_flutter_02... 2,680ms
Wrote 81 files.

All done!
In order to run your application, type:

```
$ cd belajar_flutter_02  
$ flutter run
```

Your application code is in belajar_flutter_02\lib\main.dart.

Q:\myworkspaces\flutter_ws>cd belajar_flutter_02
Q:\myworkspaces\flutter_ws\belajar_flutter_02>code .
Q:\myworkspaces\flutter_ws\belajar_flutter_02>

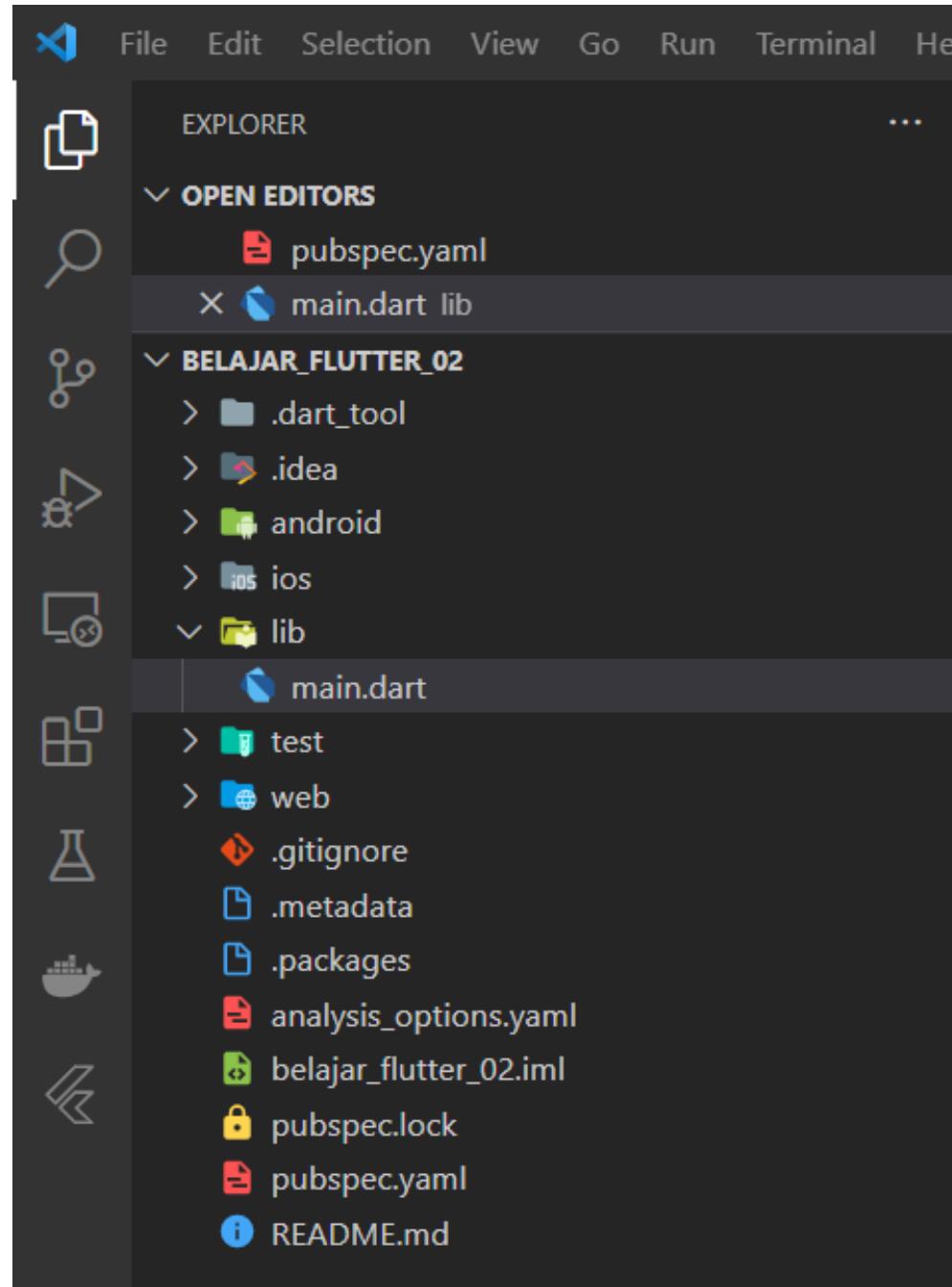


What does flutter packages get do?

When running flutter pub get (Packages get in IntelliJ or Android Studio) for the first time after adding package, **Flutter saves the concrete package version found in the pubspec.lock lockfile**. This ensures that you get the same version again if you, or another developer on your team, run flutter pub get .

itter_02>

AN OVERVIEW OF THE GENERATED FILES AND FOLDERS



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code.
- Left Sidebar (Explorer):**
 - OPEN EDITORS: pubspec.yaml, main.dart
 - BELAJAR_FLUTTER_02: .dart_tool, .idea, android, ios, lib (main.dart), test, web, .gitignore, .metadata, .packages, analysis_options.yaml, belajar_flutter_02.iml, pubspec.lock, pubspec.yaml, README.md
- Editor Area:** The main.dart file is open in the editor. The code is as follows:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key key}) : super(key: key);
9
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // Try running your application with "flutter run". You'll see the
19       // application has a blue toolbar. Then, without quitting the app, try
20       // changing the primarySwatch below to Colors.green and then invoke
21       // "hot reload" (press "r" in the console where you ran "flutter run",
22       // or simply save your changes to "lib/main.dart").
```
- Bottom Status Bar:** Ln 10, Col 50, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** pubspec.yaml - belajar_flutter_02 - Visual Studio Code
- Left Sidebar (Explorer):**
 - OPEN EDITORS: pubspec.yaml, main.dart
 - BELAJAR_FLUTTER_02 folder:
 - .dart_tool
 - .idea
 - android
 - ios
 - lib:
 - main.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Central Area:** Code editor showing the content of the pubspec.yaml file.

```
19
20 environment:
21   sdk: ">=2.11.0 <3.0.0"
22
23 # Dependencies specify other packages that your package needs in order to work.
24 # To automatically upgrade your package dependencies to the latest versions
25 # consider running `flutter pub upgrade --major-versions`. Alternatively,
26 # dependencies can be manually updated by changing the version numbers below to
27 # the latest version available on pub.dev. To see which dependencies have newer
28 # versions available, run `flutter pub outdated`.
29 dependencies:
30   flutter:
31     sdk: flutter
32
33 # The following adds the Cupertino Icons font to your application.
34 # Use with the CupertinoIcons class for iOS style icons.
35/cupertino_icons: ^1.0.2
36
37 dev_dependencies:
38   flutter_test:
39     sdk: flutter
40
```
- Bottom Status Bar:** Ln 21, Col 15, Spaces: 2, U, Zoom, TRLF, YAML, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), ✓ Prettier, ⚡

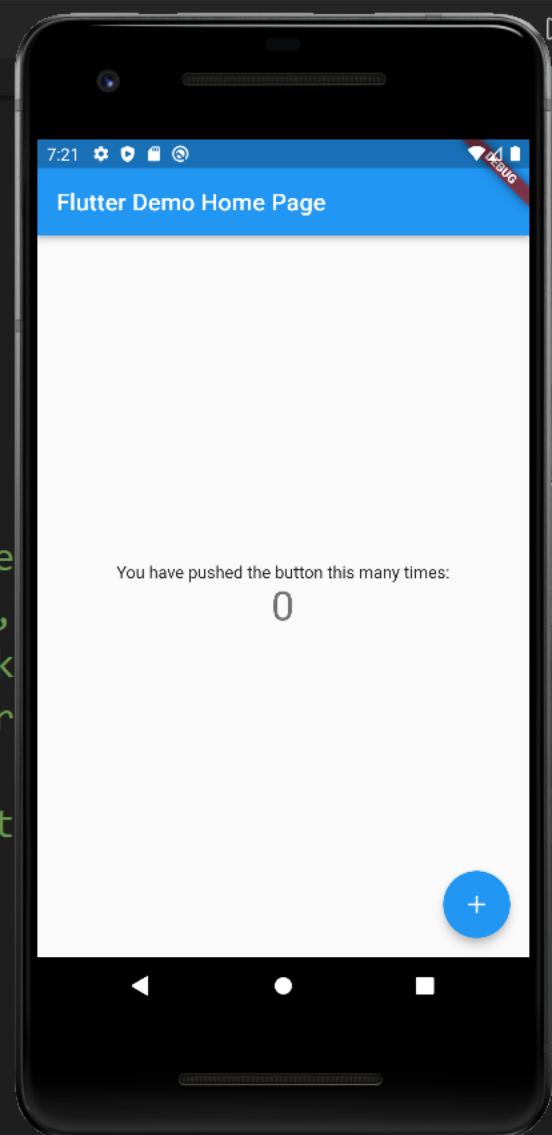
ANALYZING THE DEFAULT APP



main.dart X

lib > main.dart > _MyHomePageState > build

```
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // Try running your application with "flutter run". You'll see the
19       // application has a blue toolbar. Then, without quitting the app,
20       // changing the primarySwatch below to Colors.green and then invoking
21       // "hot reload" (press "r" in the console where you ran "flutter run"
22       // or simply save your changes to "hot reload" in a Flutter IDE).
23       // Notice that the counter didn't reset back to zero; the application
24       // is not restarted.
25       primarySwatch: Colors.blue,
26     ), // ThemeData
27     home: const MyHomePage(title: 'Flutter Demo Home Page'),
28   ); // MaterialApp
29 }
30 }
31 }
```

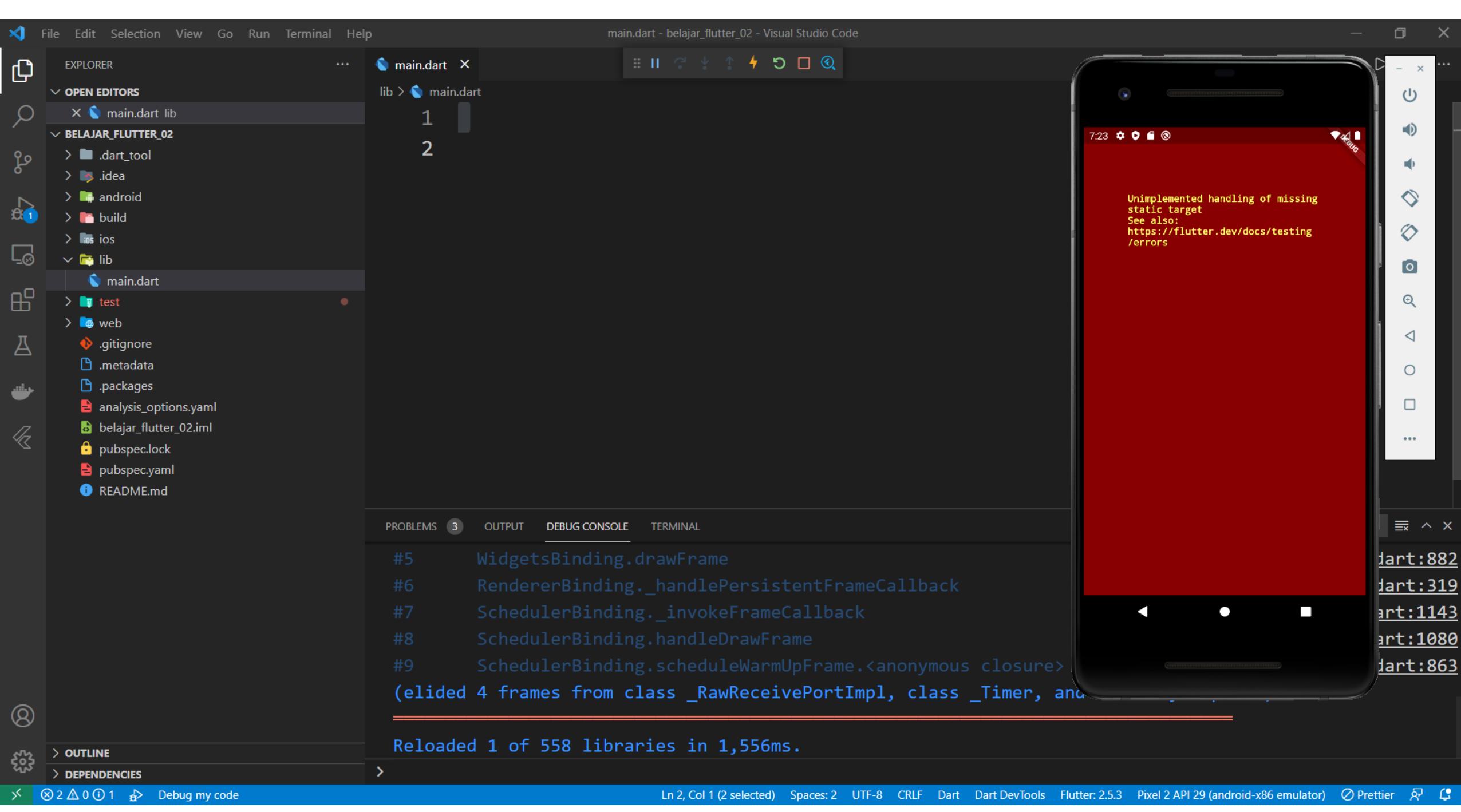




main.dart X

```
lib > main.dart > _MyHomePageState > build
94          // axis because columns are vertical (the cross axis would be
95          // horizontal).
96          mainAxisAlignment: MainAxisAlignment.center,
97          children: <Widget>[
98              const Text(
99                  'You have pushed the button this many times:',
100                 ), // Text
101                 Text(
102                     '_counter',
103                     style: Theme.of(context).textTheme.headline4,
104                     ), // Text
105                 ], // <Widget>[]
106             ), // Column
107         ), // Center
108         floatingActionButton: FloatingActionButton(
109             onPressed: _incrementCounter,
110             tooltip: 'Increment',
+ 111             child: const Icon(Icons.add),
112             ), // This trailing comma makes auto-formatting nicer for build methods
113         ); // Scaffold
114     }
115 }
116 }
```





DART BASICS

Windows install | Flutter dart pad - Bing

https://www.bing.com/search?q=dart+pad&cvid=8986e0fc48c64f50b376999ad08c4ef6&aqs=edge..69i57j0l8.5941j0j4&FORM=ANAB01&PC=EDGEDB

Microsoft Bing dart pad ALL IMAGES VIDEOS MAPS NEWS Indonesia Rizqi Get the Wallpaper App >

164.000 Results Date ▾



DartPad

<https://www.dartpad.dev> ▾

DartPad. code New Pad refresh Reset format_align_left Format get_app Install SDK Samples expand_more. more_vert launch Share; launch DartPad on GitHub; launch dart.dev; launch flutter.dev; Dart HTML CSS. play_arrow Run Console Documentation close UI Output ...

EXPLORE FURTHER

[Dart Editor \(64-bit\)](#) - Free download and software reviews ...

[Dart for Windows](#)

[DART Online Editor - DART Online IDE - DART Coding Online](#)

[DartPad | Dart](#)

[GitHub - dart-lang/dart-pad: An online Dart editor with ...](#)

Recommended to you based on what's popular • Feedback

DartPad | Dart

<https://dart.dev/tools/dartpad> ▾

DartPad is an open source tool that lets you play with the Dart language in any modern browser. Many pages in this site – especially codelabs – have embedded DartPads. To open DartPad as a standalone web page, visit the DartPad site (dartpad.dev). Tip: If you're in China, try dartpad.cn. If you have issues using DartPad, see the DartPad troubleshooting tips.

DartPad

<https://dartpad.github.io> ▾

Dart

Programming Language



Dart

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language... +



[Wikipedia](#) [Official site](#)

Developer: Google

Operating systems: Cross-platform

License: BSD licenses

Typing discipline: 1.x: Optional, 2.x: Inferred (static, strong)

Paradigm: Multi-paradigm: functional, imperative, object-oriented, reflective

Preview release: 2.14.1 / September 8, 2021; 48 days ago

People also search for

See all (15+)



New Pad Reset Format Install SDK

magenta-tulip-2609

Samples



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

Run

Console

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

Documentation



DartPad

New Pad

Reset

Format

Install SDK

magenta-tulip-2609

Samples



```
1 addNumbers
2
3 void main() {
4
5 }
6
```

Run

Console

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

Documentation

error line 1 • Variables must be declared using the keywords 'const', 'final', 'var' or a type name.

Try adding the name of the type of the variable or the keyword 'var'.

error line 1 • Expected to find ';'.

info line 1 • Prefer typing uninitialized variables and fields. ([view docs](#))



Privacy notice Send feedback



Null Safety

Zoom

3 issues [hide](#)

Based on Flutter 2.5.3 Dart SDK 2.14.4





DartPad <> New Pad C Reset ┌ Format └ Install SDK

magenta-tulip-2609

Samples ▾



```
1 addNumbers(num1, num2) {  
2   print(num1 + num2);  
3 }  
4  
5 void main() {  
6   addNumbers(1,2);  
7 }  
8
```

▶ Run

Console

3

Documentation

dynamic num2



Data Types

A program works with different types of data

Text

Strings

“Hi there!”

Data Types

A program works with different types of data

Text

Numbers

Strings

Integers

Floats

“Hi there!”

29

29.99

```
1 addNumbers(int num1, int num2) {  
2   print(num1 + num2);  
3 }  
4  
5 void main() {  
6   addNumbers(1,2);  
7   print('First Hello!');  
8   print("Second Hello!");  
9   addNumbers(1,2.9);  
10 }  
11
```

Run

Console

3

Documentation

error line 9 • The argument type 'double' can't be assigned to the parameter type 'int'. (view docs)



DartPad

New Pad

Reset

Format

Install SDK

magenta-tulip-2609

Samples



```
1 addNumbers(double num1, double num2) {  
2   print(num1 + num2);  
3 }  
4  
5 void main() {  
6   addNumbers(1,2);  
7   print('First Hello!');  
8   print("Second Hello!");  
9   addNumbers(1,2.9);  
10 }  
11
```

Run

Console

3

Documentation

abstract class double extends num

A double-precision floating point number.

Representation of Dart doubles containing double specific constants and operations and specializations of operations inherited from `num`. Dart doubles are 64-bit floating-point numbers as specified in the IEEE 754 standard.

The `double` type is contagious. Operations on `double`s return `double` results.

It is a compile-time error for a class to attempt to extend or implement `double`.



Privacy notice Send feedback



Null Safety

no issues Based on Flutter 2.5.3 Dart SDK 2.14.4





DartPad

New Pad



Format



Install SDK

magenta-tulip-2609

Samples



```
1 void addNumbers(double num1, double num2) {  
2   print(num1 + num2);  
3 }  
4  
5 void main() {  
6   addNumbers(1,2);  
7   print('First Hello!');  
8   print("Second Hello!");  
9   print(addNumbers(1,2.9));  
10 }  
11
```

Run

Console

3

Documentation

error

line 9 • This expression has a type of 'void' so its value can't be used. [\(view docs\)](#)

Try checking to see if you're using the correct API; there might be a function or call that returns void you didn't expect. Also check type parameters and variables which might also be void.



Privacy notice

Send feedback



Null Safety

1 issue [hide](#)

Based on Flutter 2.5.3 Dart SDK 2.14.4



Windows install | Flutter DartPad https://www.dartpad.dev/?null_safety=true

DartPad New Pad Reset Format Install SDK magenta-tulip-2609 Samples ...

```
1 double addNumbers(double num1, double num2) {  
2     // print(num1 + num2);  
3     return num1 + num2;  
4 }  
5  
6 void main() {  
7     // addNumbers(1,2);  
8     print('First Hello!');  
9     print("Second Hello!");  
10    print(addNumbers(1,2.9));  
11 }  
12
```

Run

Console

```
First Hello!  
Second Hello!  
3.9
```

Documentation

Privacy notice Send feedback Null Safety no issues Based on Flutter 2.5.3 Dart SDK 2.14.4

MORE DART BASICS

```
1 double addNumbers(double num1, double num2) {  
2   // print(num1 + num2);  
3   return num1 + num2;  
4 }  
5  
6 void main() {  
7   firstResult = addNumbers(1,2);  
8   print('First Hello!');  
9   print("Second Hello!");  
10  print(addNumbers(1,2.9));  
11 }  
12 }  
13 }
```

Run

Console

```
First Hello!  
Second Hello!  
3.9
```

Documentation

double addNumbers(double num1, double num2)

error line 7 • Undefined name 'firstResult'. (view docs)

Try correcting the name to one that is defined, or defining the name.



DartPad

New Pad



Reset



Format



Install SDK

magenta-tulip-2609

Samples



```
1 double addNumbers(double num1, double num2) {  
2     // print(num1 + num2);  
3     return num1 + num2;  
4 }  
5  
6 void main() {  
7     print('First Hello!');  
8     print("Second Hello!");  
9  
10    var firstResult = addNumbers(1,2);  
11    firstResult = addNumbers(1,2.9);  
12    print(firstResult + 1);  
13 }  
14
```

▶ Run

Console

```
First Hello!  
Second Hello!  
4.9
```

Documentation



DartPad

New Pad



Format

Install SDK

magenta-tulip-2609

Samples



```
1 double addNumbers(double num1, double num2) {  
2     // print(num1 + num2);  
3     return num1 + num2;  
4 }  
5  
6 void main() {  
7     print('First Hello!');  
8     print("Second Hello!");  
9  
10    double firstResult = addNumbers(1,2);  
11    firstResult = addNumbers(1,2.9);  
12    print(firstResult + 1);  
13 }  
14
```

▶ Run

Console

```
First Hello!  
Second Hello!  
4.9
```

Documentation

Windows install | Flutter DartPad https://www.dartpad.dev/?null_safety=true

DartPad New Pad Reset Format Install SDK magenta-tulip-2609 Samples ::

```
1 class Person {  
2     String name = "Rizqi";  
3     int age = 30;  
4 }  
5  
6 double addNumbers(double num1, double num2) {  
7     // print(num1 + num2);  
8     return num1 + num2;  
9 }  
10  
11 void main() {  
12     print('First Hello!');  
13     print("Second Hello!");  
14  
15     double firstResult = addNumbers(1,2);  
16     firstResult = addNumbers(1,2.9);  
17     print(firstResult + 1);  
18  
19     var p1 = new Person();  
20     var p2 = Person();  
21  
22     print(p1);  
23     print(p2.name);  
24     print(p2.age);  
25  
26     p2.name = "Robert";  
27     print(p2.name);  
28 }  
29
```

Run

Console

```
First Hello!  
Second Hello!  
4.9  
Instance of 'Person'  
Rizqi  
30  
Robert
```

Documentation

int age

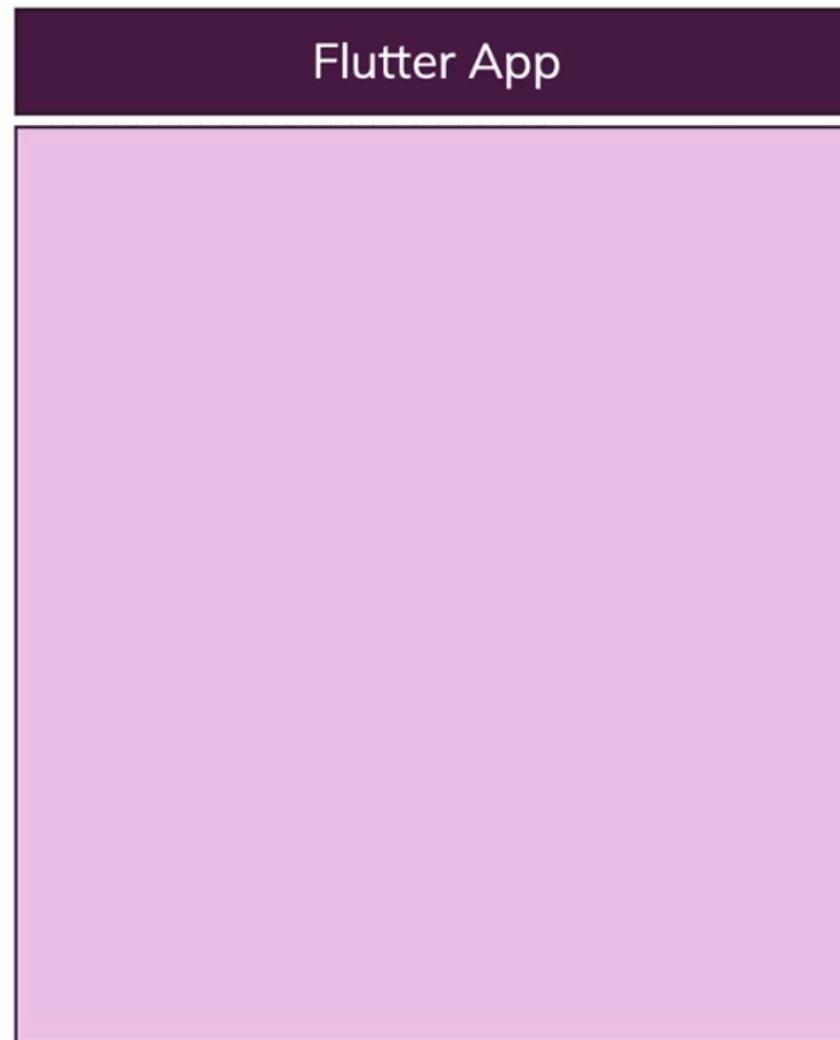
info line 19 • Unnecessary new keyword. (view docs)

Privacy notice Send feedback Null Safety 1 issue hide Based on Flutter 2.5.3 Dart SDK 2.14.4

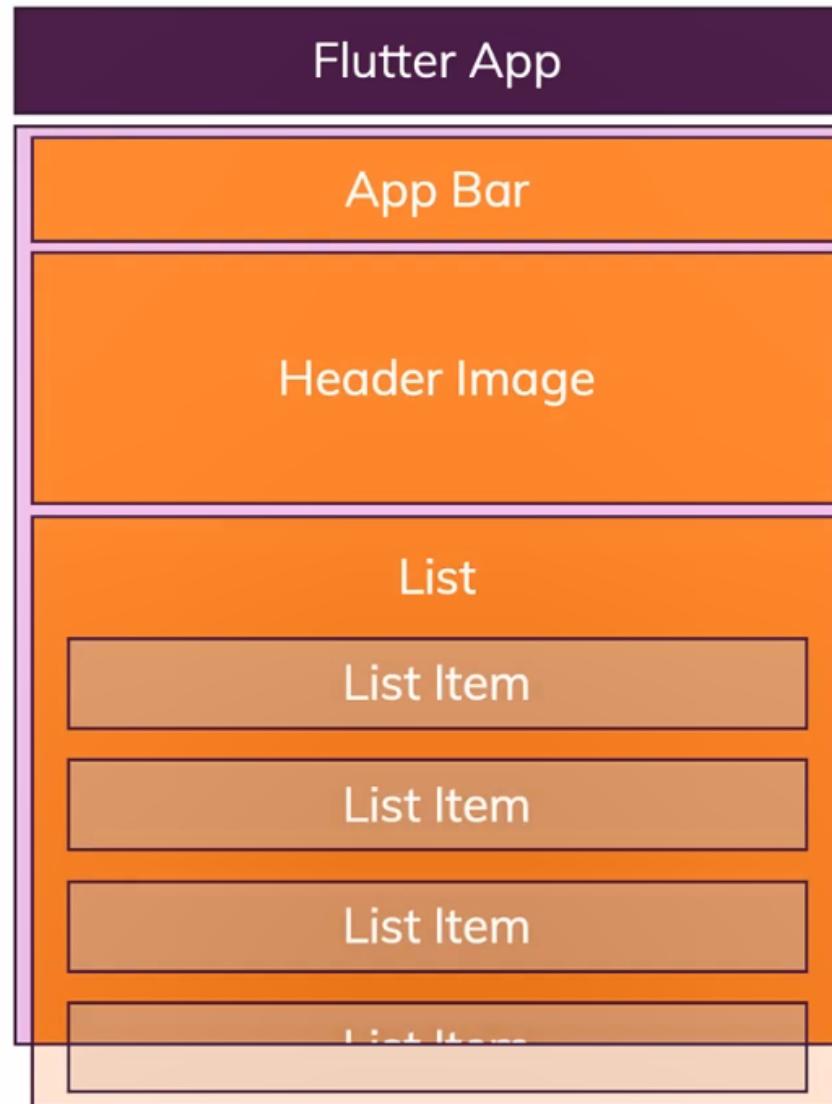
BUILDING AN APP FROM SCRATCH

dari awal

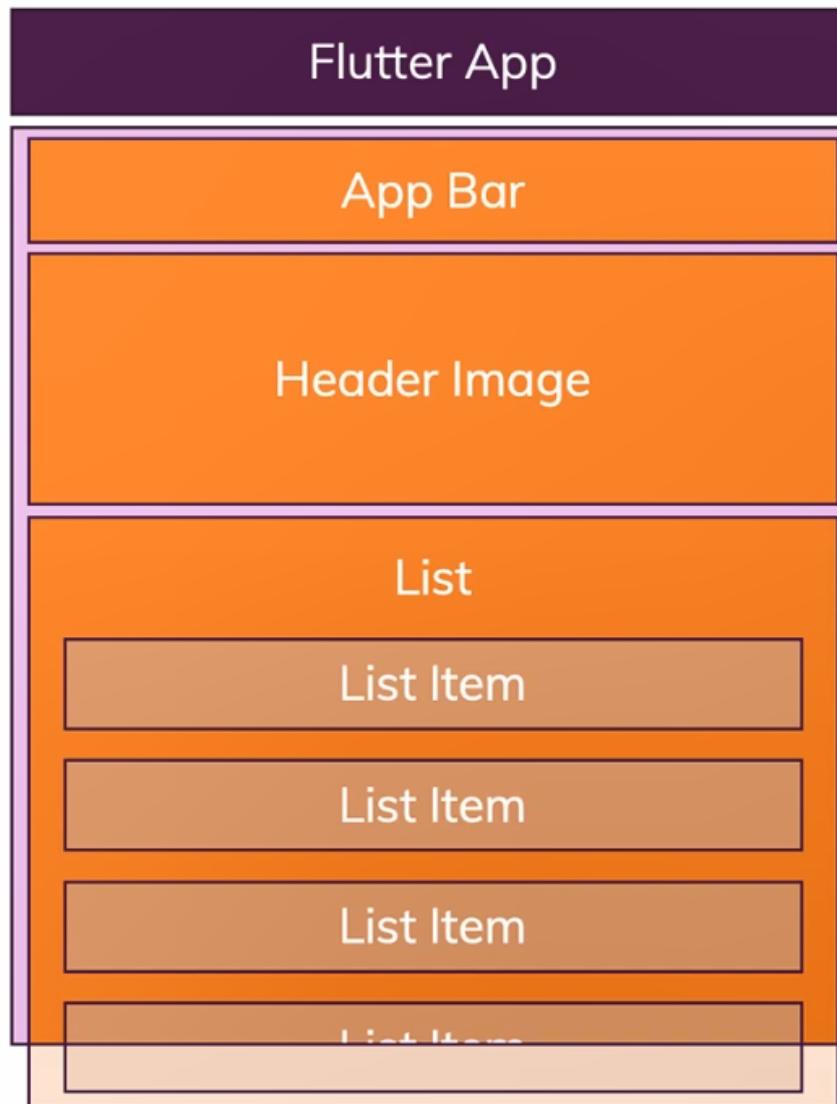
It's All About Widgets



It's All About Widgets

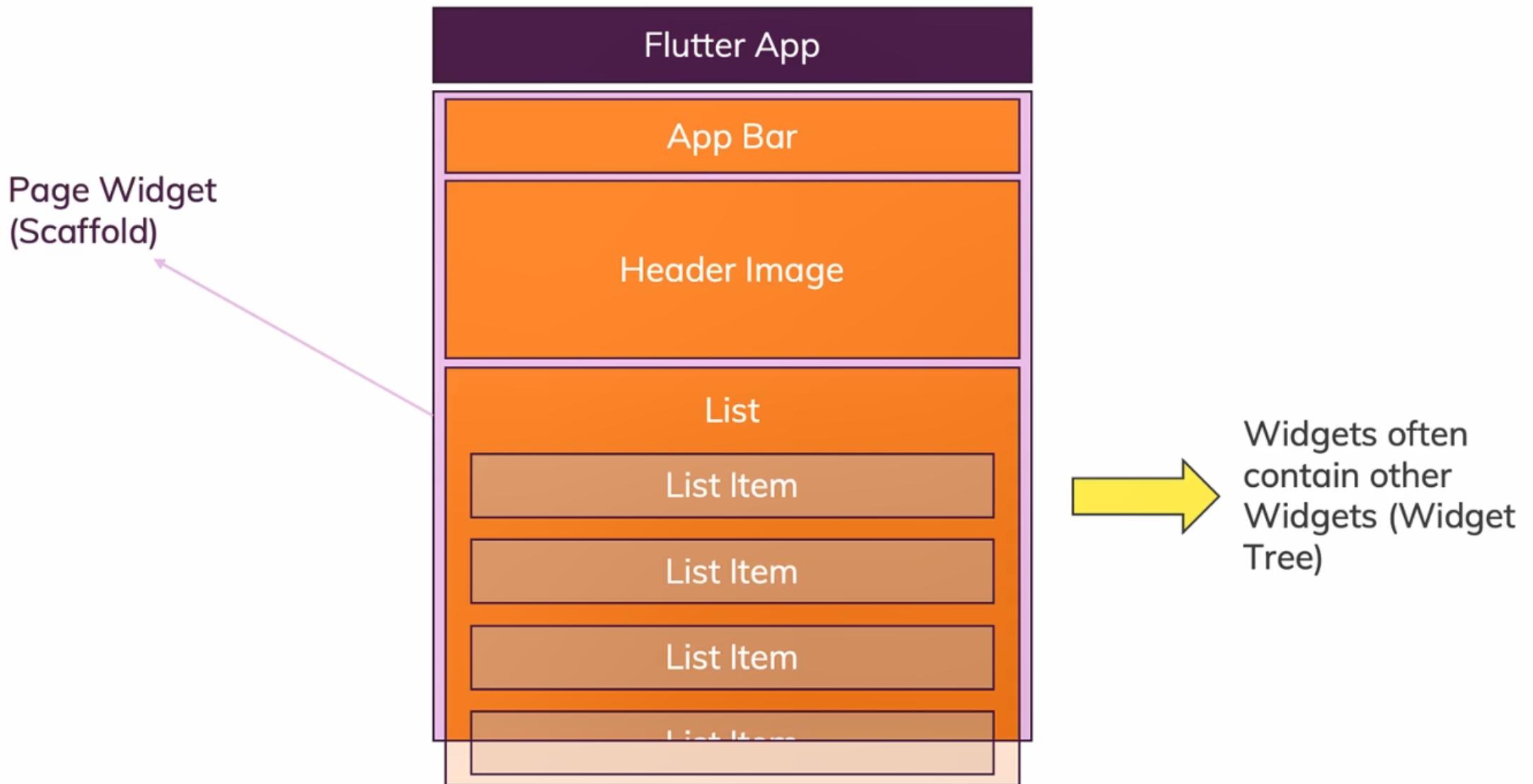


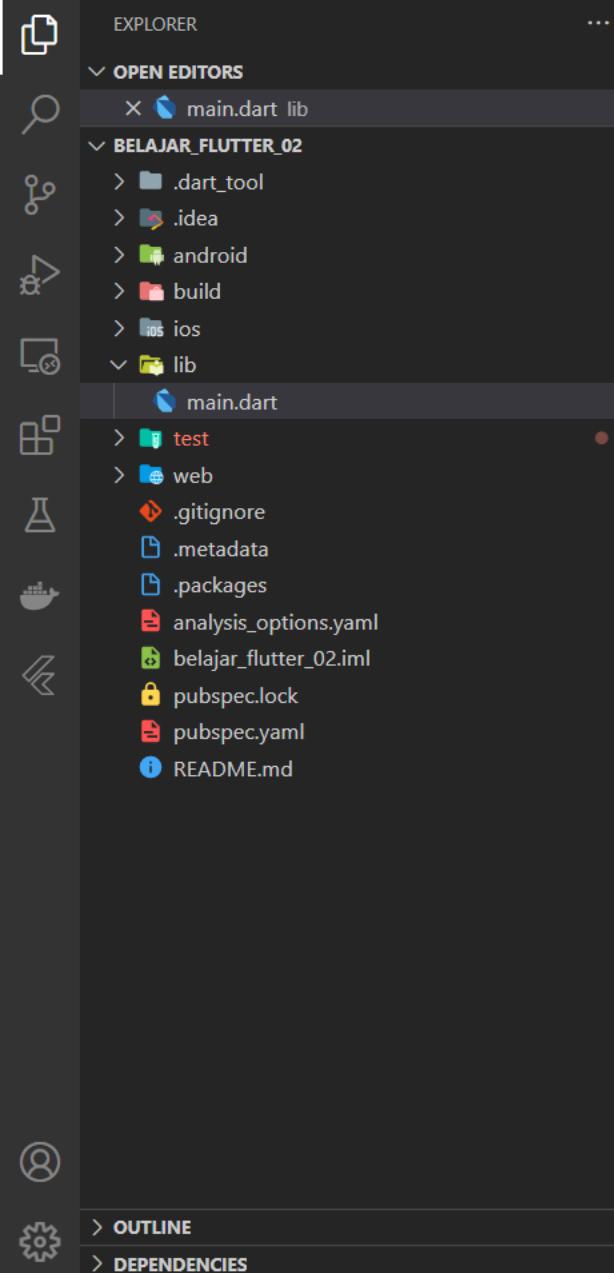
It's All About Widgets



Widgets often
contain other
Widgets (Widget
Tree)

It's All About Widgets





main.dart x

lib > main.dart > ...

