Cross Platform Mobile Programming with flutter & Dart

Prof. Dr. Aydın SEÇER

Installation, Tools and Prerequisites

- Steps
- For Version Control System (Optional)
  - Linux, macOS, Windows: Install gitSCM software from: https://git-scm.com/
- Windows
  - 1-Install flutter (Set flutter **/bin** folder as environment path correctly)
  - 2-Install Visual Studio Code:
  - 3-Install JetBrains Rider (Optional)
  - 4-For Emulators:
    - Android: Install Android SDK and Device from Visual Studio IDE
    - Remote iOS: Use physical external MacBook or Virtual MAC operating system
      - Install X-Code development platform, set permission for remote file sharing.
- Mac OS
  - 1-Install flutter (Set flutter **/bin** folder as environment path correctly)
  - 2-Install Visual Studio For Mac
  - 3-Install JetBrains Rider (Optional) or Install VS-Code(Optional)
  - 4-For Emulators:
    - Android: Install Android SDK and Device from Visual Studio IDE
    - Local iOS: Install X-Code development platform
- Linux
  - 1-Install flutter (Set flutter **/bin** folder as environment path correctly)
  - 2-Install Visual Studio Code
  - 3-For Emulators:
    - Android: Install Android SDK and Device from Visual Studio IDE
    - Remote iOS: Install X-Code development platform

Part A
Dart Language

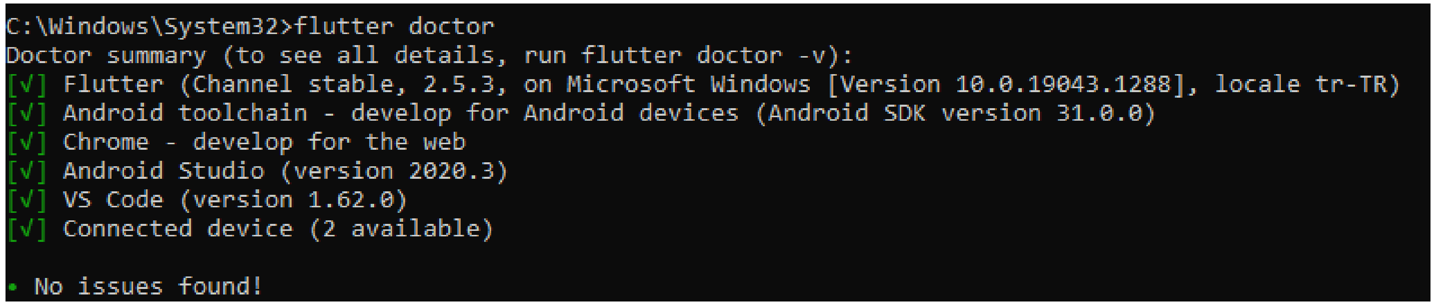
Dart Programming

Part B

Flutter Mobile App. Development

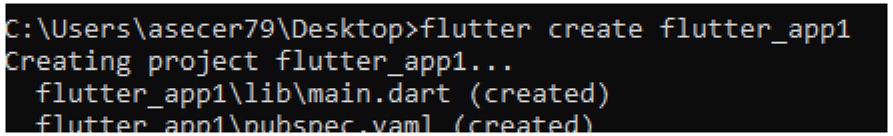
Creating a new flutter – Project FROM Terminal

- Step 1: Check flutter is ok or not! Type cmd line and Run: /> flutter doctor



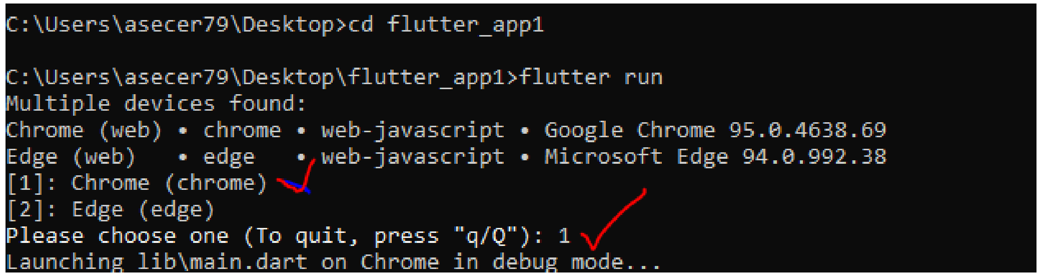
```
C:\Windows\System32>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 2.5.3, on Microsoft Windows [Version 10.0.19043.1288], locale tr-TR)
[√] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
[√] Chrome - develop for the web
[√] Android Studio (version 2020.3)
[√] VS Code (version 1.62.0)
[√] Connected device (2 available)

• No issues found!
```

- Step 2: Create flutter Project on desktop:



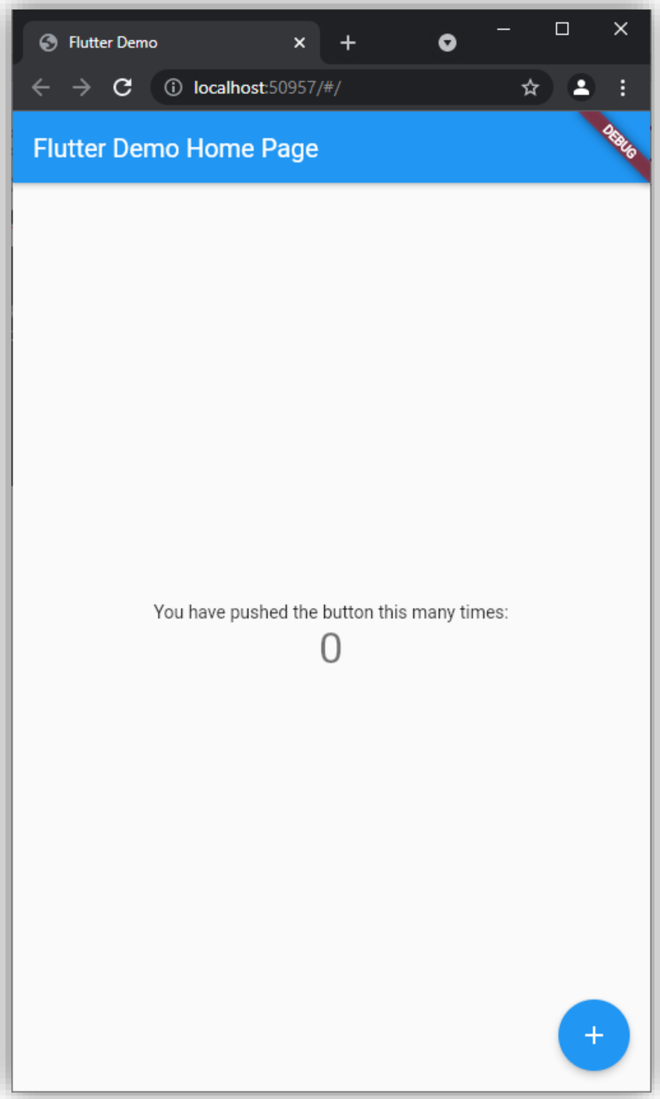
```
C:\Users\asecer79\Desktop>flutter create flutter_app1
Creating project flutter_app1...
  flutter_app1\lib\main.dart (created)
  flutter_app1\pubspec.yaml (created)
```

- Step 3: Run flutter Project from terminal

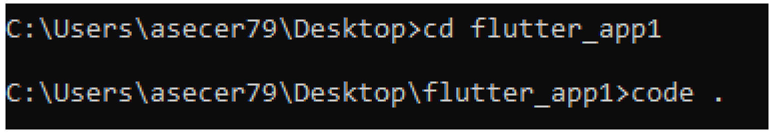


```
C:\Users\asecer79\Desktop>cd flutter_app1

C:\Users\asecer79\Desktop\flutter_app1>flutter run
Multiple devices found:
Chrome (web) • chrome • web-javascript • Google Chrome 95.0.4638.69
Edge (web)   • edge   • web-javascript • Microsoft Edge 94.0.992.38
[1]: Chrome (chrome)
[2]: Edge (edge)
Please choose one (To quit, press "q/Q"): 1
Launching lib\main.dart on Chrome in debug mode...
```