```
1 import 'package:flutter/material.dart';
2
3 void main() {}
4
5 class MyApp extends StatelessWidget {
6     @override
7     Widget build(BuildContext context) {
8         return MaterialApp(
9             home: Text("Welcome to Banyuwangi"),
10        ); // MaterialApp
11    }
12 }
13
```

RUNNING THE APP ON AN EMULATOR

File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X widget_test.dart lib > main.dart > ...

OPEN EDITORS main.dart lib widget_test.dart test

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

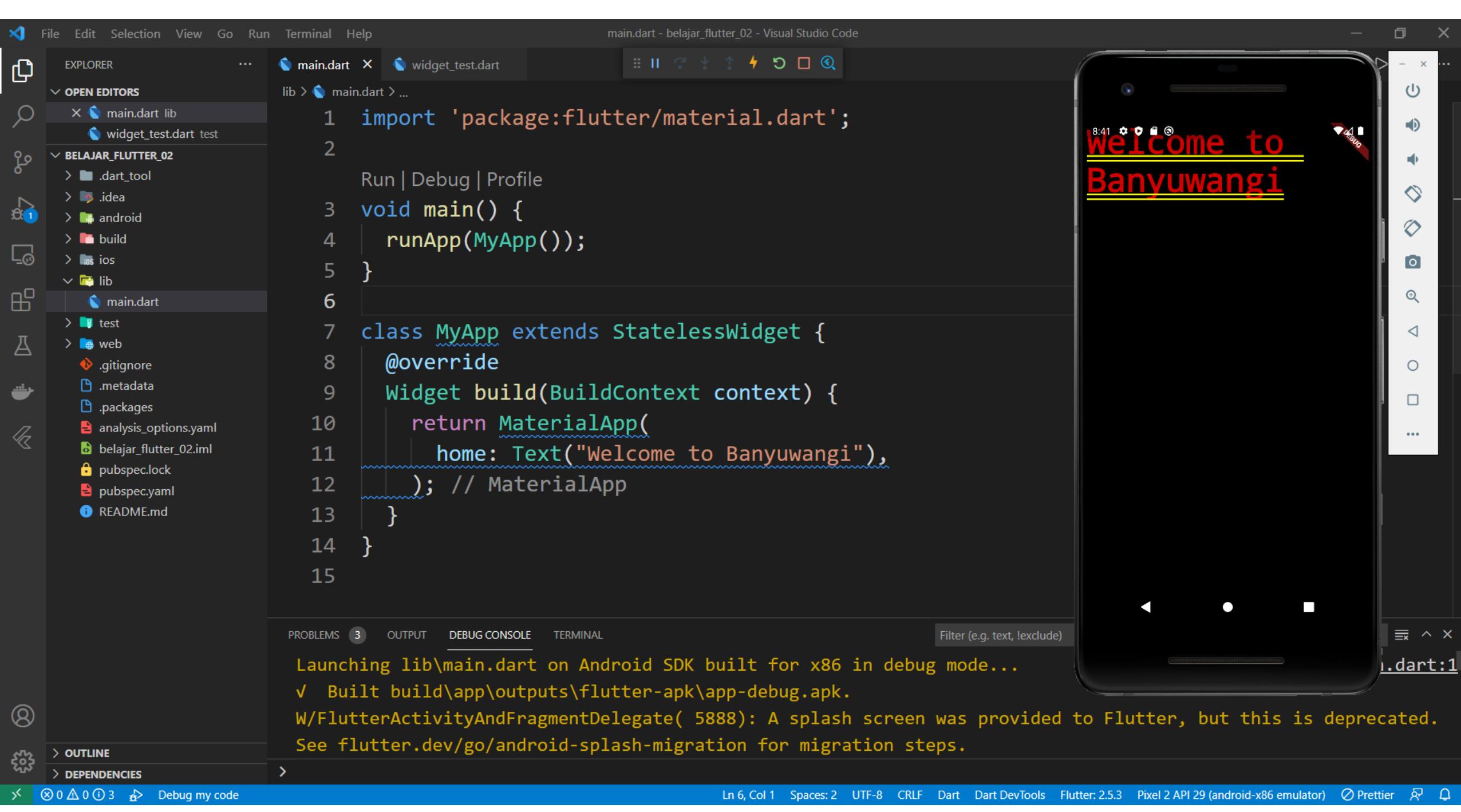
1 import 'package:flutter/material.dart';
2
3 void main() {
4 runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8 @override
9 Widget build(BuildContext context) {
10 return MaterialApp(
11 home: Text("Welcome to Banyuwangi"),
12); // MaterialApp
13 }
14 }
15

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, | exclude)

Launching lib\main.dart on Android SDK built for x86 in debug mode...
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
W/FlutterActivityAndFragmentDelegate(5888): A splash screen was provided to Flutter, but this is deprecated.
See flutter.dev/go/android-splash-migration for migration steps.

Ln 6, Col 1 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

0 △ 0 ① 3 Debug my code



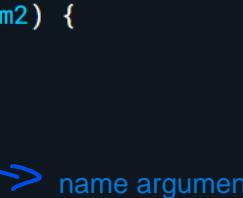
CLASS CONSTRUCTORS AND NAMED ARGUMENTS



```
1 class Person {  
2     String name = "Rizqi";  
3     int age = 30;  
4 }  
5  
6 double addNumbers(double num1, double num2) {  
7     // print(num1 + num2);  
8     return num1 + num2;  
9 }  
10  
11 void main() {  
12     print('First Hello!');  
13     print("Second Hello!");  
14  
15     double firstResult = addNumbers(1,2);  
16     firstResult = addNumbers(1,2.9);  
17     print(firstResult + 1);  
18  
19     var p1 = new Person('Rizqi', 30);  
20     var p2 = Person('Robert', 45);  
21  
22     print(p1);  
23     print(p2.name);  
24     print(p2.age);  
25  
26     p2.name = "Robert";  
27     print(p2.name);  
28 }  
29
```

Run

Console

name argument

Documentation

- info line 19 • Unnecessary new keyword. (view docs)
- error line 19 • Too many positional arguments: 0 expected, but 2 found. (view docs)
Try removing the extra arguments.
- error line 20 • Too many positional arguments: 0 expected, but 2 found. (view docs)
Try removing the extra arguments.

```
1 class Person {  
2   String name = "Anonymous";  
3   int age = -1;  
4   Person(String inputName, int inputAge) {  
5     _____  
6   }  
7 }  
8  
9 double addNumbers(double num1, double num2) {  
10  // print(num1 + num2);  
11  return num1 + num2;  
12 }  
13  
14 void main() {  
15  print('First Hello!');  
16  print("Second Hello!");  
17  
18  double firstResult = addNumbers(1,2);  
19  firstResult = addNumbers(1,2.9);  
20  print(firstResult + 1);  
21  
22  var p1 = new Person('Rizqi', 30);  
23  var p2 = Person('Robert', 45);  
24  
25  print(p2.name);  
26  print(p2.age);  
27  
28  p2.name = "Robert";  
29  print(p2.name);  
30 }  
31 }
```

Run

konstruktor

Console

```
First Hello!  
Second Hello!  
4.9  
Anonymous  
-1  
Robert
```

Documentation

- info line 4 • Use `;` instead of `}` for empty constructor bodies. ([view docs](#))
- info line 22 • The value of the local variable 'p1' isn't used. ([view docs](#))
Try removing the variable or using it.
- info line 22 • Unnecessary new keyword. ([view docs](#))



DartPad New Pad Reset Format Install SDK

magenta-tulip-2609

Samples

⋮

```
6 class Person {  
7   String name = "Anonymous";  
8   int age = -1;  
9   Person(String inputName, int inputAge) {  
10     name = inputName;  
11     age = inputAge;  
12   }  
13 }  
14  
15 void main() {  
16   print('First Hello!');  
17   print("Second Hello!");  
18  
19   double firstResult = addNumbers(1,2);  
20   firstResult = addNumbers(1,2.9);  
21   print(firstResult + 1);  
22  
23   var p1 = Person('Rizqi', 30);  
24   var p2 = Person('Robert', 45);  
25  
26   print(p2.name);  
27   print(p2.age);  
28  
29   p2.name = "Robert";  
30   print(p2.name);  
31 }  
32 }
```

Run

Console

```
First Hello!  
Second Hello!  
4.9  
Robert  
45  
Robert
```

Important

If you do get a “**non nullable field must be initialized**” error, adjust the code like this:

```
class Person {  
  String? name;  
  int? age;  
  
  ...  
}
```

Documentation

info

line 23 • The value of the local variable 'p1' isn't used. [\(view docs\)](#)

Try removing the variable or using it.



```
6 class Person {  
7   String name = "Anonymous";  
8   int age = -1;  
9   Person({String inputName, int inputAge}) {  
10     name = inputName;  
11     age = inputAge;  
12   }  
13 }  
14  
15 void main() {  
16   print('First Hello!');  
17   print("Second Hello!");  
18  
19   double firstResult = addNumbers(1,2);  
20   firstResult = addNumbers(1,2.9);  
21   print(firstResult + 1);  
22  
23   var p1 = Person(inputName: 'Rizqi', inputAge: 30);  
24   var p2 = Person(inputName: 'Robert', inputAge: 45);  
25  
26   print(p2.name);  
27   print(p2.age);
```

▶ Run

Console

```
First Hello!  
Second Hello!  
4.9  
Robert  
45
```

Documentation

{int inputAge = 0}

info

line 23 • The value of the local variable 'p1' isn't used. [\(view docs\)](#)

Try removing the variable or using it.





```
6 class Person {  
7   String name = "Anonymous";  
8   int age = -1;  
9   Person(String name, int age) {  
10    this.name = name;  
11    this.age = age;  
12  }  
13 }  
14  
15 void main() {  
16   print('First Hello!');  
17   print("Second Hello!");  
18  
19   double firstResult = addNumbers(1,2);  
20   firstResult = addNumbers(1,2.9);  
21   print(firstResult + 1);  
22  
23   var p1 = Person('Rizqi', 30);  
24   var p2 = Person('Robert', 45);  
25  
26   print(p2.name);  
27   print(p2.age);  
28  
29   p2.name = "Robert";  
30   print(p2.name);  
31 }  
32 }
```

Run

Console

```
First Hello!  
Second Hello!  
4.9  
Robert  
45  
Robert
```

Documentation

- infoline 10 • Use initializing formals when possible. ([view docs](#))□
 - infoline 11 • Use initializing formals when possible. ([view docs](#))□
 - infoline 23 • The value of the local variable 'p1' isn't used. ([view docs](#))□
- Try removing the variable or using it.

DartPad

[New Pad](#)[Reset](#)[Format](#)[Install SDK](#)

magenta-tulip-2609

[Samples](#)

```
6 class Person {  
7   String name = "Anonymous";  
8   int age = -1;  
9   Person(this.name, this.age);  
10 }  
11  
12 void main() {  
13   print('First Hello!');  
14   print("Second Hello!");  
15  
16   double firstResult = addNumbers(1,2);  
17   firstResult = addNumbers(1,2.9);  
18   print(firstResult + 1);  
19  
20   var p1 = Person('Rizqi', 30);  
21   var p2 = Person('Robert', 45);  
22  
23   print(p2.name);  
24   print(p2.age);  
25  
26   p2.name = "Robert";  
27   print(p2.name);
```

[Run](#)

Console

```
First Hello!  
Second Hello!  
4.9  
Robert  
45  
Robert
```

Documentation

[info](#)

line 20 · The value of the local variable 'p1' isn't used. [\(view docs\)](#)

Try removing the variable or using it.

BAD:

```
class Point {  
    num x, y;  
    Point(num x, num y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

GOOD:

```
class Point {  
    num x, y;  
    Point(this.x, this.y);  
}
```

BAD:

```
class Point {  
    num x, y;  
    Point({num x, num y}) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

GOOD:

```
class Point {  
    num x, y;  
    Point({this.x, this.y});  
}
```

FIRST SUMMARY AND ADDITIONAL SYNTAX

LEARN FLUTTER AND DART

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- widget_test.dart test

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
- main.dart
- test
- web
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

main.dart X widget_test.dart

lib > main.dart > MyApp > build

```
1 import 'package:flutter/material.dart';
2
3 // void main() {
4 //   runApp(MyApp());
5 // }
6
7 void main() => runApp(MyApp());
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       home: Text("Welcome to Banyuwangi"),
14     ); // MaterialApp
15   }
16 }
```

Run | Debug | Profile

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

Filter (e.g. text, !exclude)

Launching lib\main.dart on Android SDK built for x86 in debug mode...

V Built build\app\outputs\flutter-apk\app-debug.apk.

W/FlutterActivityAndFragmentDelegate(5888): A splash screen was provided to Flutter, but this is deprecated.
See flutter.dev/go/android-splash-migration for migration steps.

Ln 14, Col 7 Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

The image shows a dark-themed Visual Studio Code interface. On the left is the Explorer sidebar with project files like 'main.dart lib', 'widget_test.dart test', and 'BELAJAR_FLUTTER_02' folder contents. The main editor area shows Dart code for a Flutter application. To the right of the editor is a mobile phone emulator displaying the app's UI with the text 'Welcome to Banyuwangi'. The bottom status bar provides build information and file details.

BUILDING A WIDGET TREE

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- BELAJAR_FLUTTER_02
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - main.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md

Run | Debug | Profile

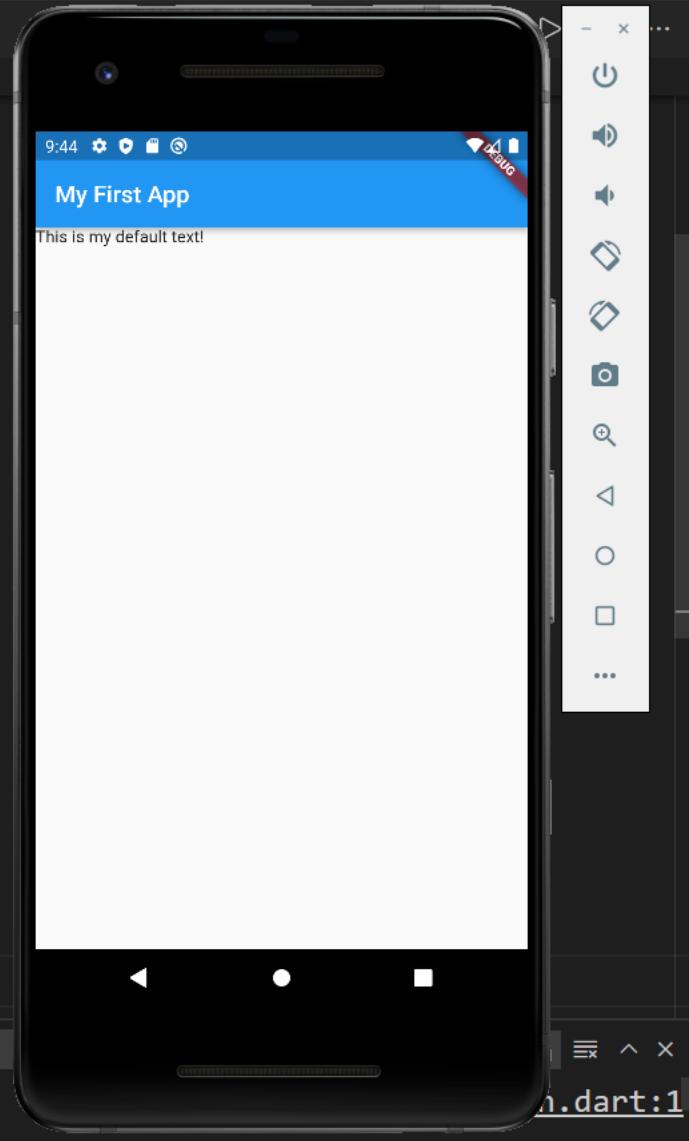
```
7 void main() => runApp(MyApp());  
8  
9 class MyApp extends StatelessWidget {  
10   @override  
11   Widget build(BuildContext context) {  
12     return MaterialApp(  
13       home: Scaffold(  
14         appBar: AppBar(  
15           title: Text("My First App"),  
16         ), // AppBar  
17         body: Text('This is my default text!'),  
18       ), // Scaffold  
19     ); // MaterialApp  
20   }  
21 }  
22 }
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

Filter (e.g. text, !exclude)

Launching lib\main.dart on Android SDK built for x86 in debug mode...
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
W/FlutterActivityAndFragmentDelegate(5888): A splash screen was provided to Flutter, but this is deprecated.
See flutter.dev/go/android-splash-migration for migration steps.

Ln 22, Col 1 Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier



VISIBLE (INPUT/OUTPUT) AND INVISIBLE (LAYOUT CONTROL) WIDGETS

Different Types of Widgets

Output & Input (Visible)

RaisedButton(), Text(), Card(), ...

Drawn onto the screen: “What the user sees”

Different Types of Widgets

Output & Input (Visible)

RaisedButton(), Text(), Card(), ...

Drawn onto the screen: “What the user sees”

Layout & Control (Invisible)

Row(), Column(), ListView(), ...

Give your app structure and control how visible widgets are drawn onto the screen (indirectly visible)

Different Types of Widgets

kelihatan di UI

Output & Input (Visible)

RaisedButton(), Text(), Card(), ...

Layout & Control (Invisible)

Row(), Column(), ListView(), ...

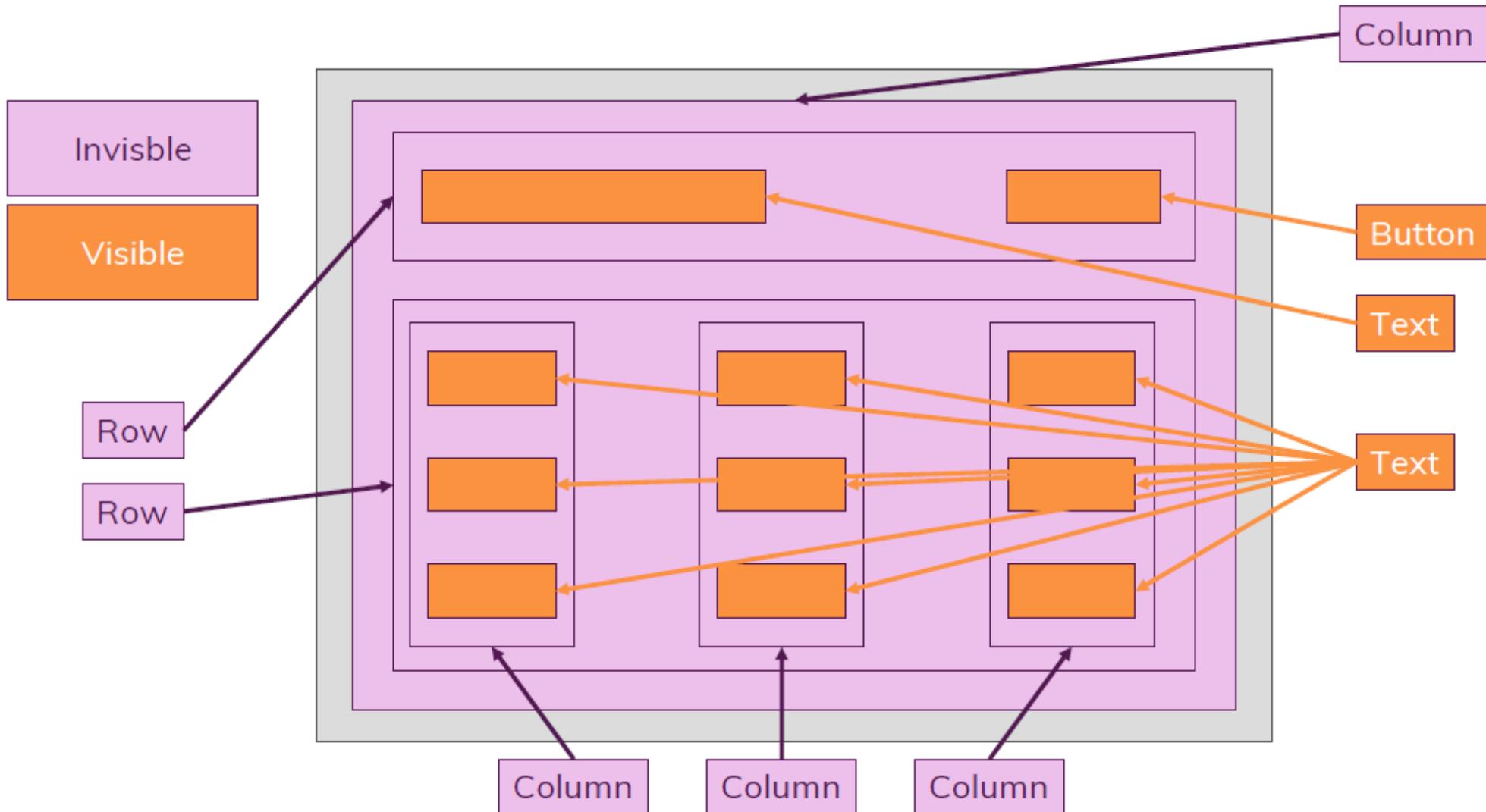
Container()

wadah

Drawn onto the screen: “What the user sees”

Give your app structure and control how visible widgets are drawn onto the screen (indirectly visible)

Visible vs Invisible Widgets



ADDING LAYOUT WIDGETS

file - Settings - belajar_flutter_02 - Visual Studio Code

EXPLORER main.dart 1 ... Settings

OPEN EDITORS 1 UNSAVED main.dart lib 1 Settings

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart 1 test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

deprecated file-reference-setting User Workspace 6 Settings Found Turn on Settings Sync

Text Editor (2) Suggestions (1)

Application (1) Telemetry (1)

Extensions (3) Docker (1)

Go (1) Python (1)

Editor: Show Deprecated Controls strikethrough deprecated variables.

Editor > Suggest: Show Deprecated When enabled IntelliSense shows **deprecated**-suggestions.

Telemetry: Telemetry Level Controls all core and first party extension telemetry. This helps us to better understand how Visual Studio Code is performing, where improvements need to be made, and how features are being used. Read more about the [data we collect](#) and our [privacy statement](#). A full restart of the application is necessary for crash reporting changes to take effect.

The following table outlines the data sent with each setting:

	Crash Reports	Error	Telemetry	Usage Data
all	✓	✓	✓	✓
error	✓	✓	-	-
crash	✓	-	-	-
off	-	-	-	-

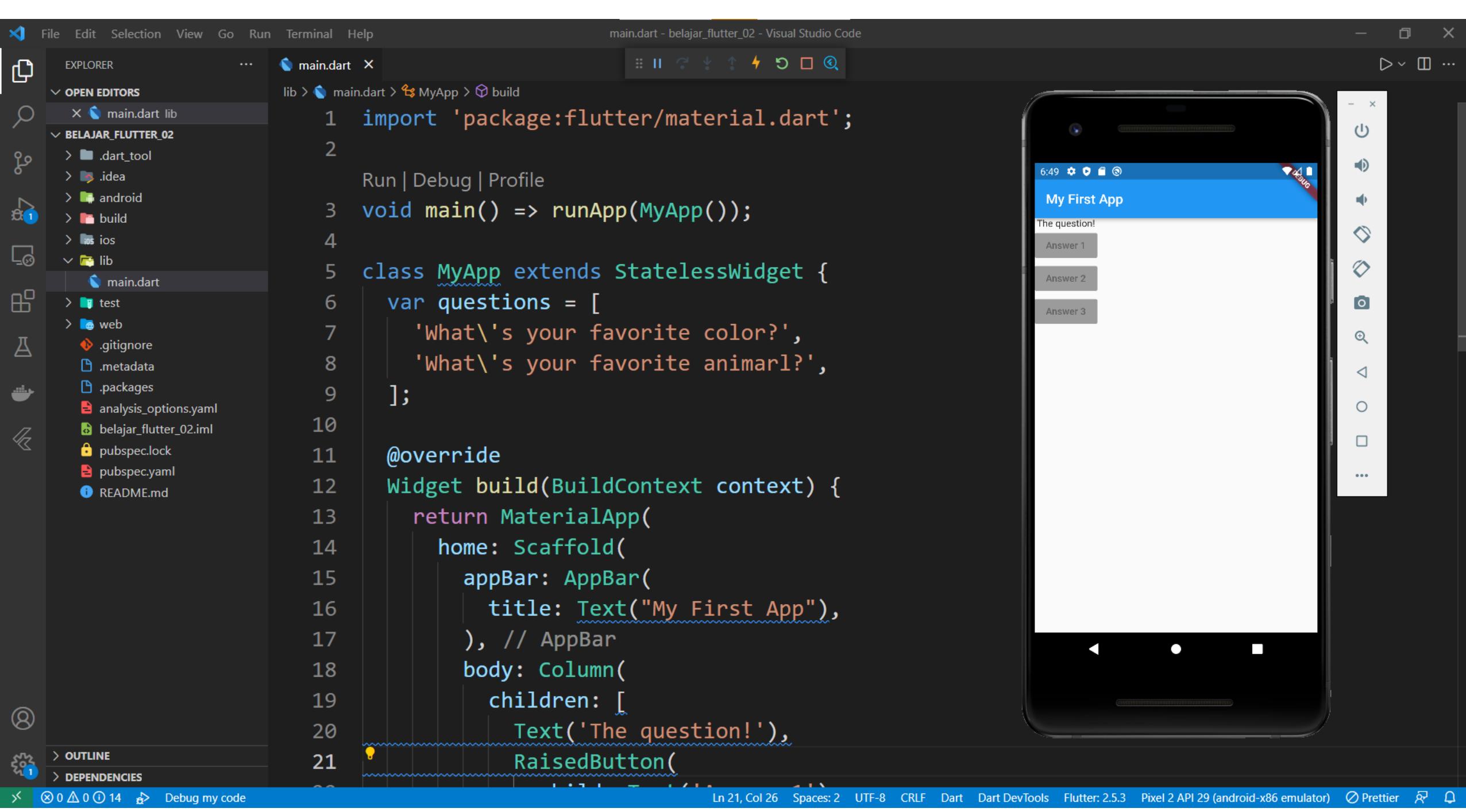
Note: If this setting is 'off', no telemetry will be sent regardless of other telemetry settings. If this setting is set to anything except 'off' and telemetry is disabled with deprecated settings, no telemetry will be sent.

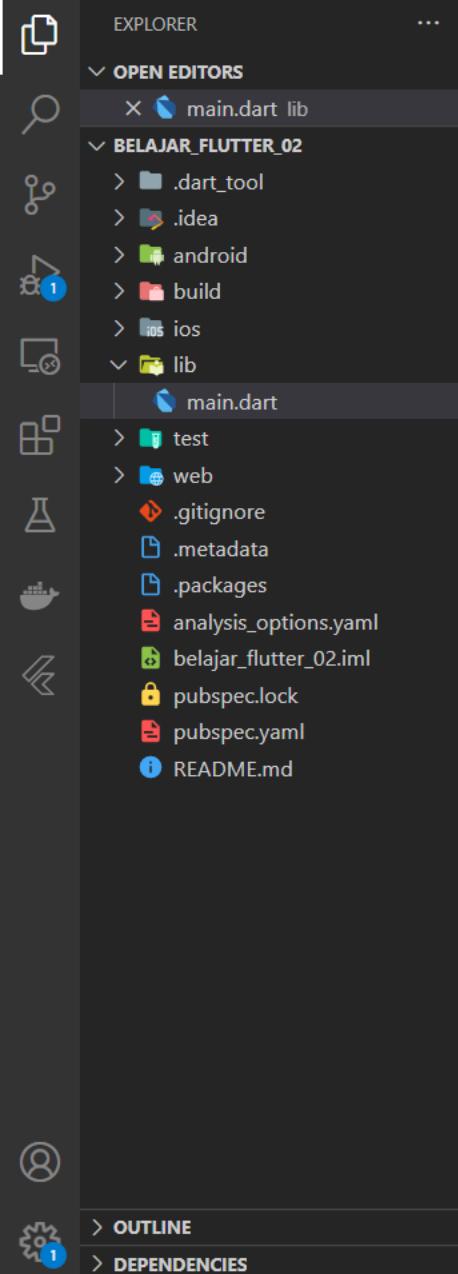
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude) Flutter (Pixel 2 API 29) ^ X

I/flutter (5888): Answer 3 chosen
I/flutter (5888): Answer 2 chosen!
I/flutter (5888): Answer 3 chosen
I/flutter (5888): Answer 2 chosen!

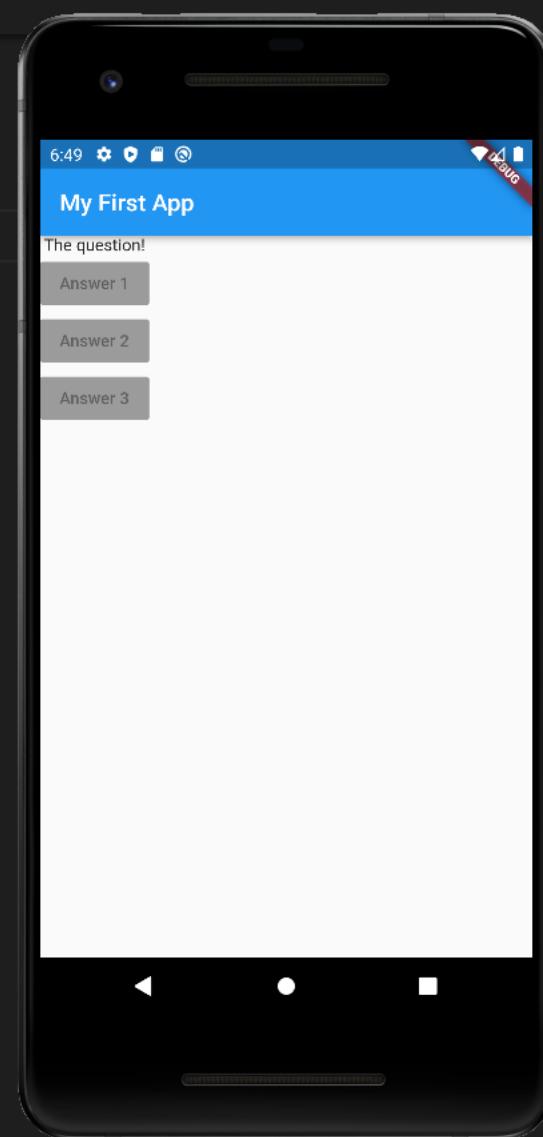
OUTLINE
DEPENDENCIES

Debug my code Dart DevTools Pixel 2 API 29 (android-x86 emulator) 🔍 🔍





```
18     body: Column(
19         children: [
20             Text('The question!'),
21             RaisedButton(
22                 child: Text('Answer 1'),
23                 onPressed: null,
24             ), // RaisedButton
25             RaisedButton(
26                 child: Text('Answer 2'),
27                 onPressed: null,
28             ), // RaisedButton
29             RaisedButton(
30                 child: Text('Answer 3'),
31                 onPressed: null,
32             ), // RaisedButton
33         ],
34     ), // Column
35 ), // Scaffold
36 ); // MaterialApp
37 }
38 }
```



CONNECTING FUNCTIONS AND BUTTONS

A screenshot of the Visual Studio Code interface, showing a Dart file named `main.dart` in the editor. The code implements a simple application with a list of questions and a printing function.

```
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
  var questions = [  
    'What\'s your favorite color?',  
    'What\'s your favorite animal?',  
  ];  
  
  void answerQuestion() {  
    print('Answer Chosen');  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text("My First App"),  
        ), // AppBar  
        body: Column(  
          children: [  
            Text('The question!'),  
          ],  
        ),  
      ),  
    );  
  }  
}  
  
void main() => runApp(MyApp());
```

The Explorer sidebar on the left shows the project structure, including `main.dart` in the `lib` directory. The bottom status bar displays file statistics: Ln 35, Col 40, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier, and a connection icon.

A screenshot of the Visual Studio Code interface, showing the code editor with Dart code for a Flutter application named `MyApp`.

The code defines a `MaterialApp` widget with a `Scaffold` body containing a `Column` with three `RaisedButton` children, each with a `Text` child and an `onPressed` callback.

```
main.dart - belajar_flutter_02 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ...
main.dart X lib > main.dart > MyApp
main.dart lib
BELAJAR_FLUTTER_02
  .dart_tool
  .idea
  android
  build
  ios
  lib
    main.dart
  test
  web
  .gitignore
  .metadata
  .packages
  analysis_options.yaml
  belajar_flutter_02.iml
  pubspec.lock
  pubspec.yaml
  README.md
OUTLINE ...
DEPENDENCIES ...
Ln 42, Col 2 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier ...
```

```
22   body: Column(
23     children: [
24       Text('The question!'),
25       RaisedButton(
26         child: Text('Answer 1'),
27         onPressed: answerQuestion,
28       ), // RaisedButton
29       RaisedButton(
30         child: Text('Answer 2'),
31         onPressed: () => answerQuestion,
32       ), // RaisedButton
33       RaisedButton(
34         child: Text('Answer 3'),
35         onPressed: answerQuestion,
36       ), // RaisedButton
37     ],
38   ), // Column
39   ), // Scaffold
40 ); // MaterialApp
41 }
42 }
43 }
```

ANONYMOUS FUNCTIONS

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X

lib > main.dart > MyApp > build

```
title: Text("My First App"),
), // AppBar
body: Column(
  children: [
    Text('The question!'),
    RaisedButton(
      child: Text('Answer 1'),
      onPressed: answerQuestion,
    ), // RaisedButton
    RaisedButton(
      child: Text('Answer 2'),
      onPressed: () => print('Answer 2 chosen!'),
    ), // RaisedButton
    RaisedButton(
      child: Text('Answer 3'),
      onPressed: () {
        // ...
        print('Answer 3 chosen');
      },
    ), // RaisedButton
  ],
), // Column
```

My First App
The question!
Answer 1
Answer 2
Answer 3

7:04 ⚡ DEBUG

File Explorer

Search

Problems

Output

Terminal

Run

Flutter DevTools

Prettier

Outline

Dependencies

Ln 34, Col 39 Spaces: 2 UTR-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

UPDATING WIDGET DATA (OR USING STATELESSWIDGET INCORRECTLY)

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X lib > main.dart > MyApp > build

OPEN EDITORS main.dart lib

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

OUTLINE DEPENDENCIES

4

```
5 class MyApp extends StatelessWidget {  
6     var questionIndex = 0;  
7  
8     void answerQuestion() {  
9         questionIndex = questionIndex + 1;  
10        print(questionIndex);  
11    }  
12  
13    var questions = [  
14        'What\'s your favorite color?',  
15        'What\'s your favorite animal?',  
16    ];  
17  
18    @override  
19    Widget build(BuildContext context) {  
20        return MaterialApp(  
21            home: Scaffold(  
22                appBar: AppBar(  
23                    title: Text("My First App"),  
24                ), // AppBar  
25                body: Column(  
26                    children: [  
27                        Text("What's your favorite color?"),  
28                        ElevatedButton(onPressed: answerQuestion, child: Text("Answer 1")),  
29                        ElevatedButton(onPressed: answerQuestion, child: Text("Answer 2")),  
30                        ElevatedButton(onPressed: answerQuestion, child: Text("Answer 3")),  
31                    ],  
32                ),  
33            ),  
34        );  
35    }  
36}
```

7:52 My First App What's your favorite color?
Answer 1 Answer 2 Answer 3

File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

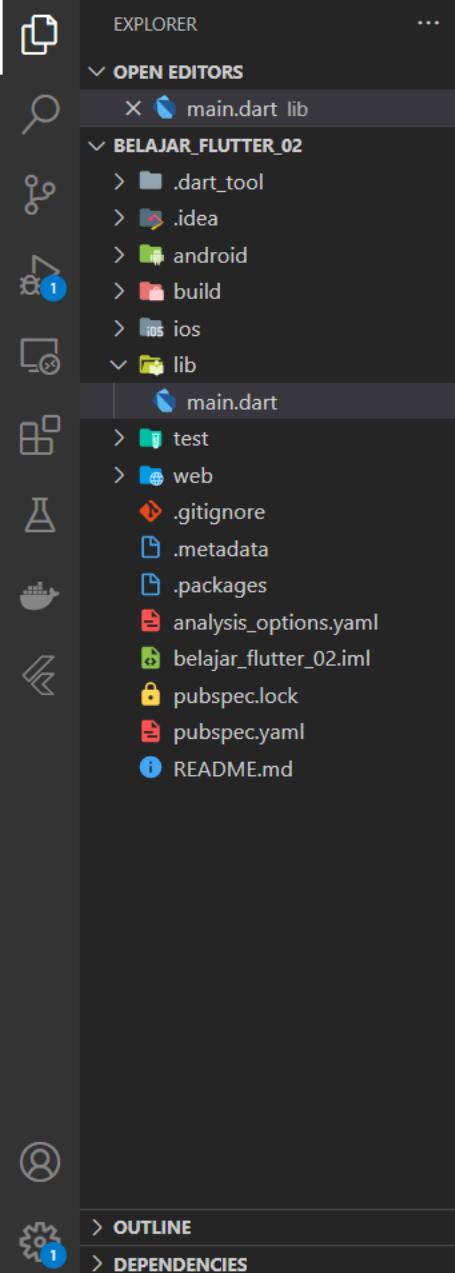
EXPLORER ... main.dart X lib > main.dart > MyApp > build

OPEN EDITORS main.dart lib

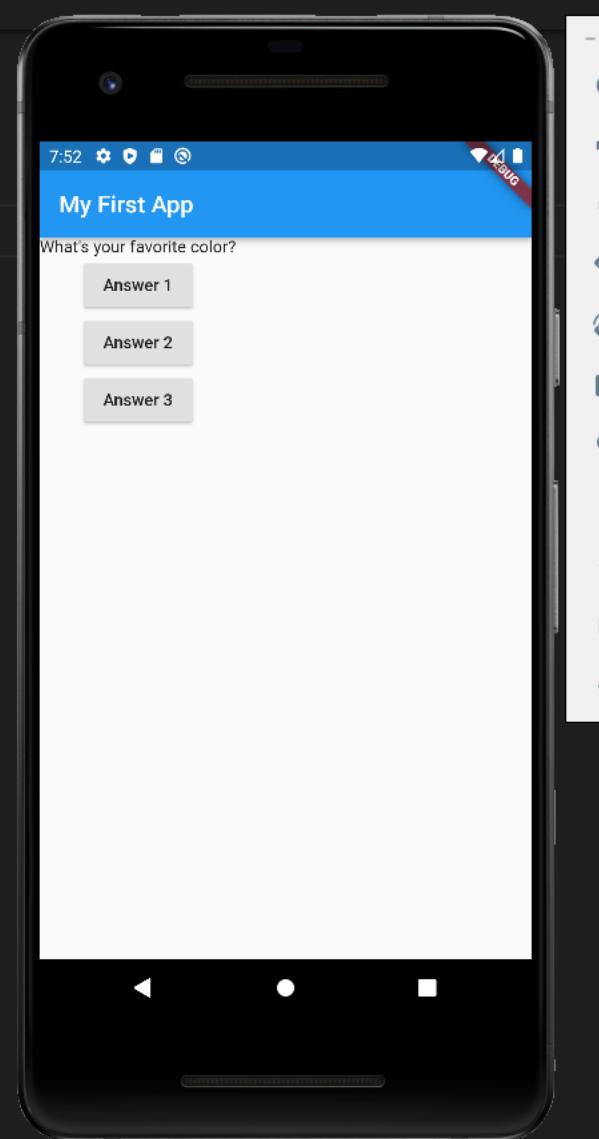
BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

OUTLINE DEPENDENCIES

Ln 27, Col 40 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier



```
24     ), // AppBar
25   body: Column(
26     children: [
27       Text(questions[questionIndex]),
28       RaisedButton(
29         child: Text('Answer 1'),
30         onPressed: answerQuestion,
31       ), // RaisedButton
32       RaisedButton(
33         child: Text('Answer 2'),
34         onPressed: () => print('Answer 2 chosen!'),
35       ), // RaisedButton
36       RaisedButton(
37         child: Text('Answer 3'),
38         onPressed: () {
39           // ...
40           print('Answer 3 chosen');
41         },
42       ), // RaisedButton
43     ],
44   ), // Column
45 ), // Scaffold
```



File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X lib > main.dart > MyApp > build

24), // AppBar
25 body: Column(
26 children: [
27 Text(questions[questionIndex]),
28 RaisedButton(
29 child: Text('Answer 1'),
30 onPressed: answerQuestion,
31), // RaisedButton
32 RaisedButton(
33 child: Text('Answer 2'),
34 onPressed: () => print('Answer 2 chosen!'),
35), // RaisedButton

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude)

Restarted application in 2,683ms.
I/flutter (5888): 1

My First App
What's your favorite color?
Answer 1
Answer 2
Answer 3

Press 'Answer 1'
Why the question didn't change?

OUTLINE
DEPENDENCIES >

Debug my code Ln 27, Col 40 Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

UPDATING CORRECTLY WITH STATEFULWIDGET

Understanding “State”

informasi / data

In General

State is Data/ Information used by your App

Understanding “State”

In General

State is Data/ Information used by your App

App State

Authenticated Users
Loaded Jobs

...

Understanding “State”

In General

State is Data/ Information used by your App

App State

Authenticated Users
Loaded Jobs

...

Widget State

Current User Input
Is a Loading Spinner being shown?

...

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code
- Left Sidebar (Icons):** Explorer, Search, Find, Open Editors, Project Explorer, Test, Web, Git, Run, Outline, Dependencies.
- Left Sidebar (Content):** Shows the project structure under "OPEN EDITORS" and "BELAJAR_FLUTTER_02".
- Central Area:** The main editor window displays the code for "main.dart".
- Code Content:**

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatelessWidget {
6     var questionIndex = 0;
7
8     void answerQuestion() {
9         questionIndex = questionIndex + 1;
10        print(questionIndex);
11    }
12
13     var questions = [
14         'What\'s your favorite color?',
15         'What\'s your favorite animal?',
16     ];
17
18     @override
19     Widget build(BuildContext context) {
20         return MaterialApp(
21             home: Scaffold(
22                 body: Center(
23                     child: Text(questions[questionIndex]),
24                 ),
25             ),
26         );
27     }
28 }
```
- Bottom Status Bar:** Ln 5, Col 36 (15 selected), Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.

Stateless vs Stateful

tdk menyimpan data tertentu

Stateless

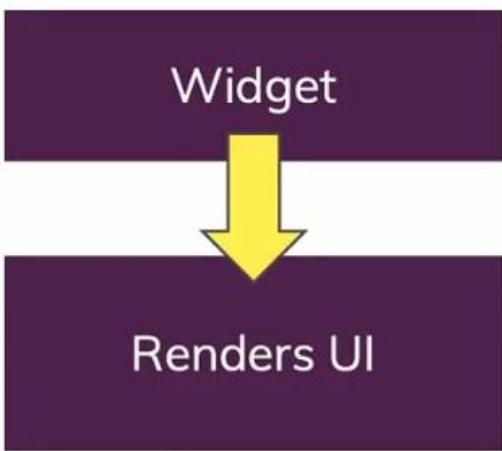
Stateful

Widget

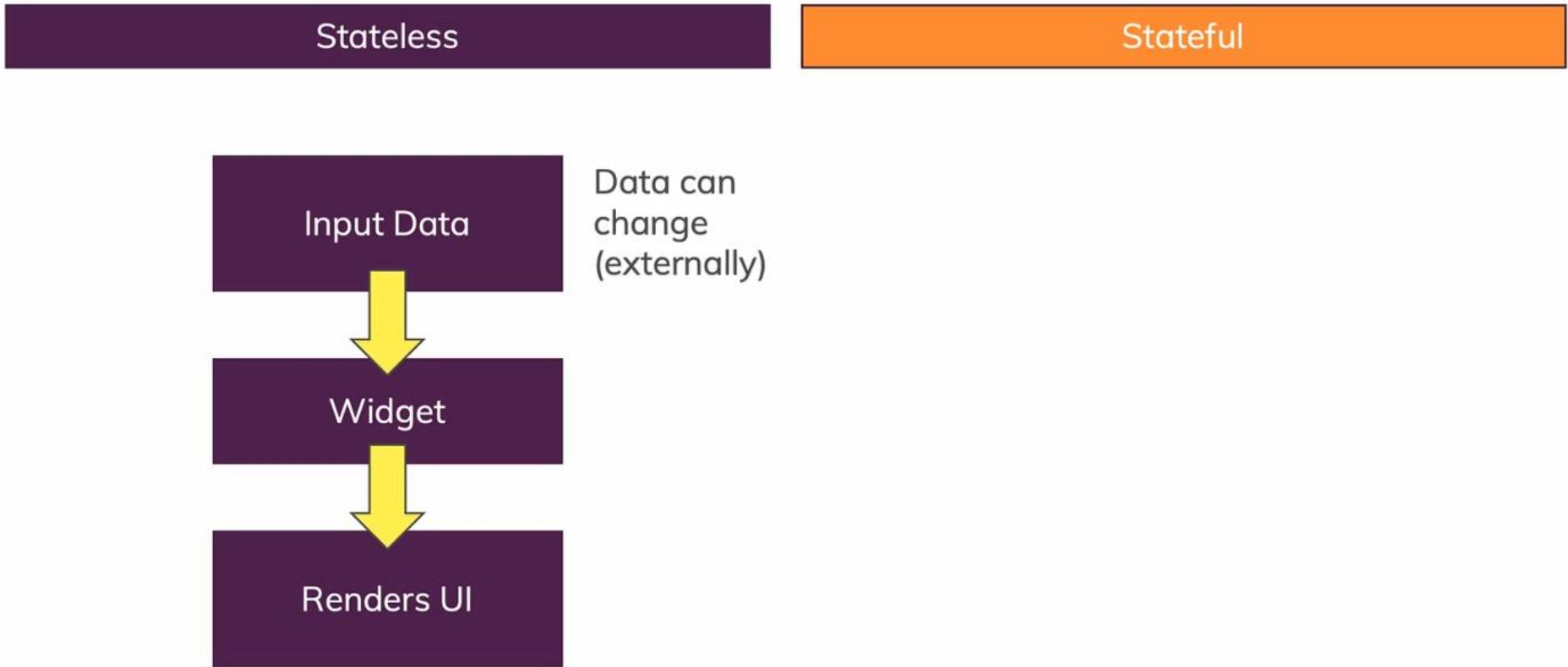
Stateless vs Stateful

Stateless

Stateful



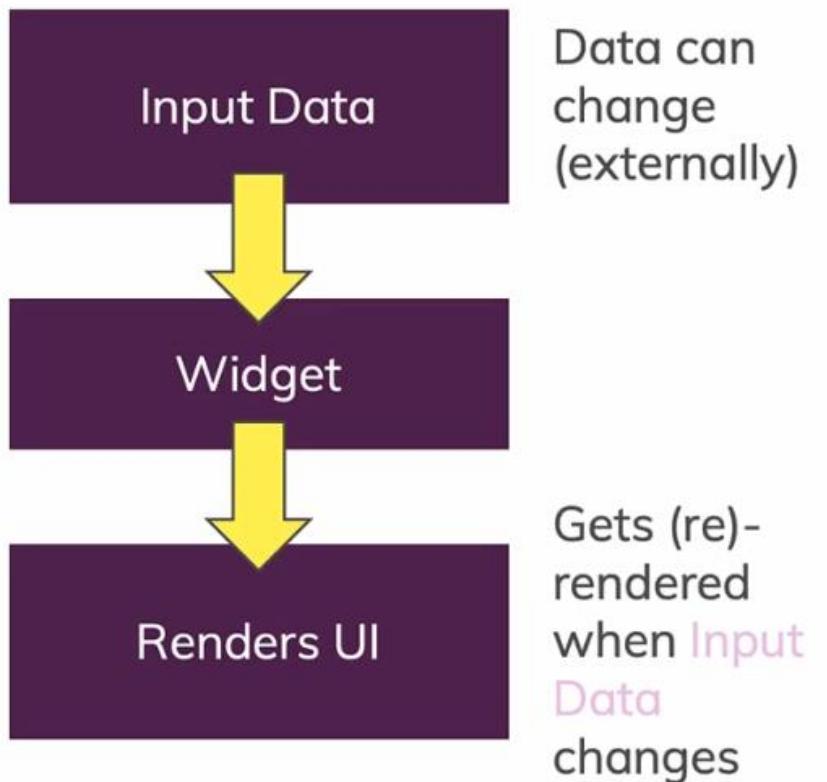
Stateless vs Stateful



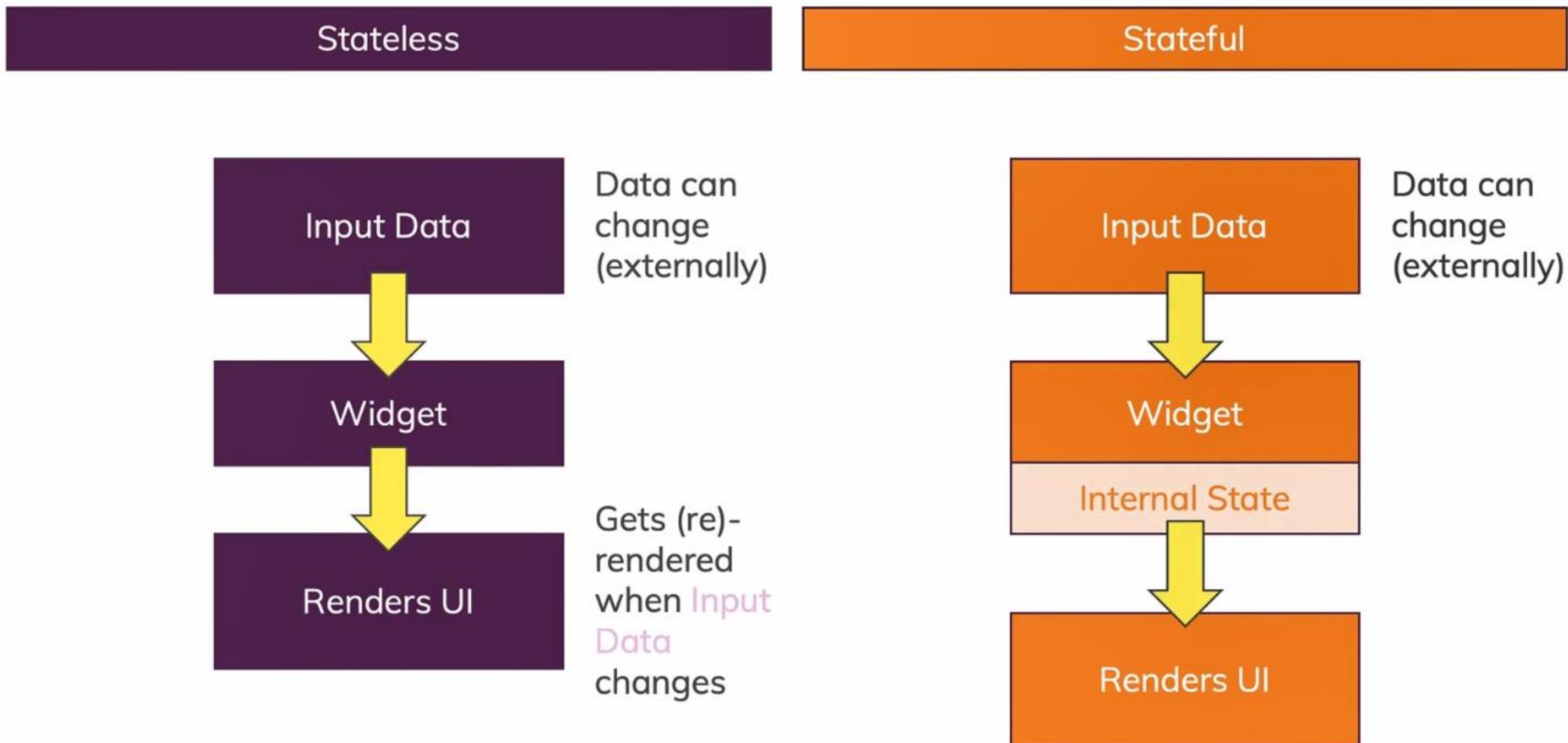
Stateless vs Stateful

Stateless

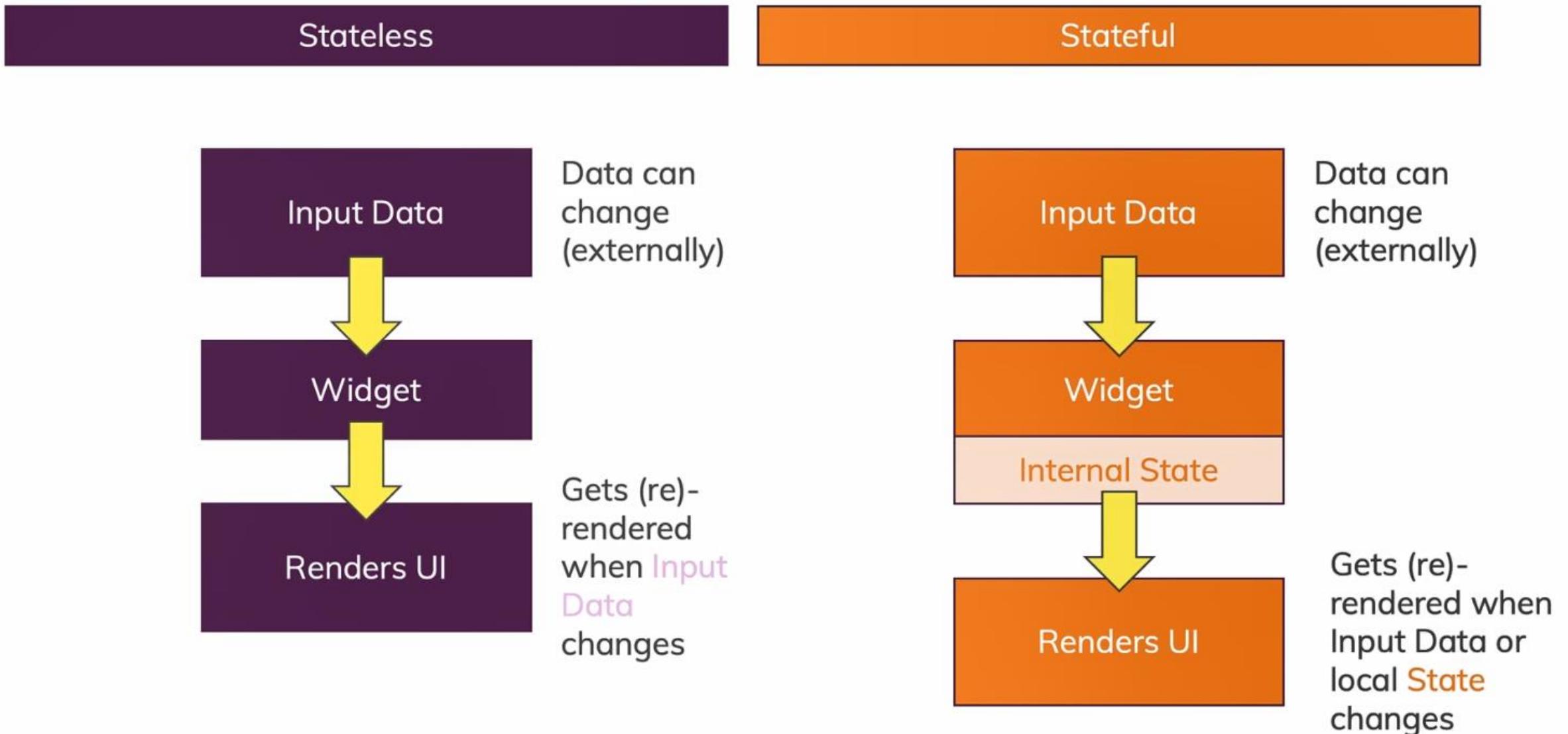
Stateful



Stateless vs Stateful



Stateless vs Stateful



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code
- Explorer View (Left):** Shows the project structure:
 - OPEN EDITORS: main.dart lib
 - BELAJAR_FLUTTER_02:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib:
 - main.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Editor View (Center):** The main.dart file content is displayed.

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatefulWidget {
6   @override
7   State<StatefulWidget> createState() {
8     return MyAppState();
9   }
10 }
11
12 class MyAppState extends State<MyApp> {
13   var questionIndex = 0;
14
15   void answerQuestion() {
16     setState(() {
17       questionIndex = questionIndex + 1;
18     });
19     print(questionIndex);
20   }
21 }
```
- Bottom Status Bar:** Ln 10, Col 2 Spaces: 2 UTF-8 CR Android Emulator - Pixel_2_API_29:5554 Pixel 2 API 29 (android-x86 emulator) Prettier