- OR

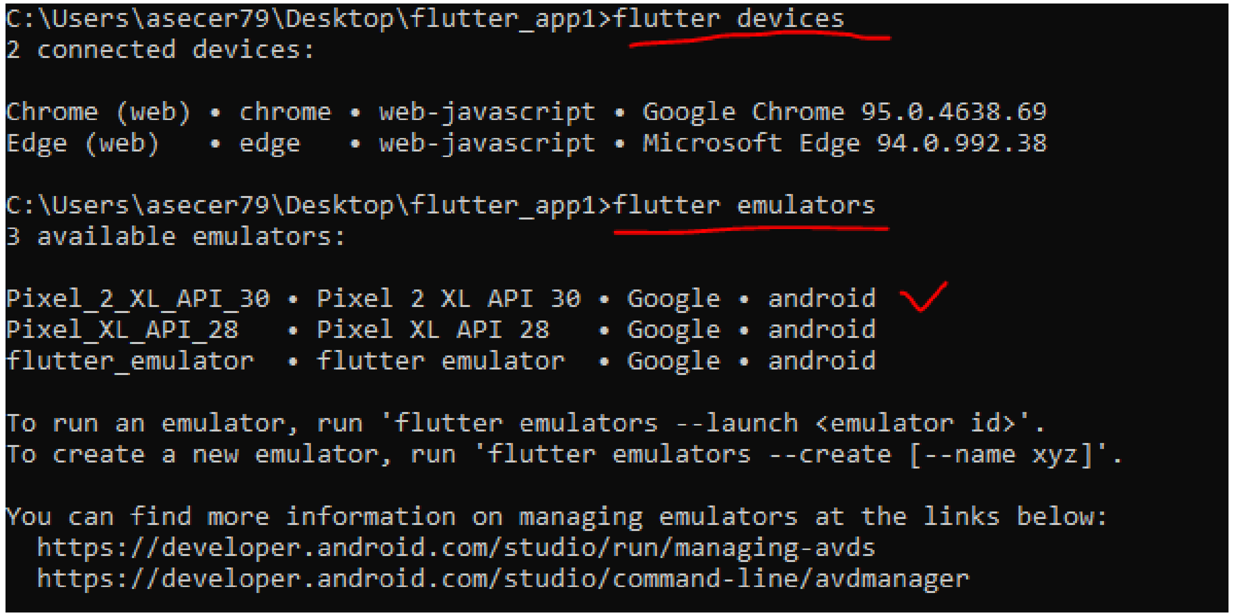


- Step 4: Open flutter Project on VSCode

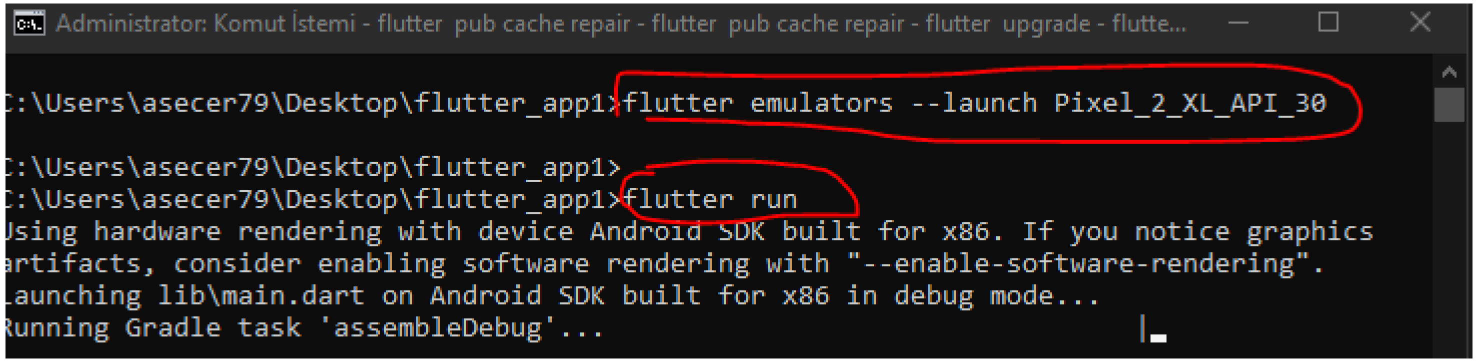


```
C:\Users\asecer79\Desktop>cd flutter_app1

C:\Users\asecer79\Desktop\flutter_app1>code .
```