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X

OPEN EDITORS main.dart lib

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

lib > main.dart > MyApp

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatefulWidget {
6   @override
7   State<StatefulWidget> createState() {
8     return MyAppState();
9   }
10}
11
12
```

Run | Debug | Profile

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude)

Restarted application in 2,102ms.

Ln 10, Col 2 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

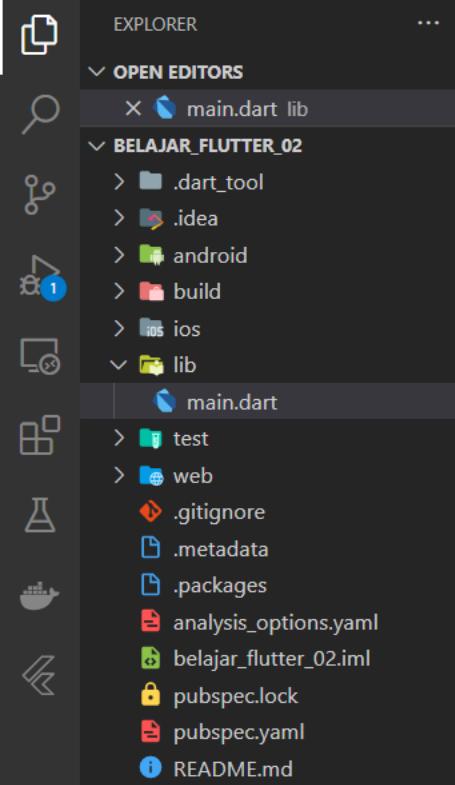
My First App

What's your favorite color?

Answer 1

Answer 2

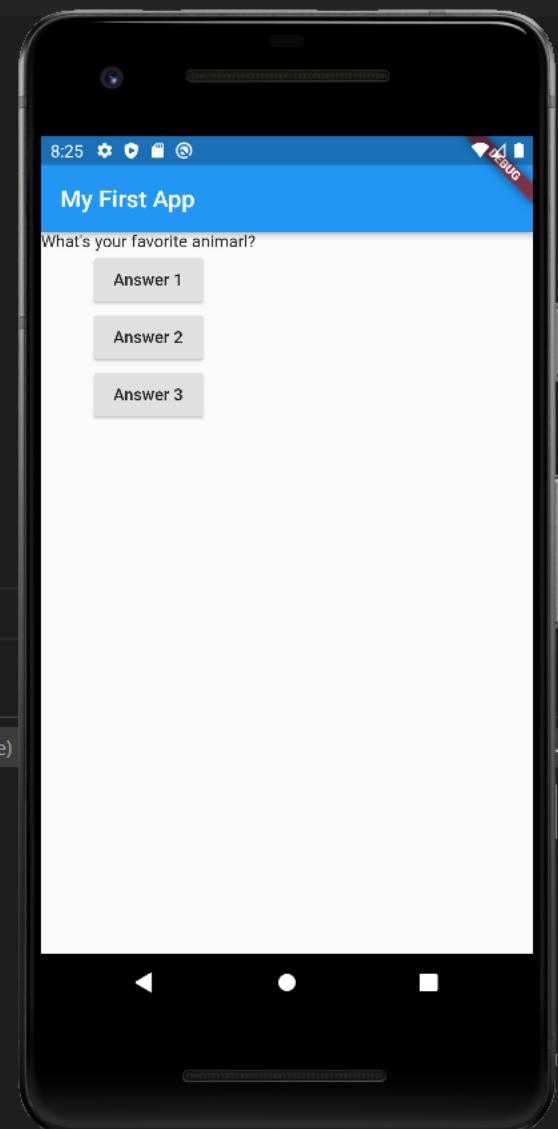
Answer 3



```
main.dart x
lib > main.dart > MyApp
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatefulWidget {
6   @override
7   State<StatefulWidget> createState() {
8     return MyAppState();
9   }
10}
11
12
```

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude)

```
Restarted application in 2,102ms.
I/flutter ( 5888): 1
```



File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X

OPEN EDITORS main.dart lib

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib main.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

lib > main.dart > MyApp

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatefulWidget {
6   @override
7   State<StatefulWidget> createState() {
8     return MyAppState();
9   }
10}
11
12
```

Run | Debug | Profile

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude)

Exception caught by widgets library

The following RangeError was thrown building MyApp(dirty, state: MyAppState#1f3d0): RangeError (index): Invalid value: Not in inclusive range 0..1: 2

The relevant error-causing widget was

MyApp

When the exception was thrown, this was the stack

#0 List.[] (dart:core-patch/growable_array.dart:254:60)

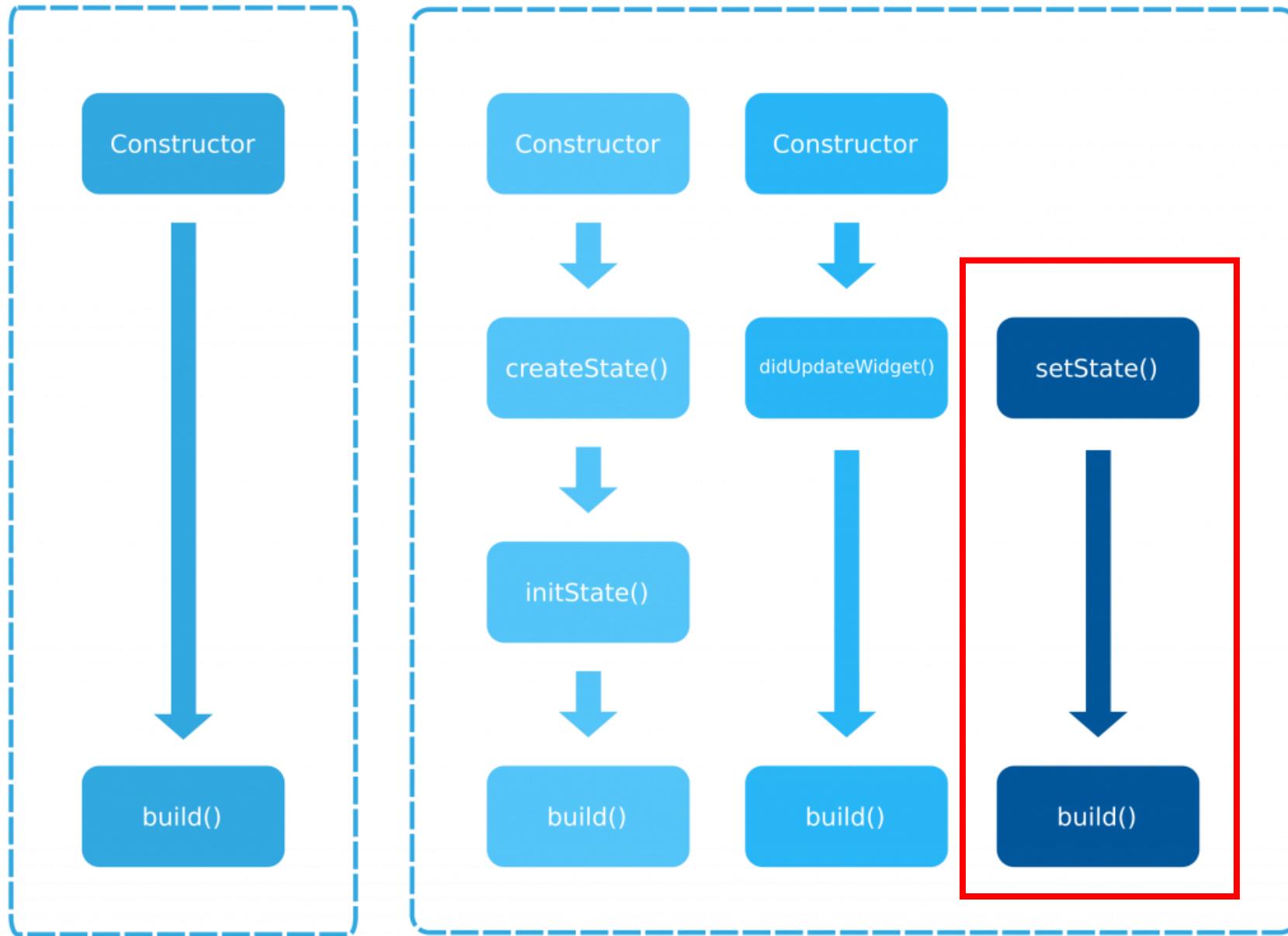
Ln 10, Col 2 Spaces: 2 UTF-8 CR Android Emulator - Pixel_2_API_29:5554 Pixel 2 API 29 (android-x86 emulator) Prettier

RangeError (index): Invalid value: Not in inclusive range 0..1: 2
See also:
<https://flutter.dev/docs/testing/errors>

A BRIEF LOOK AT WHAT FLUTTER ACTUALLY DOES

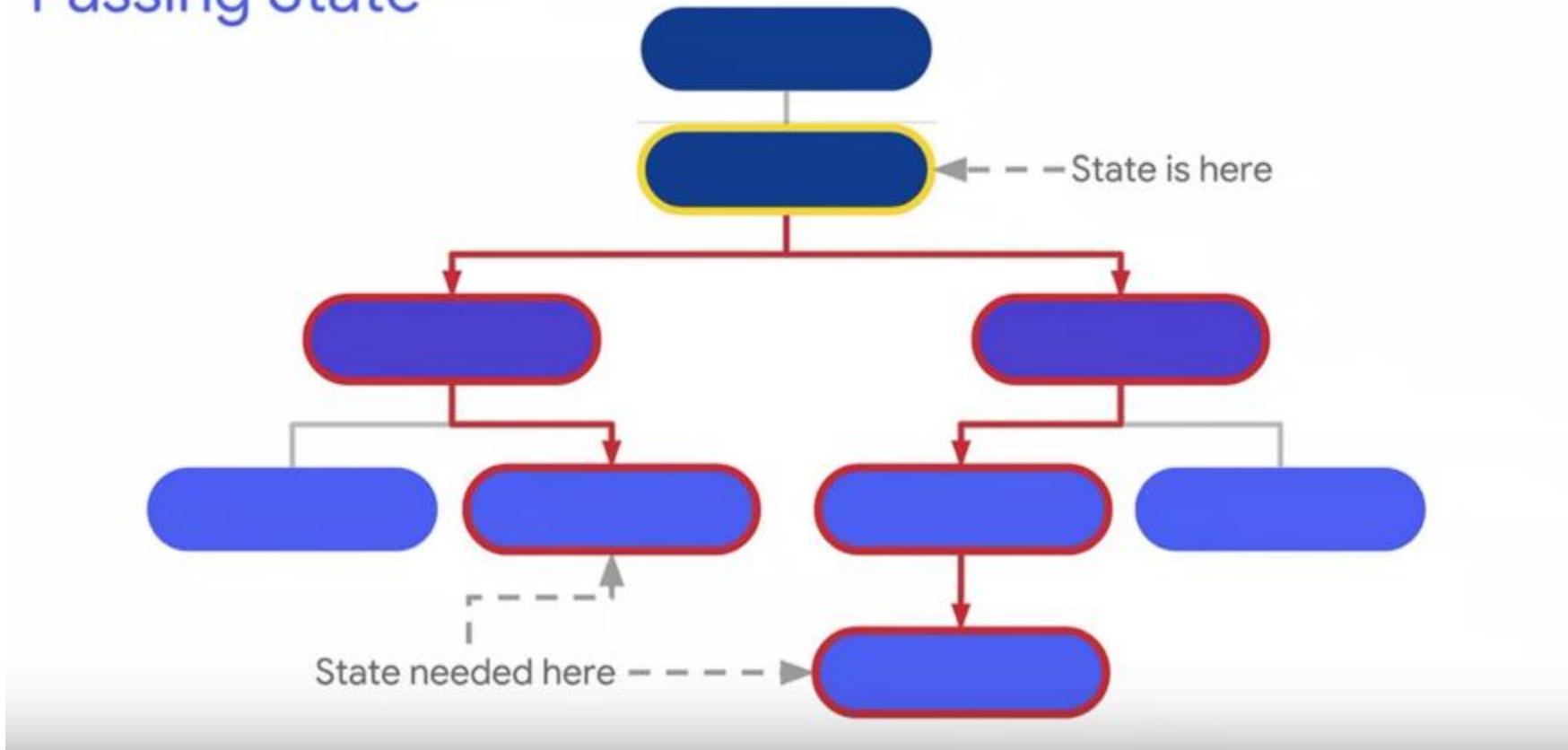
Stateless

Stateful



- Whenever we perform an operation on the widgets, suppose we have a RaisedButton and whenever we tap on it, it updates the Text, the very simple method is to use the setState().
- It is not a problem if you have a small widget tree but you need to avoid the setState() if we have a complex Widget tree.

Passing State



Suppose we have this Widget tree and we need to update only one development in the Widget tree but if we use `setState()`, it will re-render the whole widget tree and update all the elements.

USING PRIVATE PROPERTIES



EXPLORER

...

main.dart X

...

lib X main.dart lib

BELAJAR_FLUTTER_02

> .dart_tool

> .idea

> android

> build

> ios

> lib

main.dart

> test

> web

.gitignore

.metadata

.packages

analysis_options.yaml

belajar_flutter_02.iml

pubspec.lock

pubspec.yaml

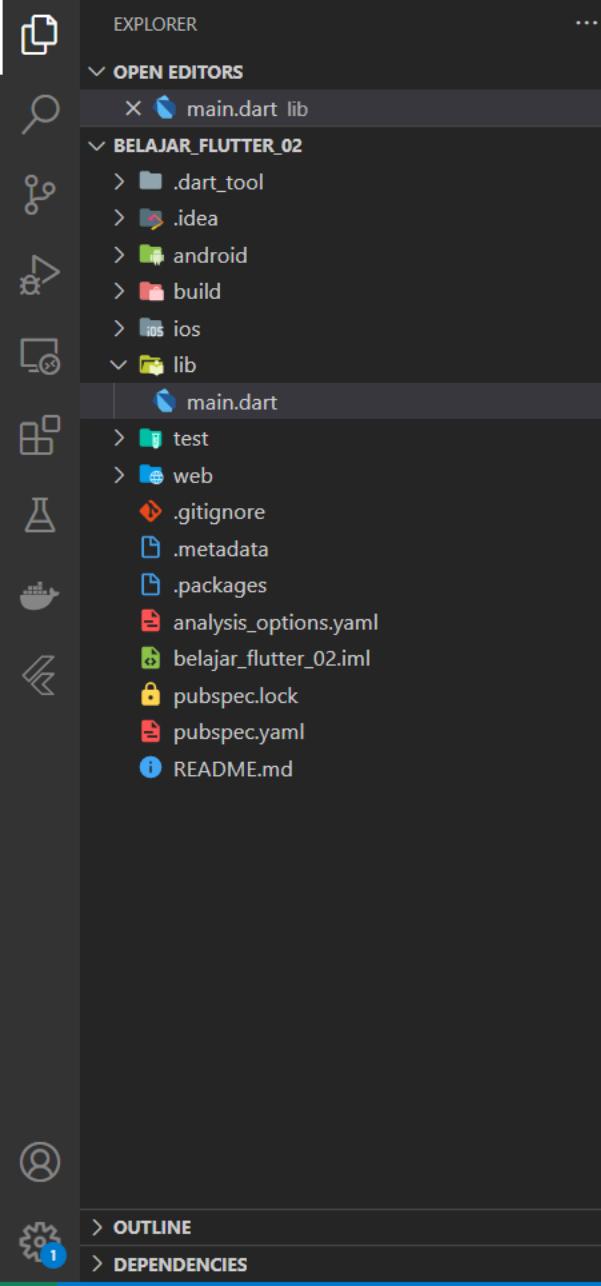
README.md



> OUTLINE

> DEPENDENCIES

```
4
5 class MyApp extends StatefulWidget {
6     @override
7     State<StatefulWidget> createState() {
8         return _MyAppState();
9     }
10 }
11
12 class _MyAppState extends State<MyApp> {
13     var _questionIndex = 0;
14
15     void _answerQuestion() {
16         setState(() {
17             _questionIndex = _questionIndex + 1;
18         });
19         print(_questionIndex);
20     }
21
22     var questions = [
23         'What\'s your favorite color?',
24         'What\'s your favorite animal?',
25     ];
```



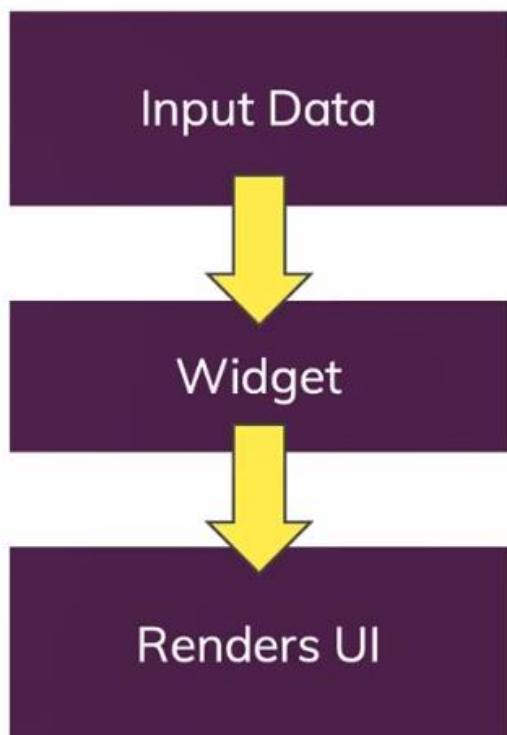
main.dart

lib > main.dart > MyApp > createState

```
26
27 @override
28 Widget build(BuildContext context) {
29   return MaterialApp(
30     home: Scaffold(
31       appBar: AppBar(
32         title: Text("My First App"),
33       ), // AppBar
34       body: Column(
35         children: [
36           Text(questions[_questionIndex]),
37           RaisedButton(
38             child: Text('Answer 1'),
39             onPressed: _answerQuestion,
40           ), // RaisedButton
41           RaisedButton(
42             child: Text('Answer 2'),
43             onPressed: () => print('Answer 2 chosen!'),
44           ), // RaisedButton
45           RaisedButton(
46             child: Text('Answer 3'),
47             onPressed: () {
```

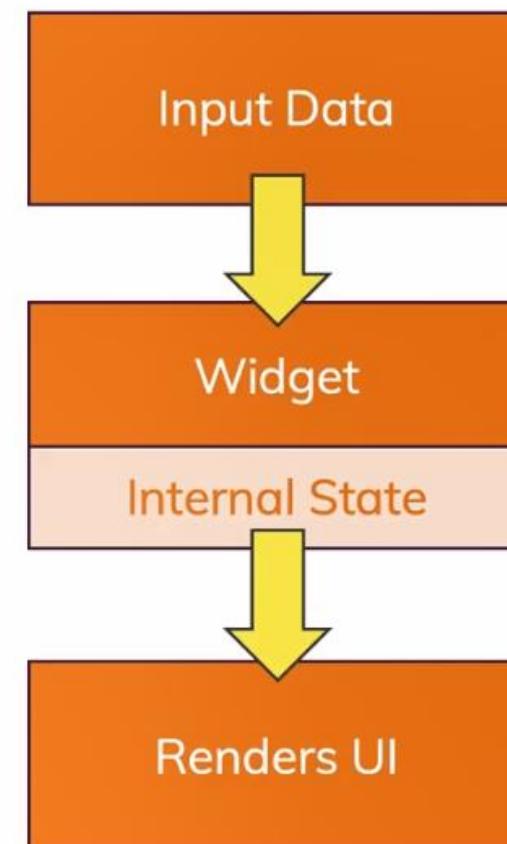
CREATING A NEW CUSTOM WIDGET

Stateless vs Stateful



Data can change (externally)

Gets (re)-rendered when **Input Data** changes



Data can change (externally)

Gets (re)-rendered when Input Data or local **State** changes

File Edit Selection View Go Run Terminal Help

question.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- question.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib

 - main.dart
 - question.dart

test

web

.gitignore

.metadata

.packages

analysis_options.yaml

belajar_flutter_02.iml

pubspec.lock

pubspec.yaml

README.md

question.dart

lib > question.dart > Question > build

```
1 import 'package:flutter/material.dart';
2
3 class Question extends StatelessWidget {
4     final String questionText;
5
6     Question(this.questionText);
7
8     @override
9     Widget build(BuildContext context) {
10        return Text(questionText);
11    }
12 }
13
```

Ln 10, Col 31 Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Chrome (web-javascript) Prettier

The screenshot shows the Visual Studio Code interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is "main.dart - belajar_flutter_02 - Visual Studio Code".

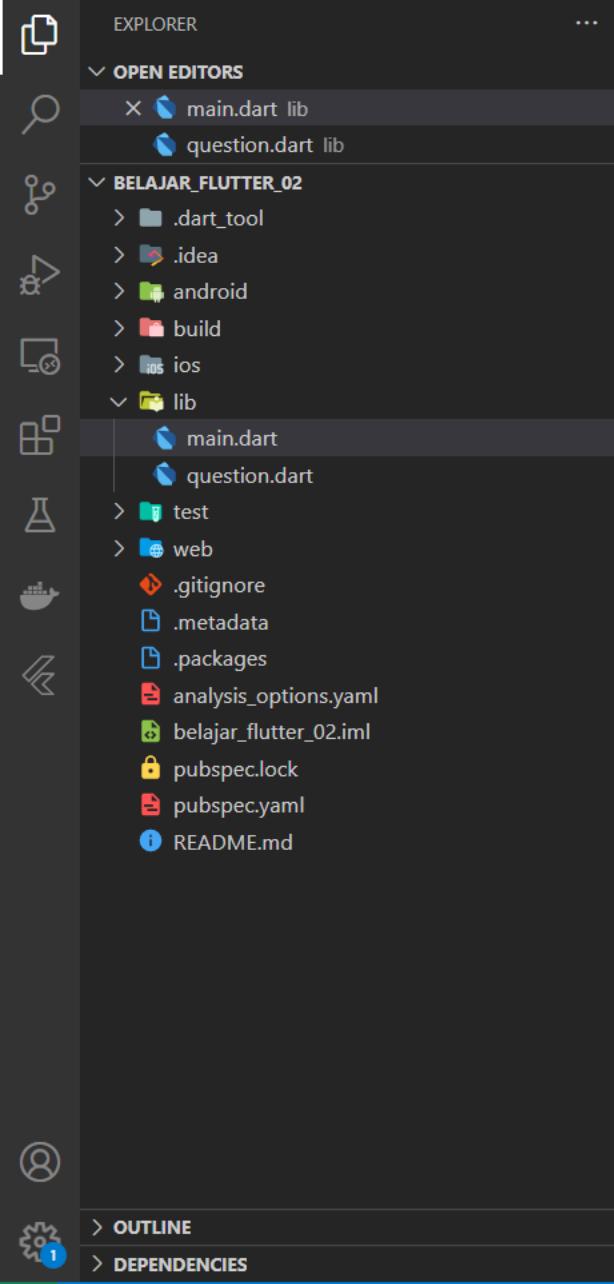
The Explorer sidebar on the left lists the project structure:

- OPEN EDITORS:
 - main.dart lib
 - question.dart lib
- BELAJAR_FLUTTER_02:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - main.dart
 - question.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md

The main editor area displays the code for "main.dart":

```
1 import 'package:flutter/material.dart';
2 import './question.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatefulWidget {
7   @override
8   State<StatefulWidget> createState() {
9     return _MyAppState();
10 }
11 }
12
13 class _MyAppState extends State<MyApp> {
14   var _questionIndex = 0;
15
16   void _answerQuestion() {
17     setState(() {
18       _questionIndex = _questionIndex + 1;
19     });
20     print(_questionIndex);
21 }
```

The status bar at the bottom shows: Ln 3, Col 1 (27 selected) Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier



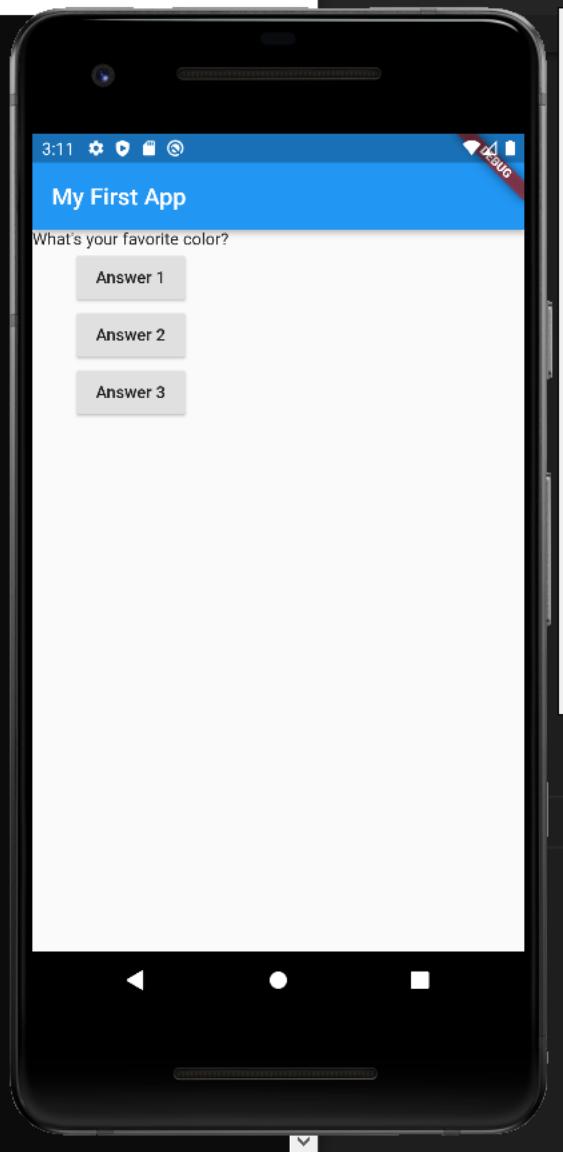
```
23 var questions = [
24   'What\'s your favorite color?',
25   'What\'s your favorite animal?',
26 ];
27
28 @override
29 Widget build(BuildContext context) {
30   return MaterialApp(
31     home: Scaffold(
32       appBar: AppBar(
33         title: Text("My First App"),
34       ), // AppBar
35       body: Column(
36         children: [
37           Question(
38             questions[_questionIndex],
39           ), // Question
40           RaisedButton(
41             child: Text('Answer 1'),
42             onPressed: _answerQuestion,
43           ), // RaisedButton
44           RaisedButton(
45             child: Text('Answer 2'),
46             onPressed: _answerQuestion,
47           ), // RaisedButton
48         ],
49       ),
50     ),
51   );
52 }
```



Command Prompt - flutter run

```
Performing hot restart...
Restarted application in 1,869ms.
I/flutter ( 5516): 1

Performing hot restart...
Restarted application in 2,068ms.
```





Command Prompt - flutter run

```
Performing hot restart...
Restarted application in 1,869ms.
I/flutter ( 5516): 1

Performing hot restart...
Restarted application in 2,068ms.
I/flutter ( 5516): 1
```



> OUTLINE

> DEPENDENCIES

44

45

RaisedButton(

FIRST STYLING AND LAYOUT STEPS

File Edit Selection View Go Run Terminal Help question.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- question.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib

 - main.dart
 - question.dart

test

web

- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

OUTLINE

DEPENDENCIES

main.dart question.dart

lib > question.dart > Question > build

```
1 import 'package:flutter/material.dart';
2
3 class Question extends StatelessWidget {
4   final String questionText;
5
6   Question(this.questionText);
7
8   @override
9   Widget build(BuildContext context) {
10    return Container(
11      width: double.infinity,
12      margin: EdgeInsets.all(10),
13      child: Text(
14        questionText,
15        style: TextStyle(fontSize: 28),
16        textAlign: TextAlign.center,
17      ), // Text
18    ); // Container
19  }
20}
21
```

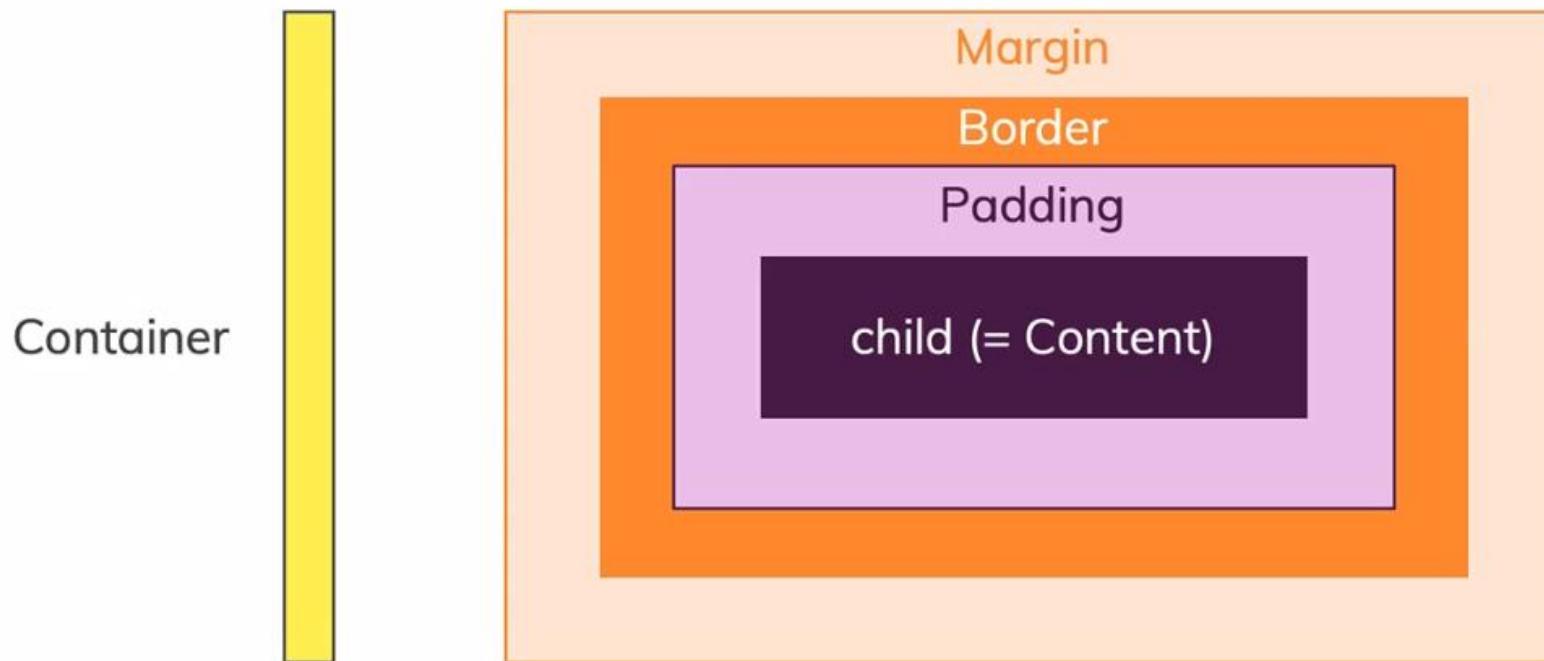
4:16 ⚡ DEBUG My First App What's your favorite animal? Answer 1 Answer 2 Answer 3

Ln 19, Col 1 Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

The Container()

child (= Content)

The Container()



ENUMS AND MULTIPLE CONSTRUCTORS

A screenshot of the Visual Studio Code interface, showing the Flutter source code for the `TextAlign` enum. The code is displayed in a dark-themed editor with syntax highlighting. The file path is `Q:\myprograms\flutter\bin\cache\pkg\sky_engine\lib\ui\text.dart`. The code defines the `TextAlign` enum with five values: `left`, `right`, `center`, `justify`, and `start`. It also includes comments explaining the behavior of each value.

```
1482 // The order of this enum must match the order of the values in RenderStyleConst
1483 enum TextAlign {
1484     /// Align the text on the left edge of the container.
1485     left,
1486
1487     /// Align the text on the right edge of the container.
1488     right,
1489
1490     /// Align the text in the center of the container.
1491     center,
1492
1493     /// Stretch lines of text that end with a soft line break to fill the width of
1494     /// the container.
1495     ///
1496     /// Lines that end with hard line breaks are aligned towards the [start] edge.
1497     justify,
1498
1499     /// Align the text on the leading edge of the container.
1500     ///
1501     /// For left-to-right text ([TextDirection.ltr]), this is the left edge.
1502     ///
1503     /// For right-to-left text ([TextDirection rtl]), this is the right edge.
```



DartPad

New Pad

Reset

Format

Install SDK

magenta-tulip-2609

Samples

⋮

```
6 class Person {  
7   String name = "Anonymous";  
8   int age = -1;  
9   Person({this.name, this.age = 30});  
10  Person.veryOld(this.name) {  
11    age = 60;  
12  }  
13 }  
14  
15 void main() {  
16  var p1 = Person(age:30, name:'Rizqi');  
17  var p2 = Person(name: 'Robert');  
18  var p3 = Person.veryOld('Toni');  
19  
20  print(p3.name);  
21  print(p3.age);  
22  
23 //  print('First Hello!');  
24 //  print("Second Hello!");  
25  
26 //  double firstResult = addNumbers(1,2);  
27 //  firstResult = addNumbers(1,2,0);
```

▶ Run

Console

Toni
60

Documentation

Person p2

local variable

info

line 16 • The value of the local variable 'p1' isn't used. [\(view docs\)](#)

Try removing the variable or using it.

info

line 17 • The value of the local variable 'p2' isn't used. [\(view docs\)](#)

Try removing the variable or using it.



OFFICIAL DOCS AND THE WIDGET CATALOGUE

A screenshot of a web browser showing the Flutter website at <https://flutter.dev>. The page features a large blue header with the Flutter logo and navigation links for Multi-Platform, Development, Ecosystem, Showcase, Docs, and Get started. Below the header, a large white button with the text "Build apps for any screen" is displayed. To the right, a white callout box contains links to "What's New", "Install Flutter", "Editor Support", "Dev Tools", "Hot Reload", "Cookbook", "Profiling", and "Tutorials". At the bottom, four mobile device screenshots demonstrate Flutter's跨平台能力: a travel app interface, a car status check app showing "ALL GOOD", a color picker, and a weather app showing "65° Clear".

DartPad | Linter for Dart | Flutter - Build apps for any screen | +

Flutter

Multi-Platform Development Ecosystem Showcase Docs Get started

Build apps for any screen

What's New Install Flutter

Editor Support Dev Tools

Hot Reload Cookbook

Profiling Tutorials

Flutter folks

Vacation Inspiration

Check Status

iX xDrive50

ALL GOOD

Updated from vehicle on 9/20/2021 01:59 PM

8:00 AM · Tue, Oct 9

Comm
Ross · Up
20 min

65° Clear

TUE WED THU

<https://docs.flutter.dev>

DartPad | Linter for Dart | Widget catalog | Flutter

https://docs.flutter.dev/development/ui/widgets

Flutter

Multi-Platform ▾ Development ▾ Ecosystem ▾ Showcase Docs ▾ Get started

Get started

Samples & tutorials

Development

User interface

- Introduction to widgets
- Building layouts
- Adding interactivity
- Assets and images
- Navigation & routing
- Animations
- Advanced UI

Widget catalog

Data & backend

Accessibility & internationalization

Platform integration

Packages & plugins

Add Flutter to existing app

Waiting for cache...

Widget catalog

Development > UI > Widgets

Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

Accessibility
Make your app accessible.
[Visit](#)

Animation and Motion
Bring animations to your app.
[Visit](#)

Assets, Images, and Icons
Manage assets, display images, and show icons.
[Visit](#)

Async
Async patterns to your Flutter application.
[Visit](#)

Basics
Widgets you absolutely need to know before building your first Flutter app.
[Visit](#)

Cupertino (iOS-style widgets)
Beautiful and high-fidelity widgets for current iOS design language.
[Visit](#)



Multi-Platform

Development

Ecosystem

Showcase

Docs



Get started

Get started

Samples & tutorials

Development

User interface

Introduction to widgets

Building layouts

Adding interactivity

Assets and images

Navigation & routing

Animations

Advanced UI

[Widget catalog](#)

Data & backend

Accessibility & internationalization

Platform integration

Packages & plugins

Add Flutter to existing app

Tools & features

Layout widgets

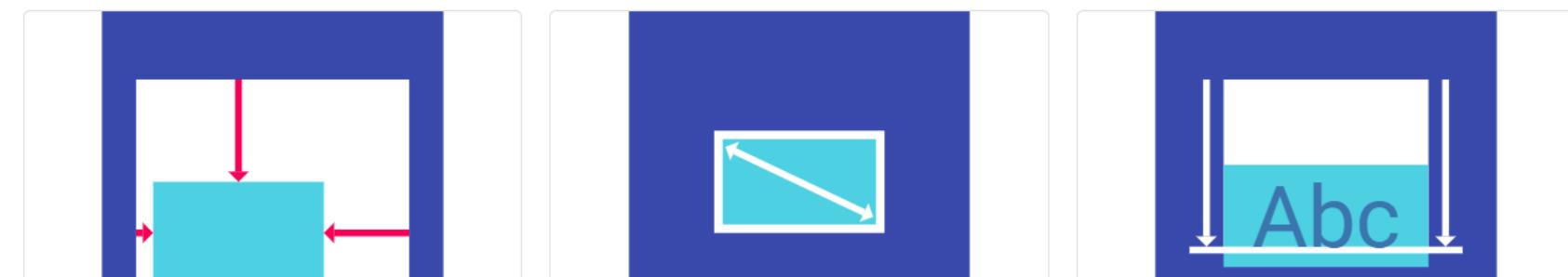
[Development](#) > [UI](#) > [Widgets](#) > Layout

Arrange other widgets columns, rows, grids, and many other layouts.

- [Single-child layout widgets](#)
- [Multi-child layout widgets](#)
- [Sliver widgets](#)

See more widgets in the [widget catalog](#).

Single-child layout widgets



CLASSES

[AbsorbPointer](#)[Accumulator](#)[Action](#)[ActionDispatcher](#)[ActionListener](#)[Actions](#)[ActivateAction](#)[ActivateIntent](#)[Align](#)[Alignment](#)[AlignmentDirectional](#)[AlignmentGeometry](#)[AlignmentGeometryTween](#)[AlignmentTween](#)[AlignTransition](#)[AlwaysScrollableScrollView](#)[AlwaysStoppedAnimation](#)[AndroidView](#)[AndroidViewSurface](#)[Animatable](#)[AnimatedAlign](#)

CONSTRUCTORS

[Container](#)

PROPERTIES

[alignment](#)[child](#)[clipBehavior](#)[color](#)[constraints](#)[decoration](#)[foregroundDecorati...](#)[hashCode](#)[key](#)[margin](#)[padding](#)[runtimeType](#)[transform](#)[transformAlignment](#)

METHODS

[build](#)[createElement](#)

PASSING THE CALLBACK FUNCTIONS AROUND

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** answer.dart - belajar_flutter_02 - Visual Studio Code.
- Explorer View (Left):**
 - OPEN EDITORS: main.dart, question.dart, answer.dart (selected).
 - BELAJAR_FLUTTER_02 folder:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - answer.dart (selected)
 - main.dart
 - question.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Code Editor (Center):** The file lib/answer.dart is open, showing the following Dart code:

```
1 import 'package:flutter/material.dart';
2
3 class Answer extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Container(
7       width: double.infinity,
8       child: RaisedButton(
9         color: Colors.blue,
10        child: Text('Answer 1'),
11        onPressed: null,
12      ), // RaisedButton
13    ); // Container
14  }
15 }
16
```
- Bottom Status Bar:** Ln 4, Col 1, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.

The screenshot shows the Visual Studio Code interface with a Flutter project named "belajar_flutter_02".

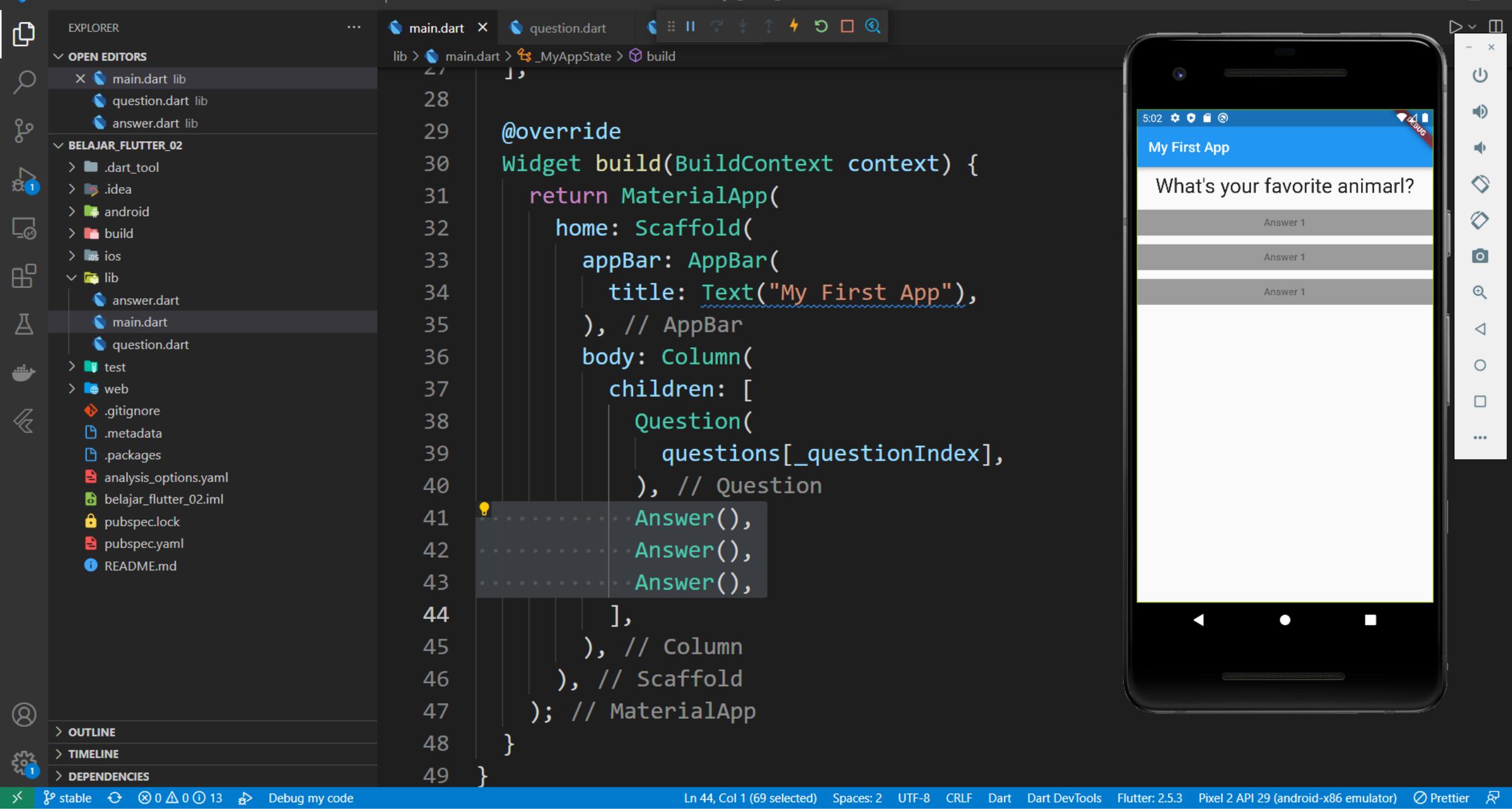
File Explorer: Shows the project structure with files like main.dart, question.dart, answer.dart, and various build folders for Android and iOS.

Open Editors: Displays two files: main.dart and question.dart.

Code Editor: The main.dart file is open, showing Dart code for a Flutter application. The code imports flutter/material.dart, question.dart, and answer.dart, then defines a MyApp class that extends StatelessWidget. It overrides the createState method to return _MyAppState(). The _MyAppState class extends State<MyApp> and initializes _questionIndex to 0. It has a _answerQuestion method that increments _questionIndex by 1 and prints its value.

```
main.dart - belajar_flutter_02 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
main.dart question.dart
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 import './question.dart';
3 import './answer.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatefulWidget {
8   @override
9   State<StatefulWidget> createState() {
10    return _MyAppState();
11  }
12 }
13
14 class _MyAppState extends State<MyApp> {
15   var _questionIndex = 0;
16
17   void _answerQuestion() {
18     setState(() {
19       _questionIndex = _questionIndex + 1;
20     });
21     print(_questionIndex);
22 }
```

Bottom Status Bar: Shows the current file is stable, has 0 errors, 0 warnings, and 13 hints. It also indicates the code is 25 selected, and the status bar includes tabs for Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier, and a few other icons.



File Edit Selection View Go Run Terminal Help answer.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart question.dart lib > answer.dart > Answer > build

OPEN EDITORS main.dart lib question.dart lib X answer.dart lib

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib answer.dart main.dart question.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

answer.dart lib answer.dart

```
1 import 'package:flutter/material.dart';
2
3 class Answer extends StatelessWidget {
4     final Function selectHandler;
5
6     Answer(this.selectHandler);
7
8     @override
9     Widget build(BuildContext context) {
10         return Container(
11             width: double.infinity,
12             child: RaisedButton(
13                 color: Colors.blue,
14                 child: Text('Answer 1'),
15                 onPressed: selectHandler,
16             ), // RaisedButton
17         ); // Container
18     }
19 }
```

Ln 16, Col 7 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER OPEN EDITORS main.dart X question.dart lib > main.dart > _MyAppState > build

29 @override
30 Widget build(BuildContext context) {
31 return MaterialApp(
32 home: Scaffold(
33 appBar: AppBar(
34 title: Text("My First App"),
35), // AppBar
36 body: Column(
37 children: [
38 Question(
39 questions[_questionIndex],
40), // Question
41 Answer(_answerQuestion),
42 Answer(_answerQuestion),
43 Answer(_answerQuestion),
44],
45), // Column
46), // Scaffold
47); // MaterialApp
48 }
49 }
50 }

5:08 My First App What's your favorite animal?
Answer 1
Answer 1
Answer 1

st able 0 △ 0 12 Debug my code Ln 43, Col 35 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

INTRODUCING MAPS

```
var mapName = {  
    key 1 : value 1,  
    key 2 : value 2,  
    ...  
    key n : value n  
}
```

A screenshot of the Visual Studio Code interface, showing the code editor with Dart code for a Flutter application named `belajar_flutter_02`.

The code in the editor is:

```
void _answerQuestion() {
    setState(() {
        _questionIndex = _questionIndex + 1;
    });
    print(_questionIndex);
}

var questions = [
{
    'questionText': 'What\'s your favorite color?',
    'answers': ['Black', 'Red', 'Green', 'White'],
},
{
    'questionText': 'What\'s your favorite animal?',
    'answers': ['Rabbit', 'Snake', 'Elephant', 'Lion'],
},
{
    'questionText': 'Where\'s your favorite city?',
    'answers': ['Banyuwangi', 'Denpasar', 'Surabaya', 'Bandung'],
},
];

```

The code editor features a dark theme. The sidebar on the left shows the project structure with files like `main.dart`, `question.dart`, and `answer.dart`. The status bar at the bottom provides information about the file (`main.dart`), line count (Ln 48), column count (Col 22), and encoding (UTF-8). The bottom right corner includes icons for Prettier and a GitHub icon.

MAPPING LISTS TO WIDGETS

The screenshot shows the Visual Studio Code interface with a Flutter project named "belajar_flutter_02".

File Explorer: Shows the project structure under "lib":

- main.dart
- question.dart
- answer.dart (selected)

Code Editor: Displays the content of "answer.dart":

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

class Answer extends StatelessWidget {
    final Function selectHandler;
    final String answerText;

    Answer(this.selectHandler, this.answerText);

    @override
    Widget build(BuildContext context) {
        return Container(
            width: double.infinity,
            child: RaisedButton(
                color: Colors.blue,
                textColor: Colors.white,
                child: Text(answerText),
                onPressed: selectHandler,
            ), // RaisedButton
        ); // Container
    }
}
```

Bottom Status Bar:

- Ln 20, Col 4 (3 selected)
- Spaces: 2
- UTF-8
- CRLF
- Dart
- Dart DevTools
- Flutter: 2.5.3
- Pixel 2 API 29 (android-x86 emulator)
- Prettier

File Edit Selection View Go Run Terminal Help

main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
- answer.dart
- main.dart
- question.dart

test

web

.gitignore

.metadata

.packages

analysis_options.yaml

belajar_flutter_02.iml

pubspec.lock

pubspec.yaml

README.md

OUTLINE

TIMELINE

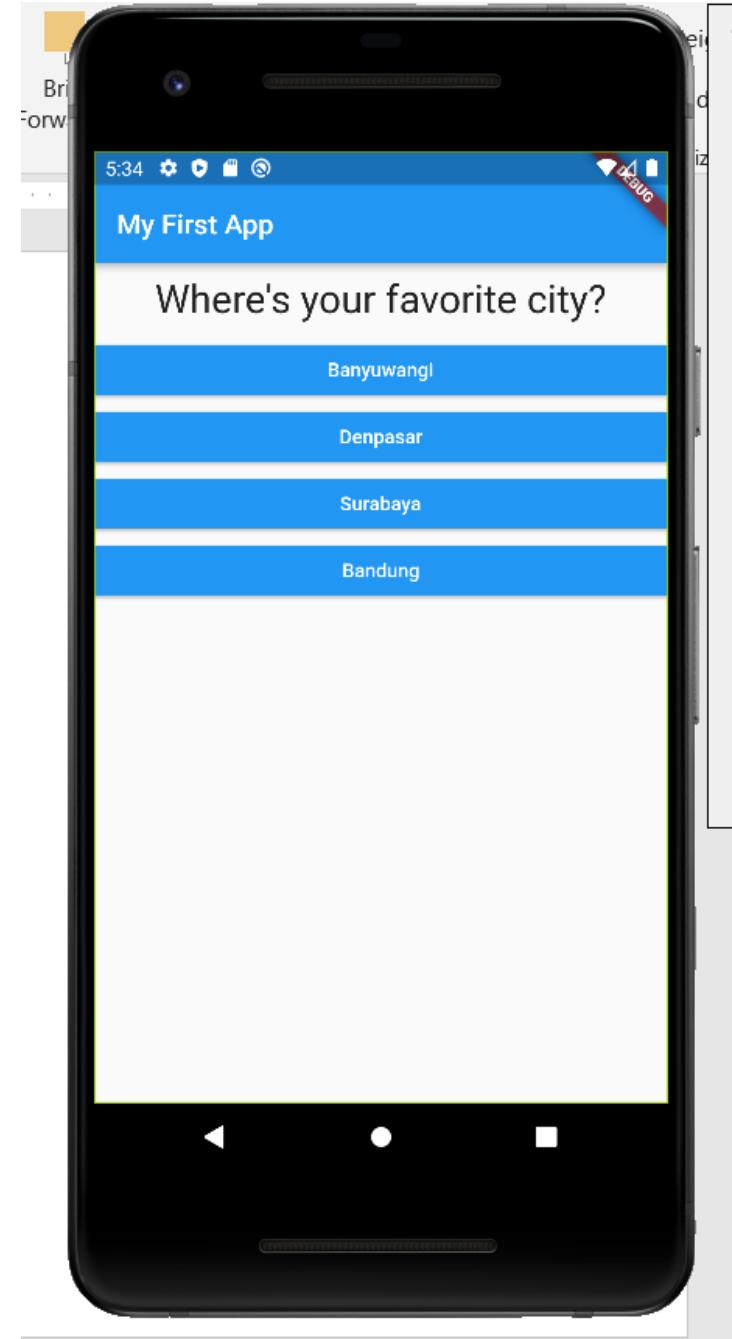
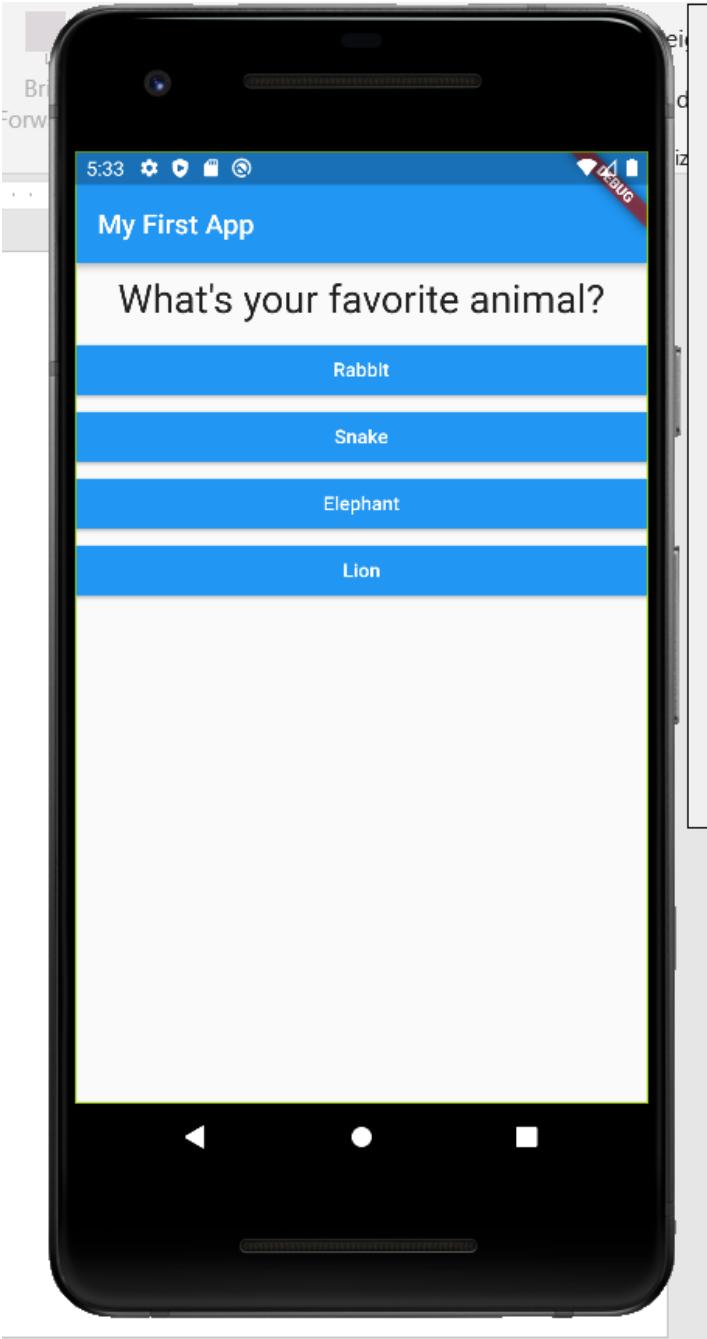
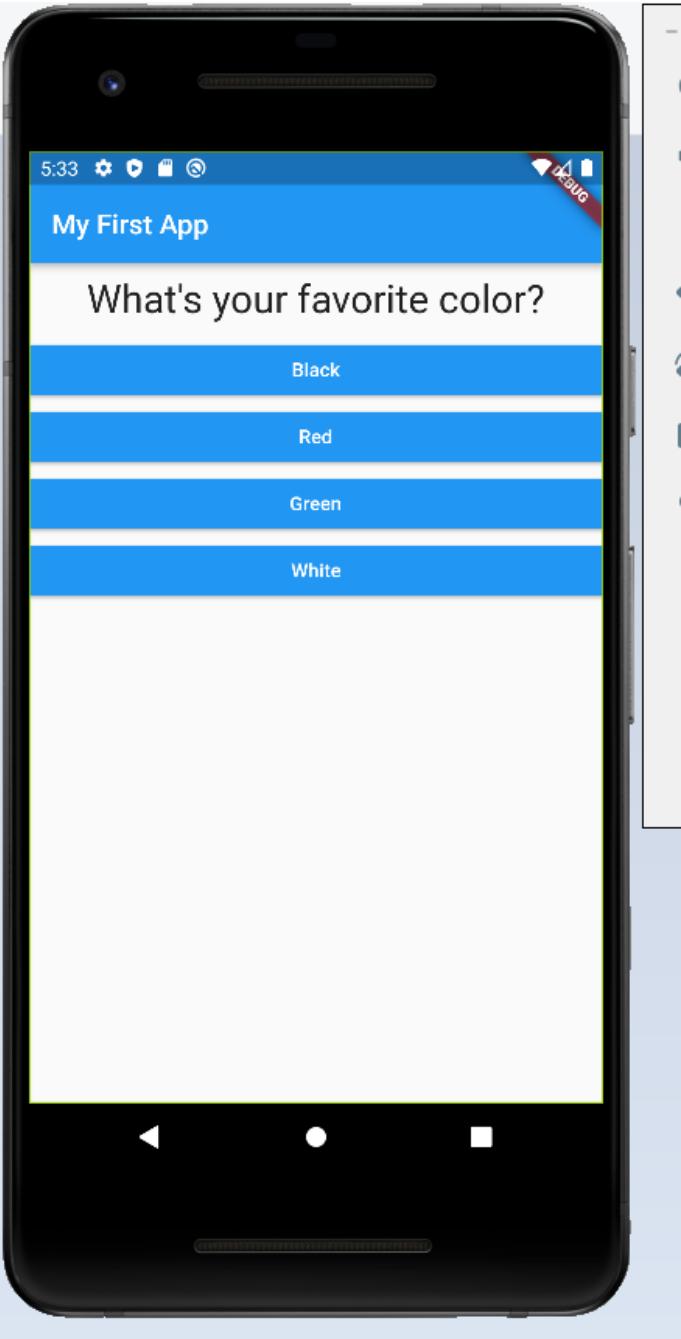
DEPENDENCIES

main.dart X question.dart

lib > main.dart > _MyAppState > build

```
39 @override
40 Widget build(BuildContext context) {
41   return MaterialApp(
42     home: Scaffold(
43       appBar: AppBar(
44         title: Text("My First App"),
45       ), // AppBar
46       body: Column(
47         children: [
48           Question(
49             questions[_questionIndex]['questionText'],
50           ), // Question
51           ...(questions[_questionIndex]['answers'] as List<String>)
52             .map((answer) {
53               return Answer(_answerQuestion, answer);
54             }).toList()
55           ],
56         ), // Column
57       ), // Scaffold
58     ); // MaterialApp
59   }
60 }
```

Ln 48, Col 22 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier



FINAL VERSUS CONST

- There are many ways we can create variables in Flutter. Two of them are const and final! If we check their functionality, they both have similar work, to create variables whose values won't be changed.

```
void main() {  
    const String data = "Flutter";  
    data = "dart";  
}
```

```
void main() {  
    final String data = "Flutter";  
    data = "dart";  
}
```

- So, both of them threw almost the same errors that final and const variables can only be assigned once! This is a bit confusing right. Then what is the exact difference?

- The main difference between `const` and `final` is that **`const`** can be considered as a compile-time constant while **`final`** can be considered as a run-time constant.
- So when you want the constant value and you are aware of the value to be assigned, at the compile-time itself, you can use **`const!`**
- But let's say you want a constant value but you don't know its value at compile-time, then you can use **`final!`**

- Example for using **const**:

```
void main() {  
    const String data = "Flutter";  
    print(data);  
}
```

- Example for using **final**:

```
void main() {  
    final DateTime date = DateTime.now();  
    print(date);  
}
```

- Now, what if we try to use const instead of final in the above example?

```
void main() {  
    const| DateTime date = DateTime.now();  
    print(date);  
}
```

INTRODUCING IF STATEMENTS & OUTPUTTING WIDGETS CONDITIONALLY

File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
- answer.dart
- main.dart
- question.dart

test

web

.gitignore

.metadata

.packages

analysis_options.yaml

belajar_flutter_02.iml

pubspec.lock

pubspec.yaml

README.md

OUTLINE

TIMELINE

DEPENDENCIES

main.dart X question.dart

lib > main.dart > _MyAppState > build

39 @override

40 Widget build(BuildContext context) {

41 return MaterialApp(

42 home: Scaffold(

43 appBar: AppBar(

44 title: Text("My First App"),

45), // AppBar

46 body: Column(

47 children: [

48 Question(

49 questions[_questionIndex]['quest'], // Question

50), // Question

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, log)

#8 RendererBinding._handlePersistentFrameCallback

#9 SchedulerBinding._invokeFrameCallback

#10 SchedulerBinding.handleDrawFrame

#11 SchedulerBinding._handleDrawFrame

#15 _invoke (dart:ui/hooks.dart:166:10)

#16 PlatformDispatcher._drawFrame (dart:ui/platform_dispatcher:129:31)

#17 _drawFrame (dart:ui/hooks.dart:129:31)

(elided 3 frames from dart:async)

8:51 8:51 RangeError (index): Invalid value: Not in inclusive range 0..2; 3 See also: https://flutter.dev/docs/testing/errors

ing.dart:319

ng.dart:1143

ng.dart:1080

ing.dart:996

Ln 48, Col 22 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

A screenshot of the Visual Studio Code interface, showing the code editor with Dart code for a Flutter application named "belajar_flutter_02".

The code defines a class `_MyAppState` that extends `State<MyApp>`. It contains a list of questions with their text and answers. The `_answerQuestion` method increments the question index.

```
main.dart - belajar_flutter_02 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
main.dart lib
question.dart lib
answer.dart lib
BELAJAR_FLUTTER_02
.dart_tool
.idea
android
build
ios
lib
answer.dart
main.dart
question.dart
test
web
.gitignore
.metadata
.packages
analysis_options.yaml
belajar_flutter_02.iml
pubspec.lock
pubspec.yaml
README.md
main.dart
question.dart
lib > main.dart > _MyAppState > _answerQuestion
Ln 34, Col 1 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier
14 class _MyAppState extends State<MyApp> {
15     var _questionIndex = 0;
16     final questions = const [
17         {
18             'questionText': 'What\'s your favorite color?',
19             'answers': ['Black', 'Red', 'Green', 'White'],
20         },
21         {
22             'questionText': 'What\'s your favorite animal?',
23             'answers': ['Rabbit', 'Snake', 'Elephant', 'Lion'],
24         },
25         {
26             'questionText': 'Where\'s your favorite city?',
27             'answers': ['Banyuwangi', 'Denpasar', 'Surabaya', 'Bandung'],
28         },
29     ];
30
31     void _answerQuestion() {
32         setState(() {
33             _questionIndex = _questionIndex + 1;
34         });
35         print(_questionIndex);
```

A screenshot of the Visual Studio Code interface, showing the code editor with Dart code for a Flutter application named "belajar_flutter_02".

The code in the main.dart file handles the logic for answering questions:

```
void _answerQuestion() {
    setState(() {
        _questionIndex = _questionIndex + 1;
    });
    print(_questionIndex);
    if (_questionIndex < questions.length) {
        print('We have more questions!');
    } else {
        print('No more questions!');
    }
}

@Override
Widget build(BuildContext context) {
    return MaterialApp(
        home: Scaffold(
            appBar: AppBar(
                title: Text("My First App"),
            ), // AppBar
            body: _questionIndex < questions.length
                ? Column(
                    children: [

```

The code editor shows syntax highlighting for Dart, including color-coded keywords like `void`, `if`, and `return`. It also highlights variable names and function names.

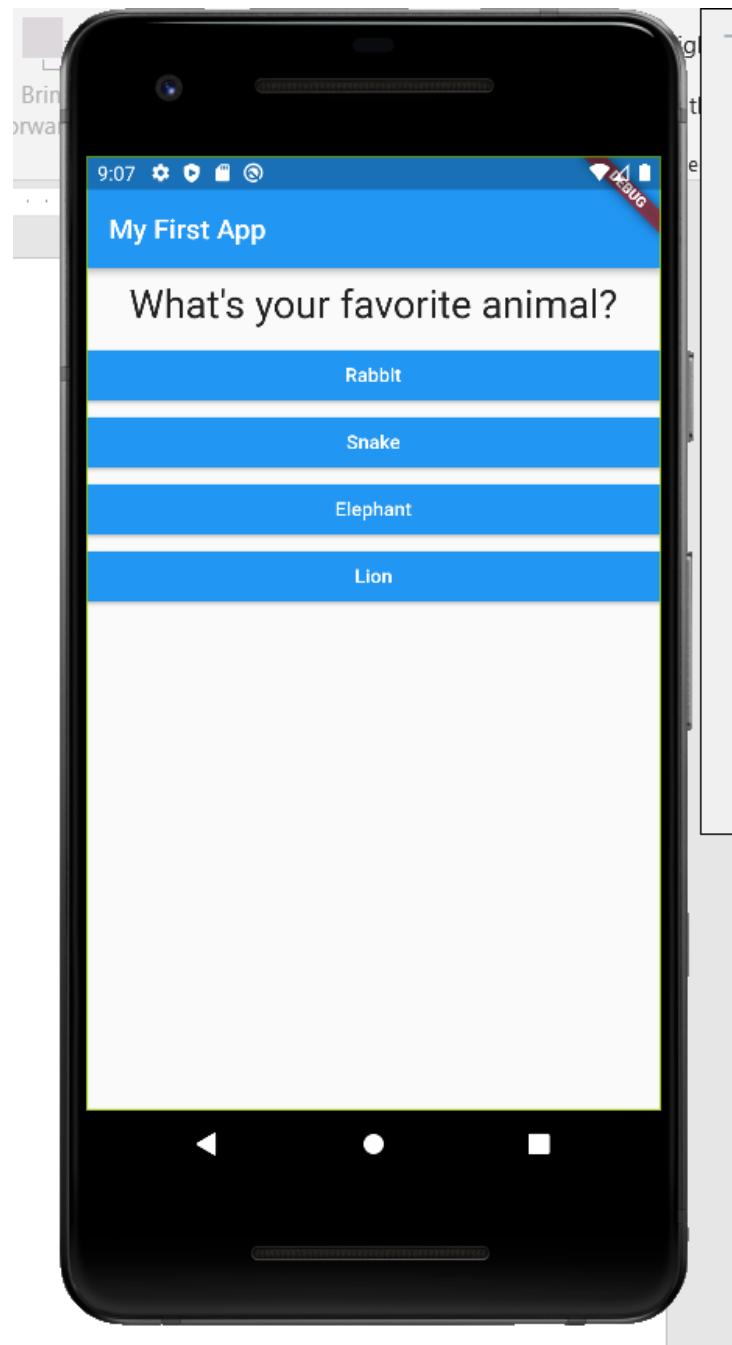
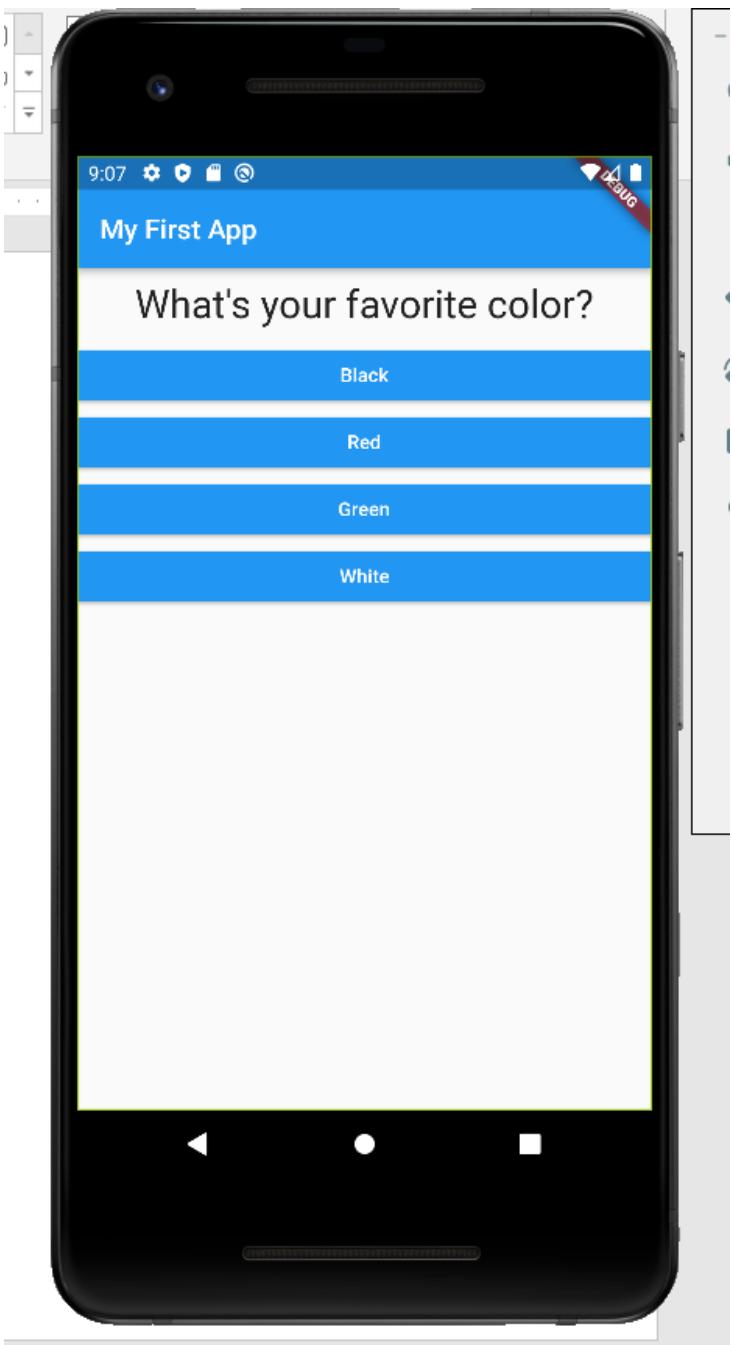
The Explorer sidebar on the left lists the project structure, including files like `main.dart lib`, `question.dart lib`, and `answer.dart lib`, along with build artifacts for iOS and Android.

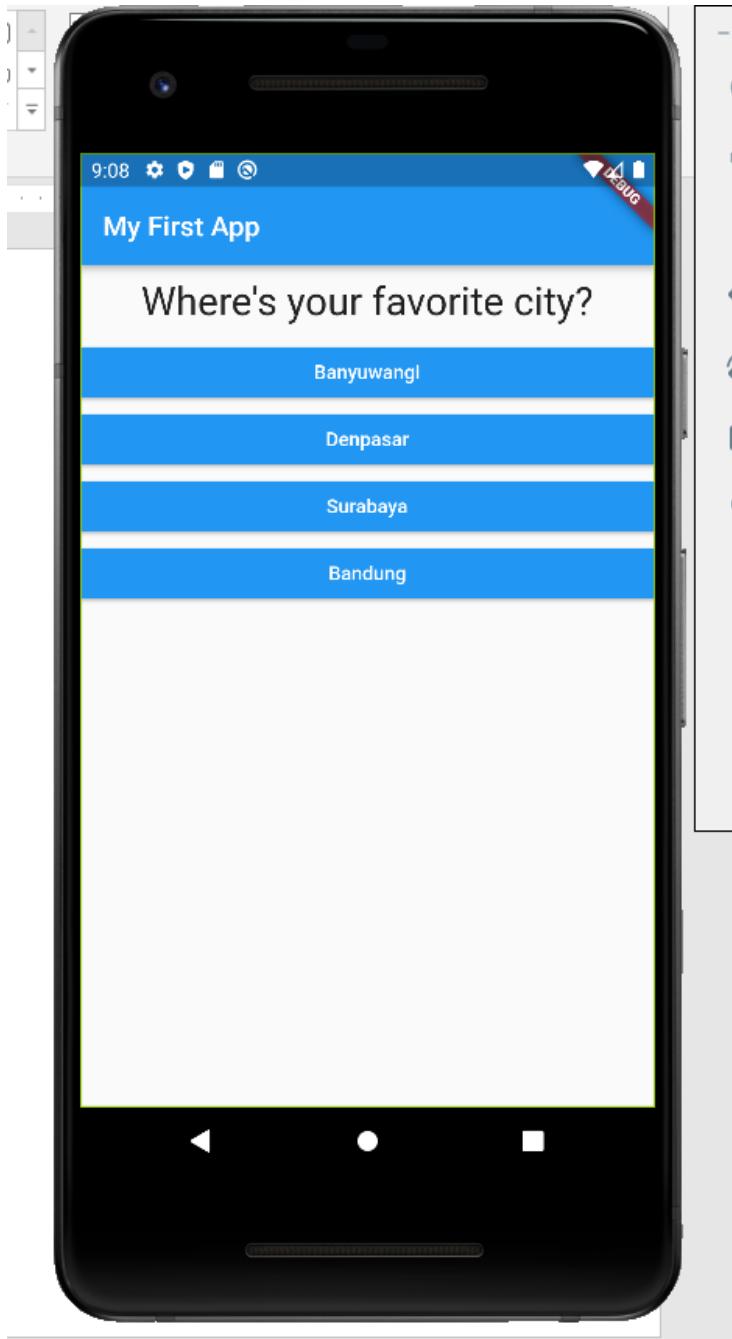
The status bar at the bottom provides information about the current file, including the line number (Ln 34, Col 1), character count (Spaces: 2), encoding (UTF-8), and end-of-line style (CRLF). It also shows the Dart DevTools icon and the Flutter version (2.5.3).

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code.
- Left Sidebar (Explorer):**
 - OPEN EDITORS: main.dart lib, question.dart lib, answer.dart lib.
 - BELAJAR_FLUTTER_02: .dart_tool, .idea, android, build, ios, lib (selected), answer.dart, main.dart, question.dart, test, web, .gitignore, .metadata, .packages, analysis_options.yaml, belajar_flutter_02.iml, pubspec.lock, pubspec.yaml, README.md.
- Central Area:** Code editor showing Dart code for a Flutter application.

```
47     appBar: AppBar(
48       title: Text("My First App"),
49     ), // AppBar
50   body: _questionIndex < questions.length
51     ? Column(
52       children: [
53         Question(
54           questions[_questionIndex]['questionText'],
55         ), // Question
56           ...(questions[_questionIndex]['answers'] as List<String>)
57             .map((answer) {
58               return Answer(_answerQuestion, answer);
59             }).toList()
60       ],
61     ) // Column
62     : Center(
63       child: Text('You did it!'),
64     ), // Center
65   ), // Scaffold
66 ); // MaterialApp
67 }
68 }
```
- Bottom Status Bar:** Ln 34, Col 1, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.





THE NULL VALUE AND NULL SAFETY



DartPad

New Pad

Reset

Format

Install SDK

magenta-tulip-2609

Samples



```
1 void main() {  
2   var userName = 'Rizqi';  
3  
4   userName = null;  
5   userName = 'Ardiansyah';  
6  
7   if (userName != null) {  
8     // only continue if it's not null  
9   }  
10 }  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

▶ Run

Console

Documentation



DartPad Linter for Dart Layout widgets | Flutter https://www.dartpad.dev/?null_safety=true

DartPad

New Pad Reset Format Install SDK magenta-tulip-2609 Samples

```
1 void main() {  
2     var userName = 'Rizqi';  
3  
4     userName = null;  
5     userName = 'Ardiansyah';  
6  
7     if (userName != null) {  
8         // only continue if it's not null  
9     }  
10 }  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

Run

Console

Documentation

error line 4 • A value of type 'Null' can't be assigned to a variable of type 'String'. (view docs)
Try changing the type of the variable, or casting the right-hand type to 'String'.

info line 7 • The operand can't be null, so the condition is always true. (view docs)
Remove the condition.

Privacy notice Send feedback Null Safety

02_31_[DART DEEP DIVE] The null Value and Null Safety Chapter 2.5.3 Dart SDK 2.14.4 Learn Flutter.mp4

DartPad Linter for Dart Layout widgets | Flutter https://www.dartpad.dev/?null_safety=true

DartPad

New Pad Reset Format Install SDK magenta-tulip-2609 Samples

```
1 void main() {  
2   String userName = 'Rizqi';  
3  
4   userName = null;  
5   userName = 'Ardiansyah';  
6  
7   if (userName != null) {  
8     // only continue if it's not null  
9   }  
10 }  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

Run

Console

Documentation

abstract class String implements

Co error line 4 • A value of type 'Null' can't be assigned to a variable of type 'String'. (view docs)

As Str rep cor Ne info line 7 • The operand can't be null, so the condition is always true. (view docs)

rep ch ch Remove the condition.

Privacy notice Send feedback Null Safety 2 issues hide Based on Flutter 2.5.3 Dart SDK 2.14.4

A screenshot of the DartPad web interface. The top navigation bar shows tabs for 'DartPad', 'Linter for Dart', and 'Layout widgets | Flutter'. The URL in the address bar is https://www.dartpad.dev/?null_safety=true. The main area contains Dart code:

```
1 void main() {  
2   String? userName = 'Rizqi';  
3  
4   userName = null;  
5   userName = 'Ardiansyah';  
6  
7   if (userName != null) {  
8     // only continue if it's not null  
9   }  
10 }  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

A blue 'Run' button is positioned above the code editor. To the right, there are two panels: 'Console' and 'Documentation'. A tooltip in the bottom right corner provides information about a warning:

info line 7 • The operand can't be null, so the condition is always true. (view docs)
Remove the condition.

The bottom of the screen includes links for 'Privacy notice', 'Send feedback', 'Null Safety' (with a toggle switch), '1 issue', 'hide', 'Based on Flutter 2.5.3 Dart SDK 2.14.4', and a help icon.

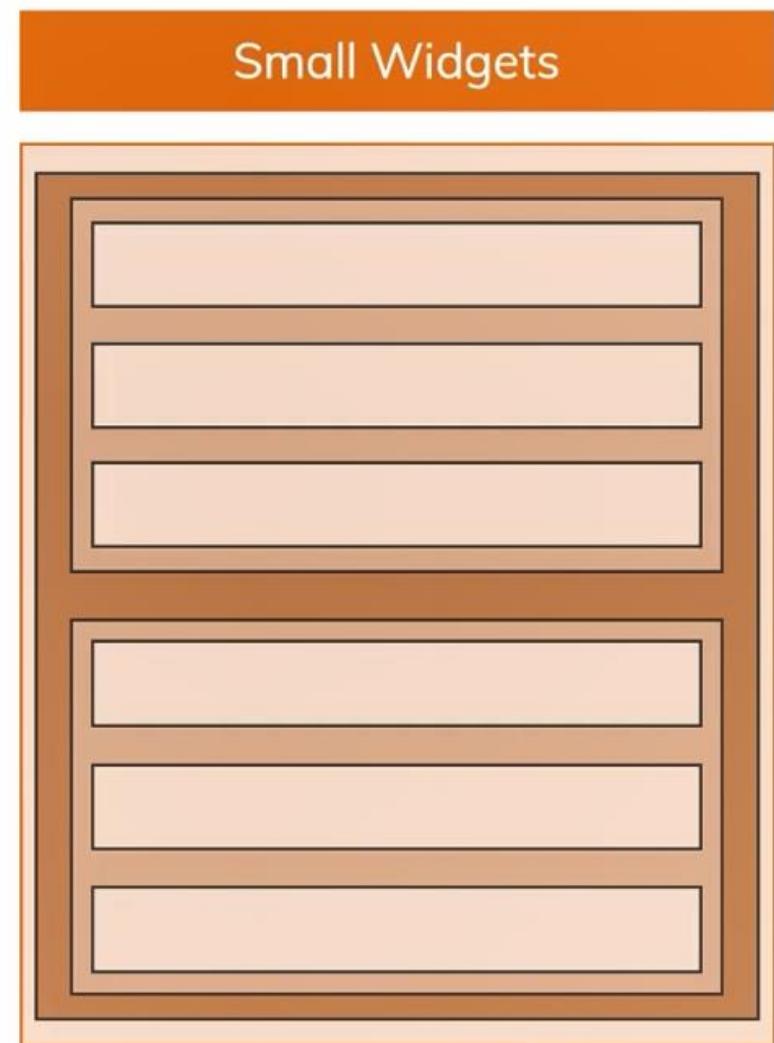
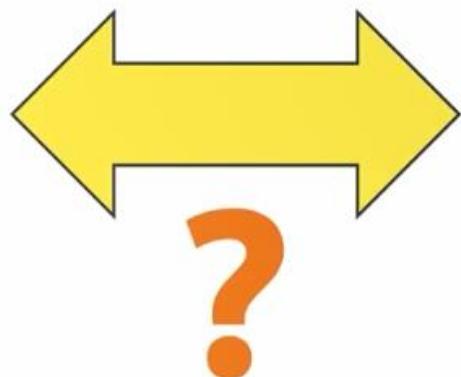
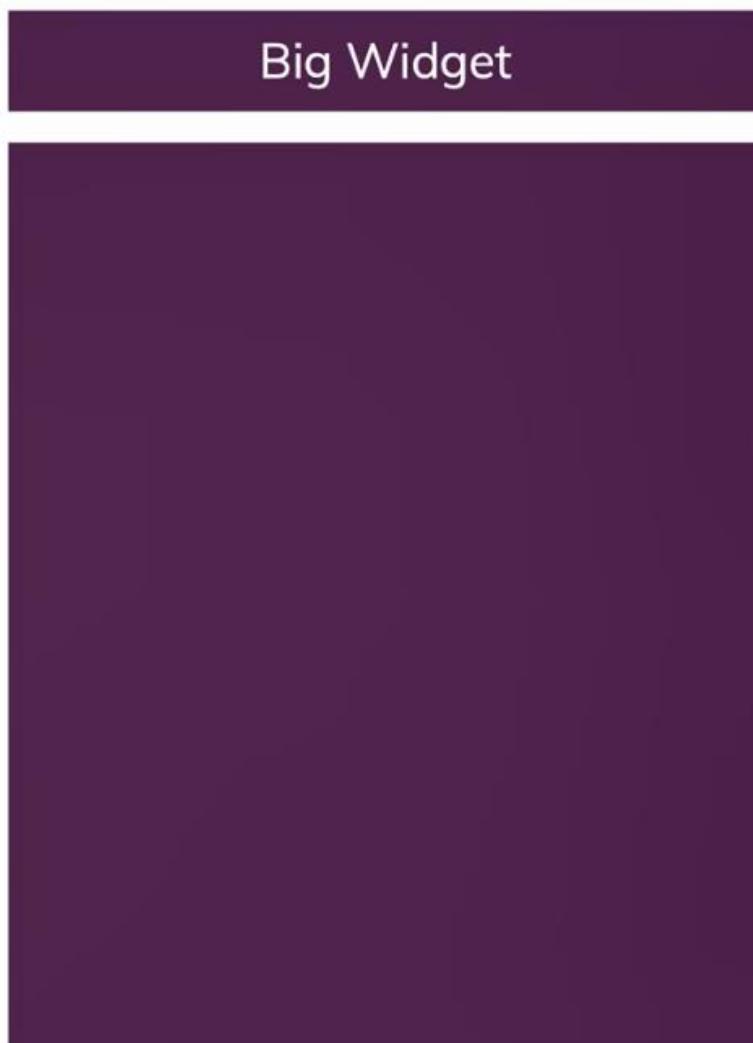
SPLITTING THE APP INTO WIDGETS

Should I Split It?

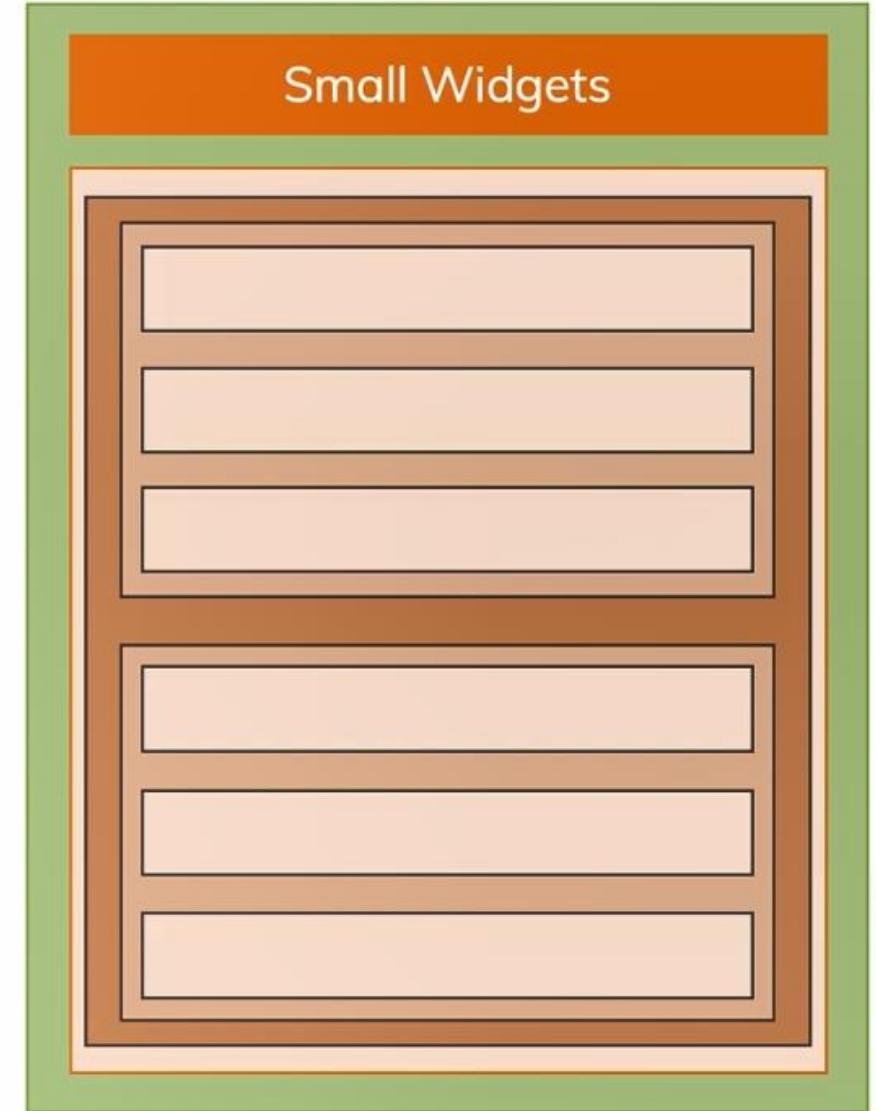
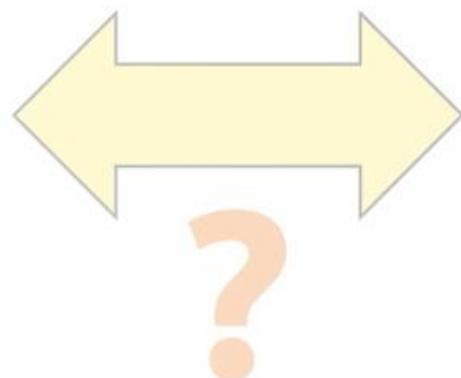
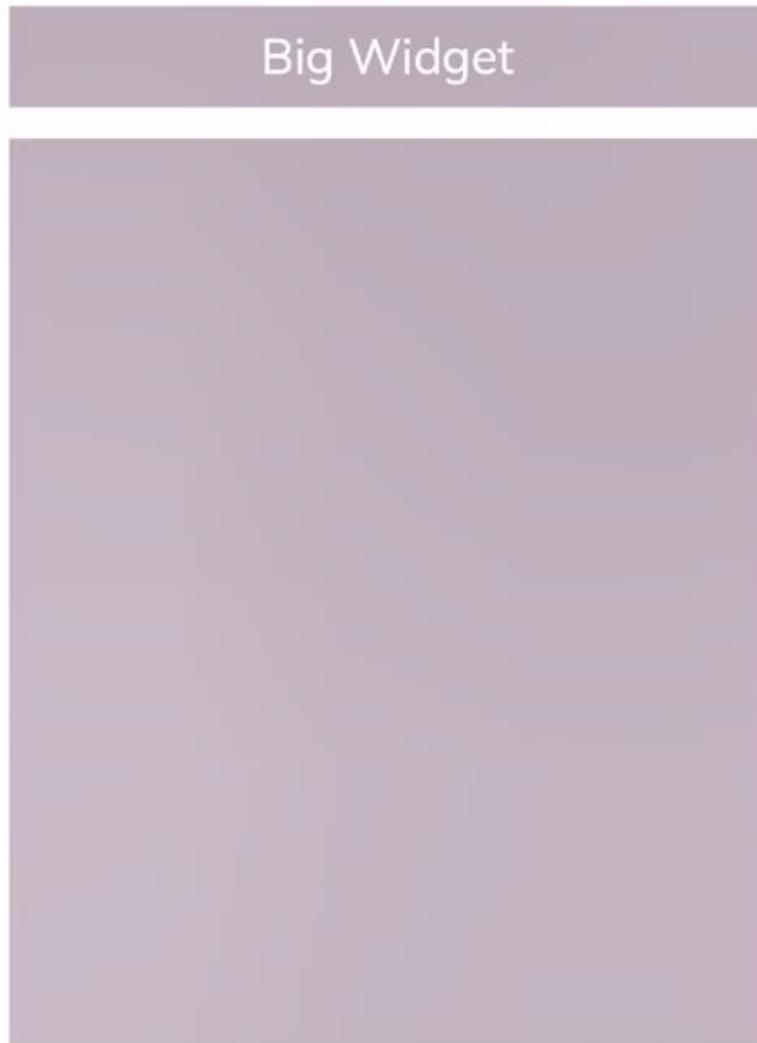
Big Widget

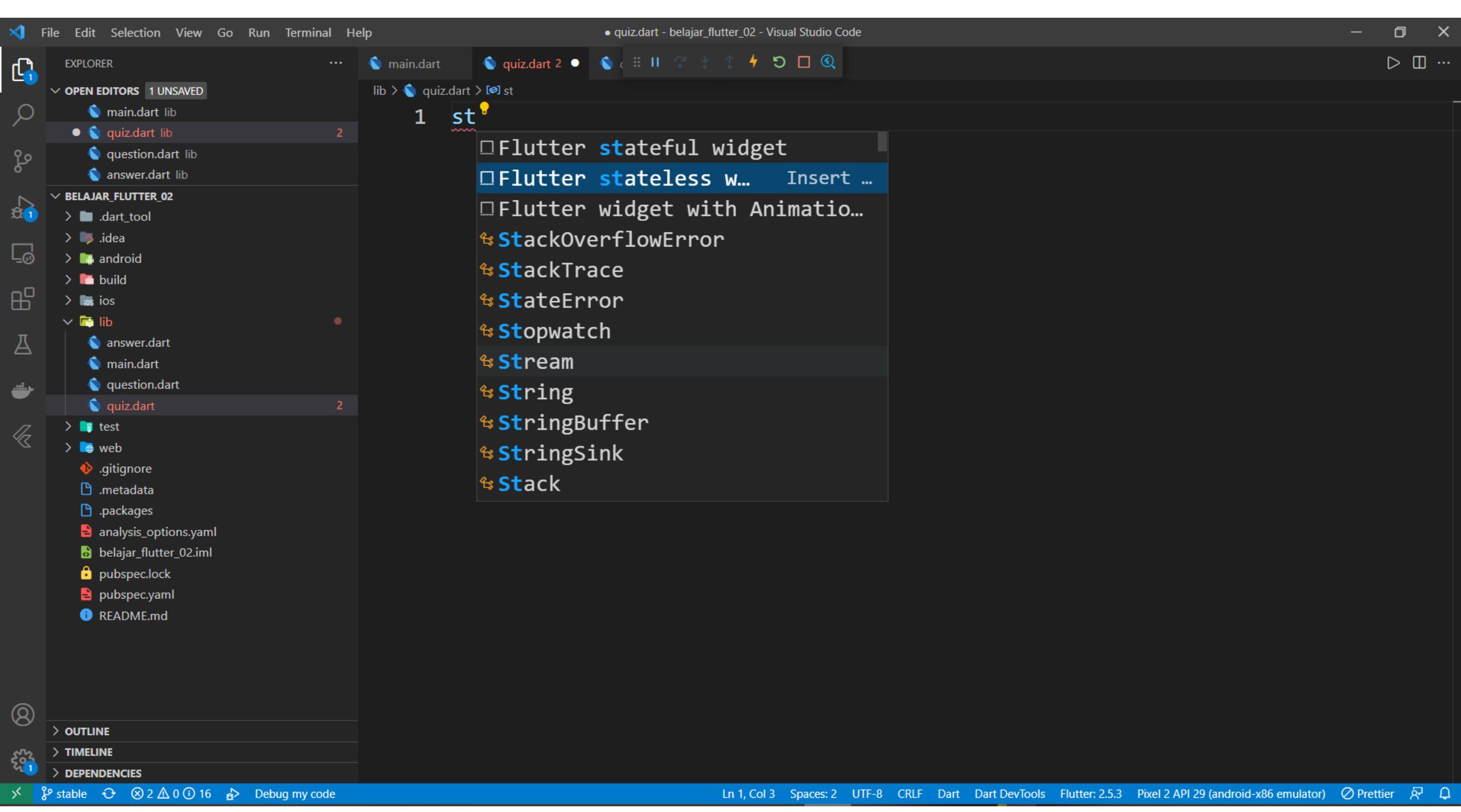


Should I Split It?



Should I Split It?





File Edit Selection View Go Run Terminal Help

quiz.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart quiz.dart result.dart wer.dart

lib > quiz.dart > Quiz > build

```
1 import 'package:flutter/material.dart';
2 import './question.dart';
3 import './answer.dart';
4
5 class Quiz extends StatelessWidget {
6   final List<Map<String, Object>> questions;
7   final int questionIndex;
8   final Function answerQuestion;
9
10 Quiz({
11   @required this.questions,
12   @required this.answerQuestion,
13   @required this.questionIndex,
14 });
15
16 @override
17 Widget build(BuildContext context) {
18   return Column(
19     children: [
20       Question(
21         questions[questionIndex]['questionText'],
22       ), // Question
23     ],
24   );
25 }
26
27 void main() {
28   runApp(MyApp());
29 }
```

Relative Position

Detorator ; Mandatory

Ln 24, Col 25 Spaces: 2 UTF-8 CRLF Dart Dart Dev Snip & Sketch 5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

File Edit Selection View Go Run Terminal Help quiz.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart quiz.dart result.dart wer.dart

OPEN EDITORS lib > quiz.dart > Quiz > build

BELAJAR_FLUTTER_02 .dart_tool .idea android build ios lib answer.dart main.dart question.dart quiz.dart result.dart test web .gitignore .metadata .packages analysis_options.yaml belajar_flutter_02.iml pubspec.lock pubspec.yaml README.md

11 @required this.questions,
12 @required this.answerQuestion,
13 @required this.questionIndex,
14);
15
16 @override
17 Widget build(BuildContext context) {
18 return Column(
19 children: [
20 Question(
21 questions[questionIndex]['questionText'],
22), // Question
23 ...(questions[questionIndex]['answers'] as List<String>).map((answer) {
24 return Answer(answerQuestion, answer);
25 }).toList()
26],
27); // Column
28 }
29 }
30 }

> OUTLINE
> TIMELINE
> DEPENDENCIES

stable Debug my code Ln 24, Col 25 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

File Edit Selection View Go Run Terminal Help question.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- quiz.dart lib
- result.dart lib
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib

 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart

- test
- web
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

question.dart - Question > build

```
1 import 'package:flutter/material.dart';
2
3 class Question extends StatelessWidget {
4     final String questionText;
5
6     Question(this.questionText);
7
8     @override
9     Widget build(BuildContext context) {
10         return Container(
11             width: double.infinity,
12             margin: EdgeInsets.all(10),
13             child: Text(
14                 questionText,
15                 style: TextStyle(fontSize: 28),
16                 textAlign: TextAlign.center,
17             ), // Text
18         ); // Container
19     }
20 }
21 
```

Ln 16, Col 23 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

File Edit Selection View Go Run Terminal Help

answer.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- quiz.dart lib
- result.dart lib
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
- answer.dart
- main.dart
- question.dart
- quiz.dart
- result.dart

test

web

.gitignore

.metadata

.packages

analysis_options.yaml

belajar_flutter_02.iml

pubspec.lock

pubspec.yaml

README.md

OUTLINE

TIMELINE

DEPENDENCIES

main.dart quiz.dart result.dart answer.dart

lib > answer.dart > Answer > build

```
1 import 'package:flutter/material.dart';
2
3 class Answer extends StatelessWidget {
4   final Function selectHandler;
5   final String answerText;
6
7   Answer(this.selectHandler, this.answerText);
8
9   @override
10  Widget build(BuildContext context) {
11    return Container(
12      width: double.infinity,
13      child: RaisedButton(
14        color: Colors.blue,
15        textColor: Colors.white,
16        child: Text(answerText),
17        onPressed: selectHandler,
18      ), // RaisedButton
19    ); // Container
20  }
21 }
22 }
```

Ln 20, Col 4 (3 selected) Spaces: 2 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

A screenshot of the Visual Studio Code interface showing a Flutter project named 'belajar_flutter_02'. The project structure is visible in the Explorer sidebar, and the code for 'result.dart' is displayed in the main editor area.

File **Edit** **Selection** **View** **Go** **Run** **Terminal** **Help**

result.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- quiz.dart lib
- result.dart lib**
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart**
- test
- web
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

OUTLINE

TIMELINE

DEPENDENCIES

stable 0 △ 0 ① 18 Debug my code

Ln 11, Col 1 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

```
1 import 'package:flutter/material.dart';
2
3 class Result extends StatelessWidget {
4     @override
5     Widget build(BuildContext context) {
6         return Center(
7             child: Text('You did it!'),
8         ); // Center
9     }
10 }
11
```

The screenshot shows the Visual Studio Code interface with a Dart project open. The project structure is as follows:

- lib**:
 - main.dart
 - quiz.dart
 - result.dart
 - question.dart
 - answer.dart
- BELAJAR_FLUTTER_02**:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md

The `main.dart` file is the active editor, showing the following code:

```
1 import 'package:flutter/material.dart';
2 import './quiz.dart';
3 import './result.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatefulWidget {
8   @override
9   State<StatefulWidget> createState() {
10     return _MyAppState();
11   }
12 }
13
14 class _MyAppState extends State<MyApp> {
15   var _questionIndex = 0;
16   final _questions = const [
17     {
18       'questionText': 'What\'s your favorite color?',
19       'answers': ['Black', 'Red', 'Green', 'White'],
20     },
21     {
22       'questionText': 'What\'s your favorite number?',
23       'answers': ['One', 'Two', 'Three', 'Four'],
24     },
25   ];
26   void _nextQuestion() {
27     setState(() {
28       if (_questionIndex < _questions.length - 1) {
29         _questionIndex++;
30       } else {
31         Navigator.pushNamed(context, '/result');
32       }
33     });
34   }
35 }
```

The status bar at the bottom provides build information: Ln 50, Col 33, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier, and a connection status icon.

The screenshot shows the Visual Studio Code interface with the following details:

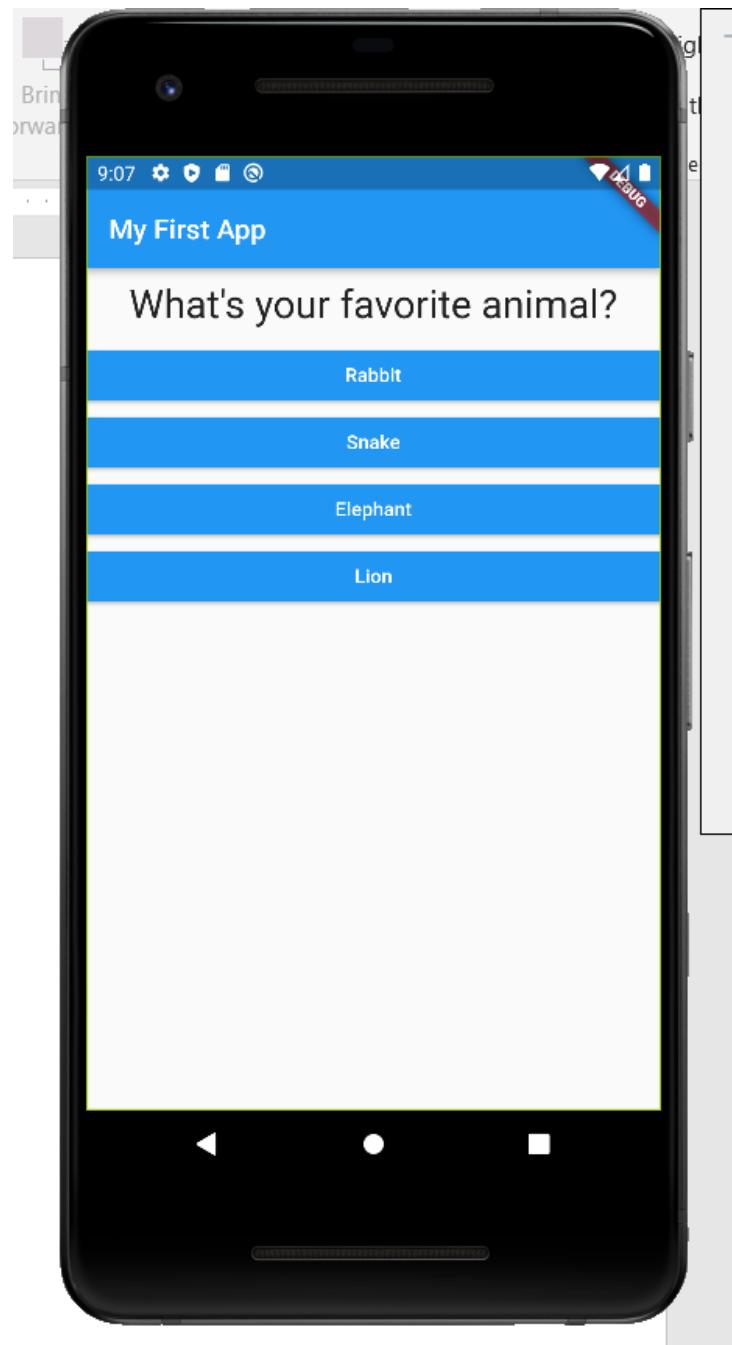
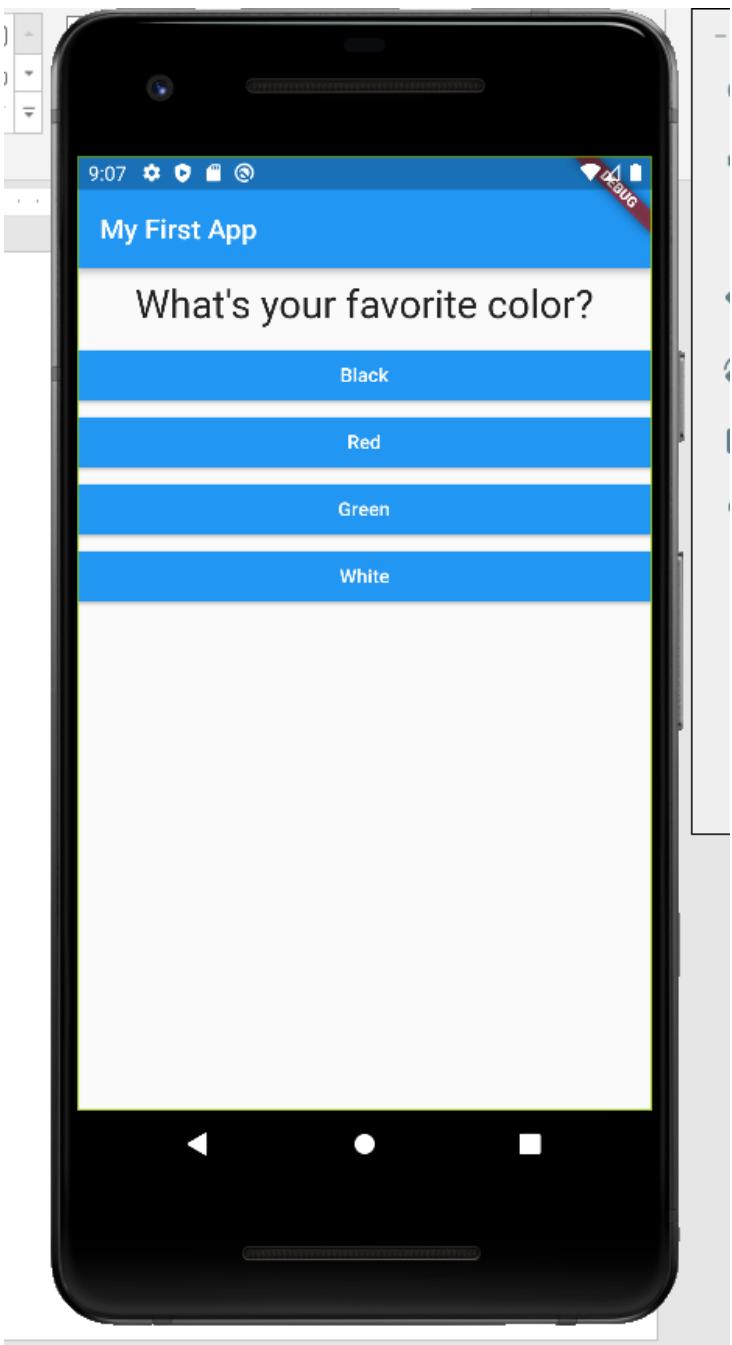
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code.
- Explorer Bar (Left):**
 - OPEN EDITORS: main.dart lib (highlighted), quiz.dart lib, result.dart lib, question.dart lib, answer.dart lib.
 - BELAJAR_FLUTTER_02: .dart_tool, .idea, android, build, ios, lib (highlighted), answer.dart, main.dart, question.dart, quiz.dart, result.dart.
 - Others: test, web, .gitignore, .metadata, .packages, analysis_options.yaml, belajar_flutter_02.iml, pubspec.lock, pubspec.yaml, README.md.
- Editor Area (Center):** The code editor displays the following Dart code:

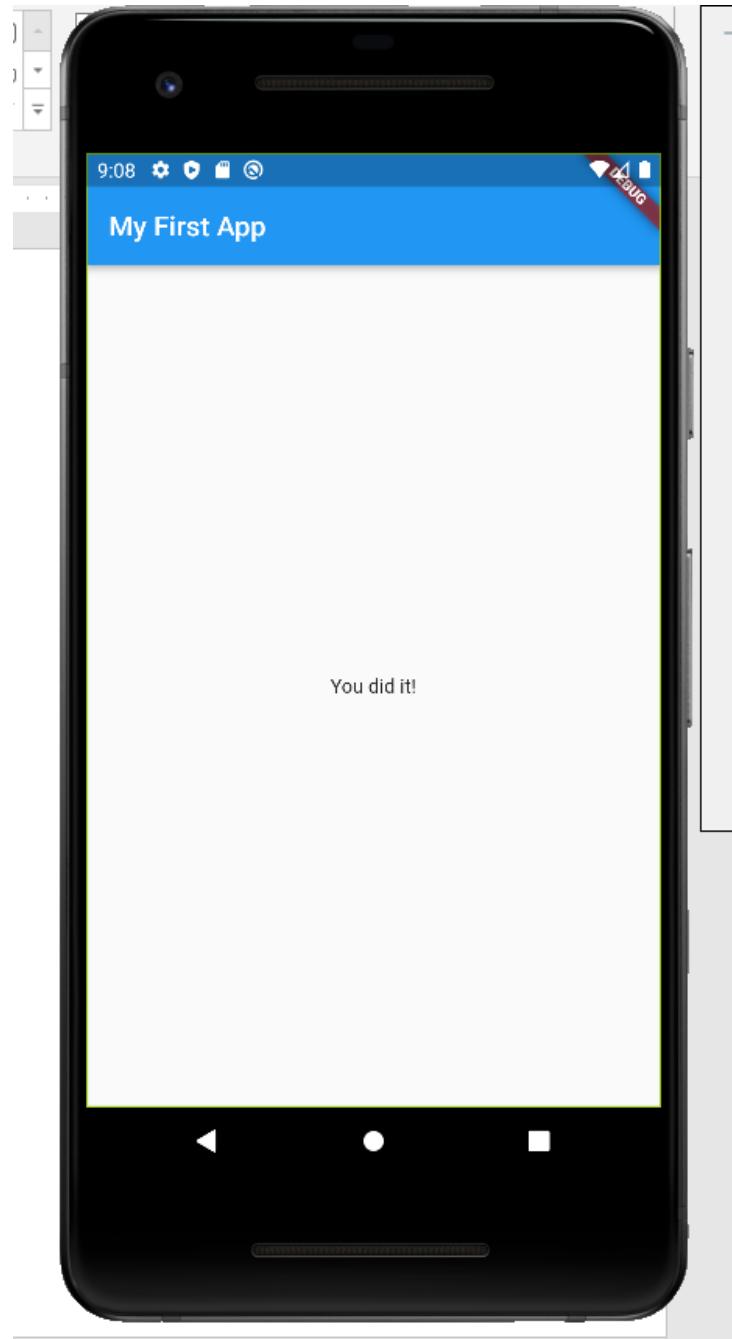
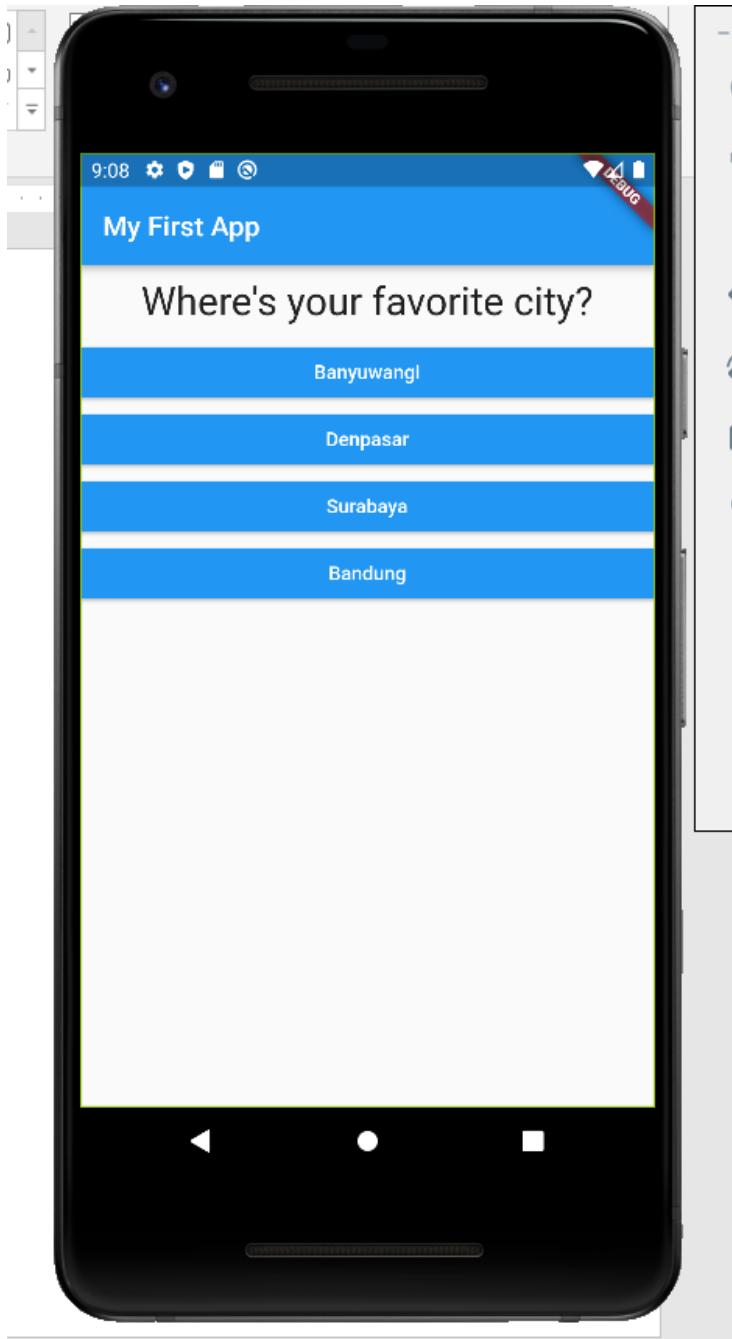
```
21  {
22      'questionText': 'What\'s your favorite animal?',
23      'answers': ['Rabbit', 'Snake', 'Elephant', 'Lion'],
24  },
25  {
26      'questionText': 'Where\'s your favorite city?',
27      'answers': ['Banyuwangi', 'Denpasar', 'Surabaya', 'Bandung'],
28  },
29 ];
30
31 void _answerQuestion() {
32     setState(() {
33         _questionIndex = _questionIndex + 1;
34     });
35     print(_questionIndex);
36     if (_questionIndex < _questions.length) {
37         print('We have more questions!');
38     } else {
39         print('No more questions!');
40     }
41 }
42 }
```
- Bottom Status Bar:** Ln 50, Col 33, Spaces: 2, Android Emulator - Pixel_2_API_29:5554, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier, and other icons.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code
- Left Sidebar (Explorer):**
 - OPEN EDITORS:** main.dart lib (selected), quiz.dart, result.dart, question.dart lib, answer.dart lib.
 - BELAJAR_FLUTTER_02:** .dart_tool, .idea, android, build, ios, lib (selected), answer.dart, main.dart, question.dart, quiz.dart, result.dart, test, web, .gitignore, .metadata, .packages, analysis_options.yaml, belajar_flutter_02.iml, pubspec.lock, pubspec.yaml, README.md.
- Central Area:** Code editor showing Dart code for a Flutter application.

```
main.dart - belajar_flutter_02 - Visual Studio Code
main.dart X quiz.dart result.dart build
lib > main.dart > _MyAppState > build
40    }
41    }
42
43 @override
44 Widget build(BuildContext context) {
45   return MaterialApp(
46     home: Scaffold(
47       appBar: AppBar(
48         title: Text("My First App"),
49       ), // AppBar
50       body: _questionIndex < _questions.length
51         ? Quiz(
52           answerQuestion: _answerQuestion,
53           questionIndex: _questionIndex,
54           questions: _questions,
55         ) // Quiz
56         : Result(),
57     ), // Scaffold
58   ); // MaterialApp
59 }
60 }
61 }
```
- Bottom Status Bar:** Ln 50, Col 33, Spaces: 2, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.





CALCULATING THE TOTAL SCORE

A screenshot of the Visual Studio Code interface showing a Dart file named `result.dart` in the editor. The code defines a class `_MyAppState` that extends `State<MyApp>`. It contains variables for the current question index and total score, and a list of questions. Each question has a text and a list of answers with their respective scores.

```
14 class _MyAppState extends State<MyApp> {
15     var _questionIndex = 0;
16     var _totalScore = 0;
17     final _questions = const [
18         {
19             'questionText': 'What\'s your favorite color?',
20             'answers': [
21                 {'text': 'Black', 'score': 10},
22                 {'text': 'Red', 'score': 5},
23                 {'text': 'Green', 'score': 3},
24                 {'text': 'White', 'score': 1},
25             ],
26         },
27         {
28             'questionText': 'What\'s your favorite animal?',
29             'answers': [
30                 {'text': 'Rabbit', 'score': 3},
31                 {'text': 'Snake', 'score': 11},
32                 {'text': 'Elephant', 'score': 5},
33                 {'text': 'Lion', 'score': 9},
34             ],
35         },
36     ];
37 }
```

The Explorer sidebar shows the project structure under the `BELAJAR_FLUTTER_02` folder, including files like `main.dart`, `quiz.dart`, and `result.dart`. The status bar at the bottom indicates the code is stable, using spaces for indentation, and is running a Gradle task.

File Edit Selection View Go Run Terminal Help main.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart X quiz.dart result.dart wer.dart

lib > main.dart > _MyAppState > _questions

35 },
36 {
37 'questionText': 'Where\'s your favorite city?',
38 'answers': [
39 {'text': 'Banyuwangi', 'score': 1},
40 {'text': 'Denpasar', 'score': 1},
41 {'text': 'Surabaya', 'score': 1},
42 {'text': 'Bandung', 'score': 1},
43],
44 },
45];
46
47 void _answerQuestion(int score) {
48 _totalScore += score;
49 setState(() {
50 _questionIndex = _questionIndex + 1;
51 });
52 print(_questionIndex);
53 if (_questionIndex < _questions.length) {
54 print('We have more questions!');
55 } else {
56 print('No more questions!');

stable 0 △ 0 ① 18 Debug my code Ln 42, Col 39 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.dart - belajar_flutter_02 - Visual Studio Code.
- Left Sidebar (Explorer):**
 - OPEN EDITORS: main.dart lib (selected), quiz.dart, result.dart, question.dart lib, answer.dart lib.
 - BELAJAR_FLUTTER_02: .dart_tool, .idea, android, build, ios, lib (selected), test, web, .gitignore, .metadata, .packages, analysis_options.yaml, belajar_flutter_02.iml, pubspec.lock, pubspec.yaml, README.md.
- Central Area:** Code editor showing Dart code for a Flutter application.

```
main.dart - belajar_flutter_02 - Visual Studio Code
main.dart X quiz.dart result.dart _MyAppState.dart answer.dart

lib/main.dart:50 PRINT('No more questions!');

57     }
58 }
59
60 @override
61 Widget build(BuildContext context) {
62     return MaterialApp(
63         home: Scaffold(
64             appBar: AppBar(
65                 title: Text("My First App"),
66             ), // AppBar
67             body: _questionIndex < _questions.length
68                 ? Quiz(
69                     answerQuestion: _answerQuestion,
70                     questionIndex: _questionIndex,
71                     questions: _questions,
72                 ) // Quiz
73                 : Result(),
74         ), // Scaffold
75     ); // MaterialApp
76 }
77 }
78

Ln 17, Col 1 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier
```
- Bottom Status Bar:** stable, 0△0 19, Debug my code, Ln 17, Col 1, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.
- Bottom Right:** Running Gradle task 'assembleDebug'...

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** quiz.dart - belajar_flutter_02 - Visual Studio Code.
- Left Sidebar (Explorer):**
 - OPEN EDITORS: main.dart, quiz.dart (selected), result.dart, answer.dart.
 - BELAJAR_FLUTTER_02 folder structure:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart (selected)
 - result.dart
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
 - OUTLINE, TIMELINE, and DEPENDENCIES sections.
- Central Area:** Code editor showing the `quiz.dart` file content.

```
1 import 'package:flutter/material.dart';
2 import './question.dart';
3 import './answer.dart';
4
5 class Quiz extends StatelessWidget {
6   final List<Map<String, Object>> questions;
7   final int questionIndex;
8   final Function answerQuestion;
9
10 Quiz({
11   @required this.questions,
12   @required this.answerQuestion,
13   @required this.questionIndex,
14 });
15
16 @override
17 Widget build(BuildContext context) {
18   return Column(
19     children: [
20       Question(
21         questions[questionIndex]['questionText'] as String,
22       ), // Question
23     ],
24   );
25 }
26
```
- Bottom Status Bar:** Ln 21, Col 53, Spaces: 2, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.

A screenshot of the Visual Studio Code interface showing a Dart file named `quiz.dart` in the editor. The code is part of a class definition, likely `Quiz`, which contains methods for building a UI with `Column` and `Question` widgets. The code uses `@required` annotations and `Map<String, Object>` types. The editor shows line numbers from 11 to 32. The status bar at the bottom indicates the code is stable, has 19 issues, and is running on Pixel 2 API 29 (android-x86 emulator) with Flutter 2.5.3.

```
File Edit Selection View Go Run Terminal Help quiz.dart - belajar_flutter_02 - Visual Studio Code EXPLORER ... main.dart quiz.dart result.dart wer.dart lib > quiz.dart > Quiz > build 11 @required this.questions, 12 @required this.answerQuestion, 13 @required this.questionIndex, 14 ); 15 16 @override 17 Widget build(BuildContext context) { 18   return Column( 19     children: [ 20       Question( 21         questions[questionIndex]['questionText'] as String, 22       ), // Question 23       ...(questions[questionIndex]['answers'] as List<Map<String, Object>>) 24         .map((answer) { 25           return Answer( 26             () => answerQuestion(answer['score']), answer['text'] as String); // A 27           }).toList() 28       ], 29     ); // Column 30   } 31 } 32 
```

File Edit Selection View Go Run Terminal Help quiz.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER ... main.dart quiz.dart result.dart wer.dart lib > quiz.dart > Quiz > build

11 @required this.questions,
12 @required this.answerQuestion,
13 @required this.questionIndex,
14);
15
16 @override
17 Widget build(BuildContext context) {
18 return Column(
19 children: [
20 Question(
21 questions[questionIndex]['questionText'] as String,
22), // Question
23 ...(questions[questionIndex]['answers'] as List<Map<String, Object>>)
24 .map((answer) {
25 return Answer(
26 () => answerQuestion(answer['score']), answer['text'] as String); // A
27 }).toList()
28],
29); // Column
30 }
31 }
32

Ln 21, Col 53 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

GETTERS AND ELSE-IF

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** result.dart - belajar_flutter_02 - Visual Studio Code
- Left Sidebar (Explorer):**
 - OPEN EDITORS:
 - main.dart
 - quiz.dart
 - result.dart (selected)
 - question.dart
 - answer.dart
 - BELAJAR_FLUTTER_02:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart (selected)
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Central Area:** The code editor displays the content of the result.dart file. The file defines a class Result that extends StatelessWidget. It contains a final int resultScore variable and a String resultPhrase method that returns a string based on the resultScore value. The code uses Dart's conditional operators and return statements to determine the resultText string.

```
1 import 'package:flutter/material.dart';
2
3 class Result extends StatelessWidget {
4     final int resultScore;
5
6     String get resultPhrase {
7         String resultText;
8         if (resultScore <= 8) {
9             resultText = 'You are awesome and innocent!';
10        } else if (resultScore <= 12) {
11            resultText = 'Pretty likeable!';
12        } else if (resultScore <= 16) {
13            resultText = 'You are ... strange?!';
14        } else {
15            resultText = 'You are so bad!';
16        }
17        return resultText;
18    }
19
20    Result(this.resultScore);
21    @override
22    Widget build(BuildContext context) {
```
- Bottom Status Bar:** Ln 25, Col 22, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier, and other status indicators.

A screenshot of the Visual Studio Code interface, showing the code editor with Dart code for a result page. The file is named `result.dart` and is part of the `lib` directory.

The code defines a class `Result` with a constructor `Result(resultScore)` and a `build` method. The `build` method returns a `Center` widget containing a `Text` widget with the `resultPhrase` and `resultText`.

```
resultText = 'Pretty likeable!';
} else if (resultScore <= 16) {
  resultText = 'You are ... strange?!';
} else {
  resultText = 'You are so bad!';
}
return resultText;
}

Result(this.resultScore);
@Override
Widget build(BuildContext context) {
  return Center(
    child: Text(
      resultPhrase,
      style: TextStyle(fontSize: 36, fontWeight: FontWeight.bold),
      textAlign: TextAlign.center,
    ), // Text
  ); // Center
}
}
```

The code editor shows line numbers from 11 to 32. A yellow warning icon is visible near line 25. The status bar at the bottom indicates the code is stable, has 18 changes, and is running on a Pixel 2 API 29 (android-x86 emulator).

EXPLORER

OPEN EDITORS

- main.dart lib
- quiz.dart lib
- result.dart lib
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart
- test
- web
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

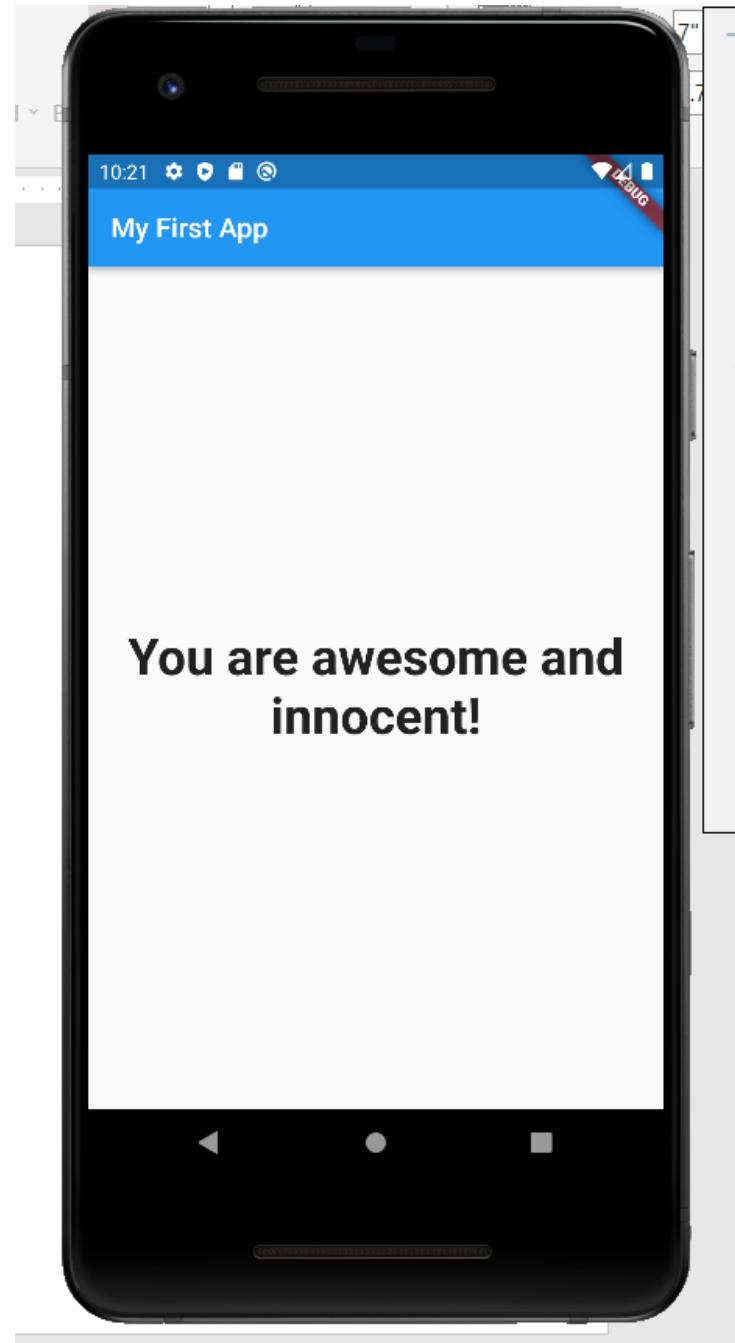
OUTLINE

TIMELINE

DEPENDENCIES

main.dart X quiz.dart result.dart **_MyAppState** _questions

```
56     print("No more questions! ");
57 }
58 }
59
60 @override
61 Widget build(BuildContext context) {
62     return MaterialApp(
63         home: Scaffold(
64             appBar: AppBar(
65                 title: Text("My First App"),
66             ), // AppBar
67             body: _questionIndex < _questions.length
68                 ? Quiz(
69                     answerQuestion: _answerQuestion,
70                     questionIndex: _questionIndex,
71                     questions: _questions,
72                 ) // Quiz
73                 : Result(_totalScore),
74         ), // Scaffold
75     ); // MaterialApp
76 }
77 }
78 }
```



RESETTING THE QUIZ

A screenshot of the Visual Studio Code interface showing a Dart file named `main.dart` from a project titled "belajar_flutter_02". The code implements a quiz application with a scaffold, appBar, and body sections. It includes logic for resetting the quiz and displaying results.

```
main.dart - belajar_flutter_02 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... main.dart X quiz.dart result.dart ... wer.dart
OPEN EDITORS lib > main.dart > _MyAppState
main.dart lib
quiz.dart lib
result.dart lib
question.dart lib
answer.dart lib
BELAJAR_FLUTTER_02
  .dart_tool
  .idea
  android
  build
  ios
  lib
    answer.dart
    main.dart
    question.dart
    quiz.dart
    result.dart
  test
  web
  .gitignore
  .metadata
  .packages
  analysis_options.yaml
  belajar_flutter_02.iml
  pubspec.lock
  pubspec.yaml
  README.md
OUTLINE
TIMELINE
DEPENDENCIES
stable Debug my code
Ln 67, Col 1 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier
```

```
void _resetQuiz() {
  setState(() {
    _questionIndex = 0;
    _totalScore = 0;
  });
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text("My First App"),
      ), // AppBar
      body: _questionIndex < _questions.length
        ? Quiz(
            answerQuestion: _answerQuestion,
            questionIndex: _questionIndex,
            questions: _questions,
          ) // Quiz
        : Result(_totalScore, _resetQuiz),
    ), // Scaffold
}
```

File Edit Selection View Go Run Terminal Help

result.dart - belajar_flutter_02 - Visual Studio Code

EXPLORER

OPEN EDITORS

- main.dart lib
- quiz.dart lib
- result.dart lib**
- question.dart lib
- answer.dart lib

BELAJAR_FLUTTER_02

- .dart_tool
- .idea
- android
- build
- ios
- lib
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart**
- test
- web
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- belajar_flutter_02.iml
- pubspec.lock
- pubspec.yaml
- README.md

OUTLINE

TIMELINE

DEPENDENCIES

main.dart quiz.dart result.dart wer.dart

lib > result.dart > Result > resultPhrase

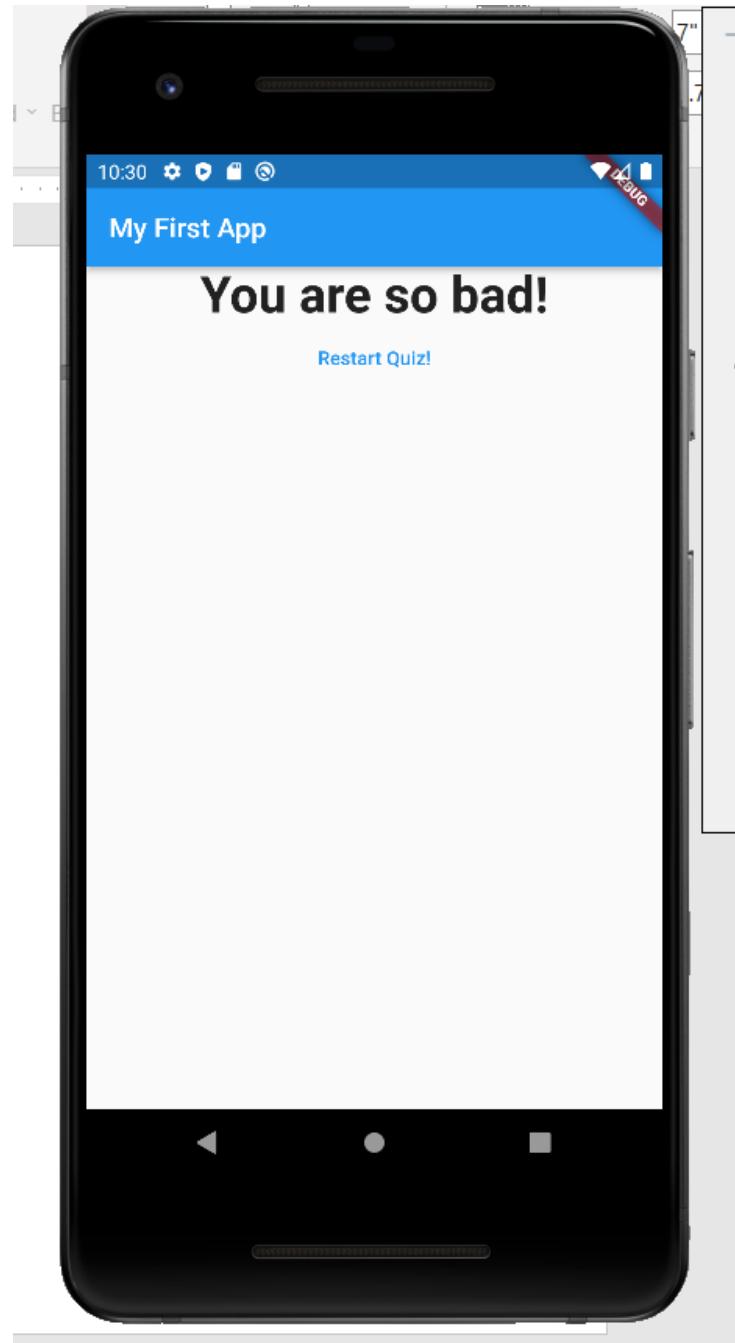
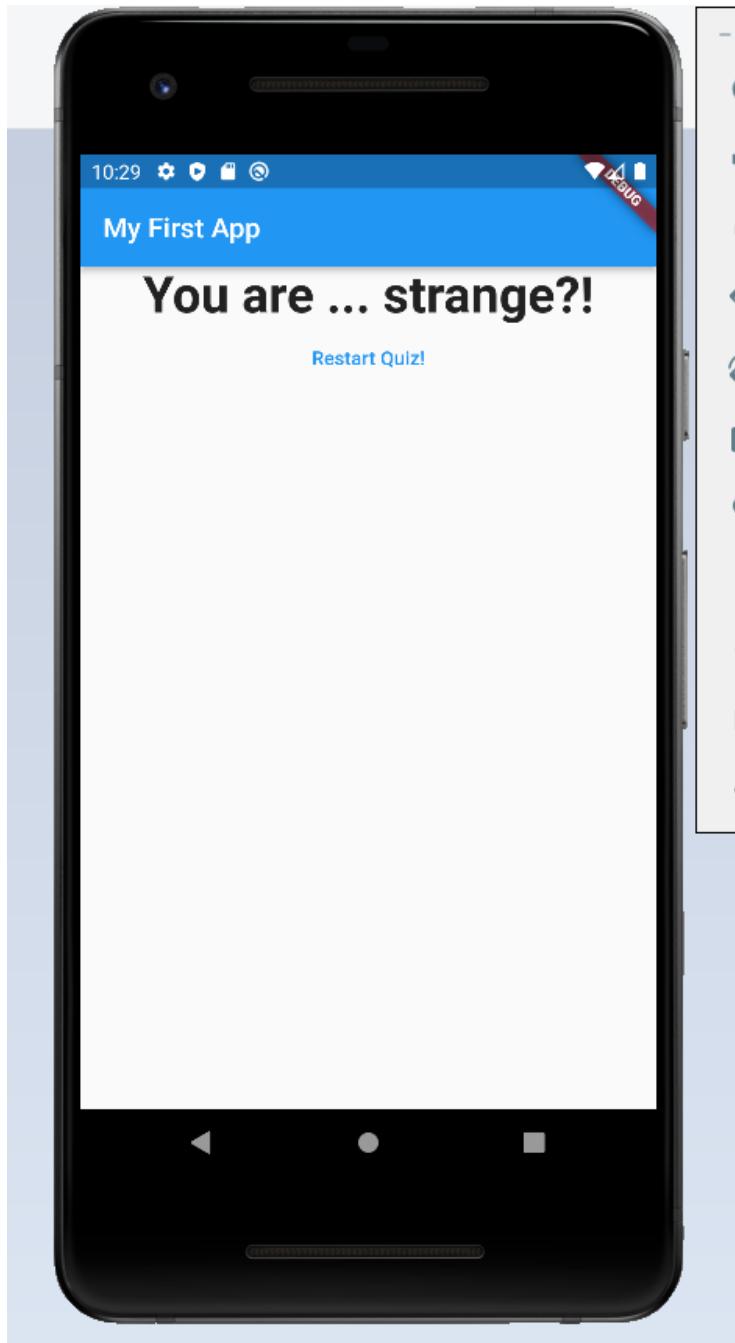
```
1 import 'package:flutter/material.dart';
2
3 class Result extends StatelessWidget {
4   final int resultScore;
5   final Function resetHandler;
6
7   Result(this.resultScore, this.resetHandler);
8
9   String get resultPhrase {
10    String resultText;
11    if (resultScore <= 8) {
12      resultText = 'You are awesome and innocent!';
13    } else if (resultScore <= 12) {
14      resultText = 'Pretty likeable!';
15    } else if (resultScore <= 16) {
16      resultText = 'You are ... strange?!';
17    } else {
18      resultText = 'You are so bad!';
19    }
20    return resultText;
21  }
22}
```

Ln 18, Col 33 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 2.5.3 Pixel 2 API 29 (android-x86 emulator) Prettier

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** result.dart - belajar_flutter_02 - Visual Studio Code.
- Explorer View (Left):**
 - OPEN EDITORS: main.dart, quiz.dart, result.dart (selected), question.dart, answer.dart.
 - BELAJAR_FLUTTER_02 folder:
 - .dart_tool
 - .idea
 - android
 - build
 - ios
 - lib folder:
 - answer.dart
 - main.dart
 - question.dart
 - quiz.dart
 - result.dart (selected)
 - test
 - web
 - .gitignore
 - .metadata
 - .packages
 - analysis_options.yaml
 - belajar_flutter_02.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Code Editor (Center):** The code for the `result.dart` file is displayed. The cursor is at line 44. The code uses `Text` and `FlatButton` widgets to display a result phrase and a restart button.

```
22
23 @override
24 Widget build(BuildContext context) {
25   return Center(
26     child: Column(
27       children: <Widget>[
28         Text(
29           resultPhrase,
30           style: TextStyle(fontSize: 36, fontWeight: FontWeight.bold),
31           textAlign: TextAlign.center,
32         ), // Text
33         FlatButton(
34           child: Text(
35             'Restart Quiz!',
36           ), // Text
37           textColor: Colors.blue,
38           onPressed: resetHandler,
39         ), // FlatButton
40       ], // <Widget>[]
41     ), // Column
42   ); // Center
43 }
44 }
```
- Bottom Status Bar:** Ln 18, Col 33, Spaces: 2, UTF-8, CRLF, Dart, Dart DevTools, Flutter: 2.5.3, Pixel 2 API 29 (android-x86 emulator), Prettier.

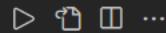


NEW FLUTTER BUTTONS VERSUS OLD BUTTONS



main.dart

Settings X



6 Settings Found

Turn on Settings Sync

deprecated

User Workspace

Text Editor (2)

Suggestions (1)

Application (1)

Telemetry (1)

Extensions (3)

Docker (1)

Go (1)

Python (1)

Editor: Show Deprecated Controls strikethrough deprecated variables.**Editor > Suggest: Show Deprecated** When enabled IntelliSense shows **deprecated**-suggestions.**Telemetry: Telemetry Level**

Controls all core and first party extension telemetry. This helps us to better understand how Visual Studio Code is performing, where improvements need to be made, and how features are being used. Read more about the [data we collect](#) and our [privacy statement](#). A full restart of the application is necessary for crash reporting changes to take effect.

The following table outlines the data sent with each setting:

Crash Reports Error Telemetry Usage Data

all	✓	✓	✓
error	✓	✓	-
crash	✓	-	-
off	-	-	-

Note: If this setting is 'off', no telemetry will be sent regardless of other telemetry settings. If this setting is set to anything except 'off' and telemetry is disabled with deprecated settings, no telemetry will be sent.

all

**Go: Language Server Experimental Features**

Temporary flag to enable/disable diagnostics from the language server. This setting will be deprecated soon. Please see and response to [Issue 50](#).

 If true, the language server will provide build, vet errors and the extension will ignore the 'buildOnSave', 'vetOnSave' s...

main.dart X

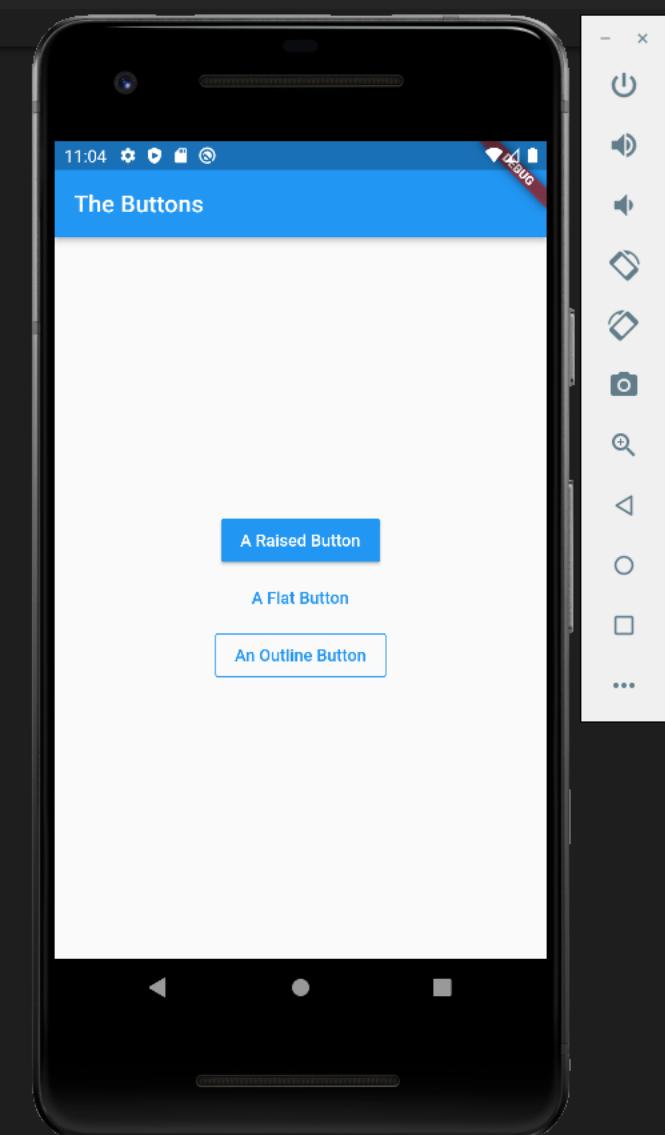
lib > main.dart > MyApp > build

```
1 import 'package:flutter/material.dart';
2
3     Run | Debug | Profile
4 void main() {
5     runApp(MyApp());
6 }
7 class MyApp extends StatelessWidget {
8     @override
9     Widget build(BuildContext context) {
10         return MaterialApp(
11             home: Scaffold(
12                 appBar: AppBar(
13                     title: Text('The Buttons'),
14                 ), // AppBar
15                 body: Center(
16                     child: Column(
17                         mainAxisAlignment: MainAxisAlignment.center,
18                         children: [
19                             RaisedButton(
20                                 color: Colors.blue,
21                                 textColor: Colors.white,
```



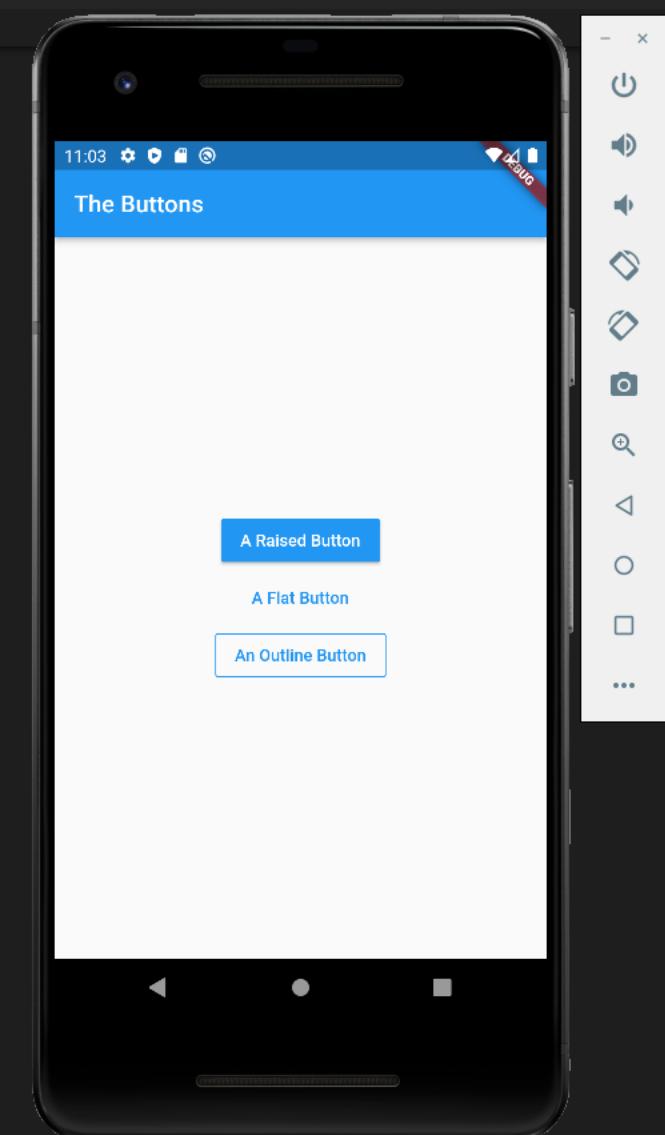
main.dart X
lib > main.dart > MyApp > build

```
18      children: [
19        RaisedButton(
20          color: Colors.blue,
21          textColor: Colors.white,
22          child: Text('A Raised Button'),
23          onPressed: () {
24            print('Pressed raised button');
25          },
26        ), // RaisedButton
27        FlatButton(
28          child: Text('A Flat Button'),
29          textColor: Colors.blue,
30          onPressed: () {
31            print('Pressed flat button');
32          },
33        ), // FlatButton
34        OutlineButton(
35          child: Text('An Outline Button'),
36          borderSide: BorderSide(color: Colors.blue),
37          textColor: Colors.blue,
38          onPressed: () {
39            print('Pressed outline button');
```



main.dart X
lib > main.dart > MyApp > build

```
28     child: Text('A Raised Button'),
29     textColor: Colors.blue,
30     onPressed: () {
31         print('Pressed raised button');
32     },
33 ), // RaisedButton
34 OutlineButton(
35     child: Text('An Outline Button'),
36     borderSide: BorderSide(color: Colors.blue),
37     textColor: Colors.blue,
38     onPressed: () {
39         print('Pressed outline button');
40     },
41 ), // OutlineButton
42 ],
43 ), // Column
44 ), // Center
45 ), // Scaffold
46 ); // MaterialApp
47 }
48 }
49 }
```

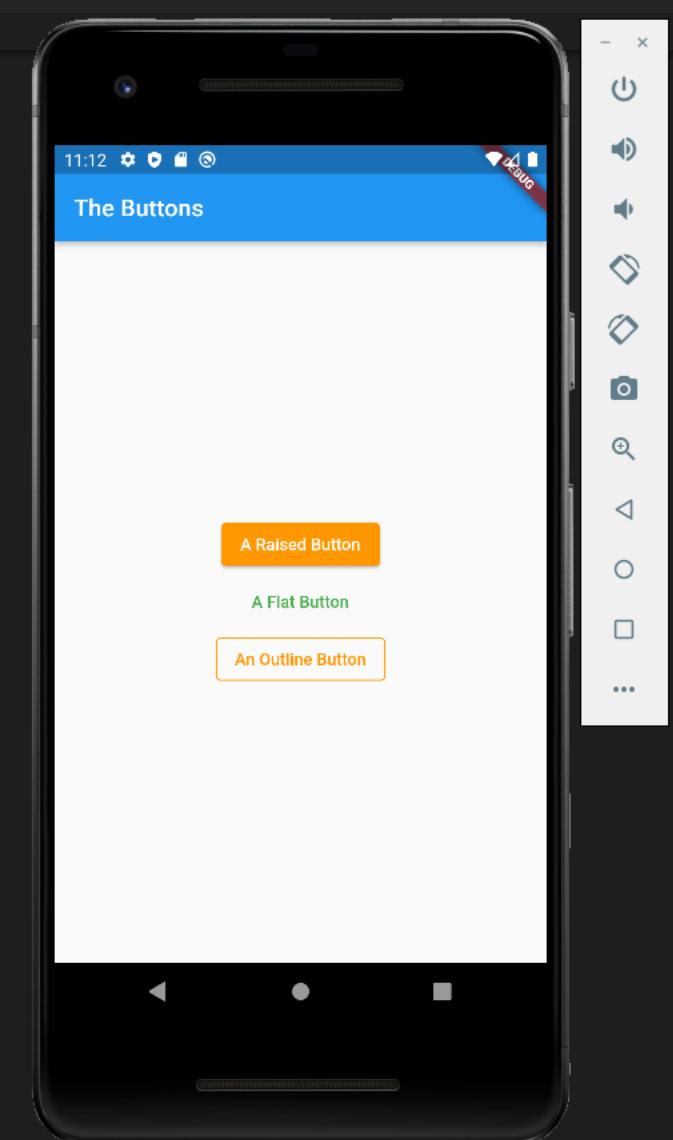


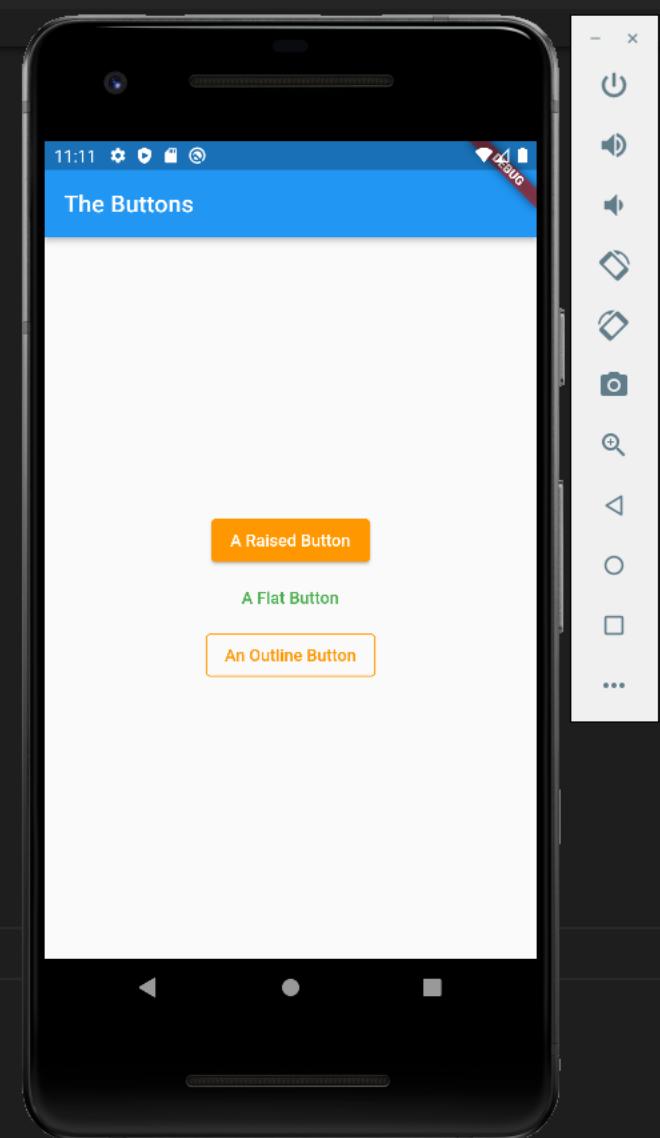


main.dart X

lib > main.dart > MyApp > build

```
17     mainAxisAlignment: MainAxisAlignment.center,  
18     children: [  
19       ElevatedButton(  
20         style: ElevatedButton.styleFrom(  
21           primary: Colors.orange,  
22           onPrimary: Colors.white,  
23         ),  
24         child: Text('A Raised Button'),  
25         onPressed: () {  
26           print('Pressed raised button');  
27         },  
28       ), // ElevatedButton  
29       TextButton(  
30         style: TextButton.styleFrom(  
31           primary: Colors.green,  
32         ),  
33         child: Text('A Flat Button'),  
34         onPressed: () {  
35           print('Pressed flat button');  
36         },  
37       ), // TextButton  
38       OutlinedButton(  
39         style: OutlinedButton.styleFrom(  
40           primary: Colors.purple,  
41           onPrimary: Colors.pink,  
42         ),  
43         child: Text('An Outline Button'),  
44         onPressed: () {  
45           print('Pressed outline button');  
46         },  
47       ), // OutlinedButton
```





Module Summary

Dart

- Object-oriented, strongly typed programming language
- Everything's an object!
- Types: String, num, int, double, List<...>
- Variables, Functions, Properties, Methods
- Constructor functions & shorthands

Widgets

- Your App's UI = Tree of Widgets
- Flutter controls every pixel on the screen
- Two types of Widgets: Stateless and Stateful Widgets
- Widgets have build() methods which return a Widget

Types of Widgets

- Stateful & Stateless
- Visible (Input/ Output) and Invisible (Layout/ Control)
- Complex, with lots of config options (e.g. MaterialApp) and simpler (e.g. Text)

Styling & Positioning

- Styling and positioning is done via code (=> in the Widget tree)
- Widgets can work together to achieve a certain look/ structure