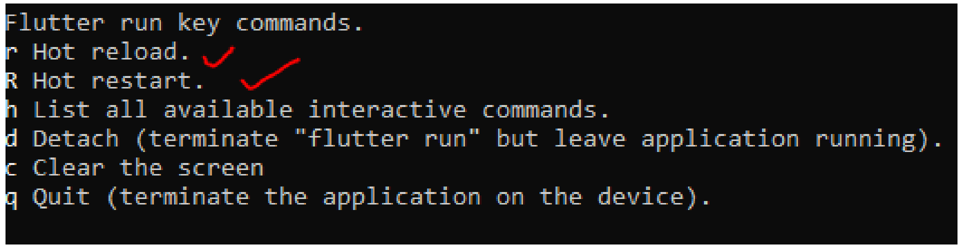
Running Emulators from Terminal



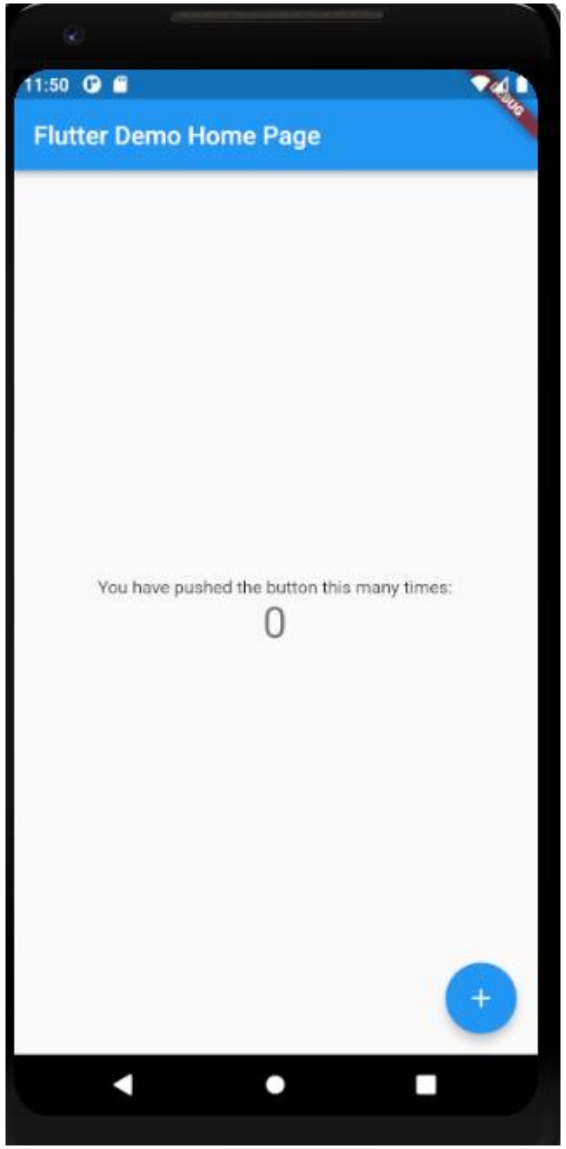
```
C:\Users\asecer79\Desktop\flutter_app1>flutter devices
2 connected devices:

Chrome (web) • chrome • web-javascript • Google Chrome 95.0.4638.69
Edge (web)   • edge   • web-javascript • Microsoft Edge 94.0.992.38

C:\Users\asecer79\Desktop\flutter_app1>flutter emulators
3 available emulators:

Pixel_2_XL_API_30 • Pixel 2 XL API 30 • Google • android ✓
Pixel_XL_API_28   • Pixel XL API 28   • Google • android
flutter_emulator  • flutter emulator  • Google • android

To run an emulator, run 'flutter emulators --launch <emulator id>'.
To create a new emulator, run 'flutter emulators --create [--name xyz]'.

You can find more information on managing emulators at the links below:
  https://developer.android.com/studio/run/managing-avds
  https://developer.android.com/studio/command-line/avdmanager
```



```
Administrator: Komut İstemi - flutter  pub cache repair - flutter  pub cache repair - flutter  upgrade - flutte...

:\Users\asecer79\Desktop\flutter_app1>flutter emulators --launch Pixel_2_XL_API_30

:\Users\asecer79\Desktop\flutter_app1>
:\Users\asecer79\Desktop\flutter_app1>flutter run
Jsing hardware rendering with device Android SDK built for x86. If you notice graphics
artifacts, consider enabling software rendering with "--enable-software-rendering".
.aunching lib\main.dart on Android SDK built for x86 in debug mode...
Running Gradle task 'assembleDebug'...
```

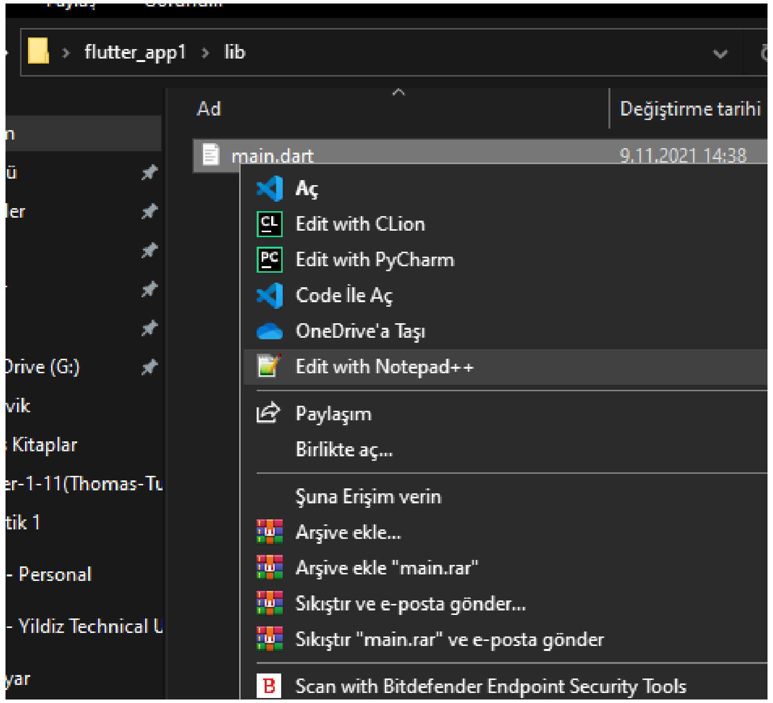


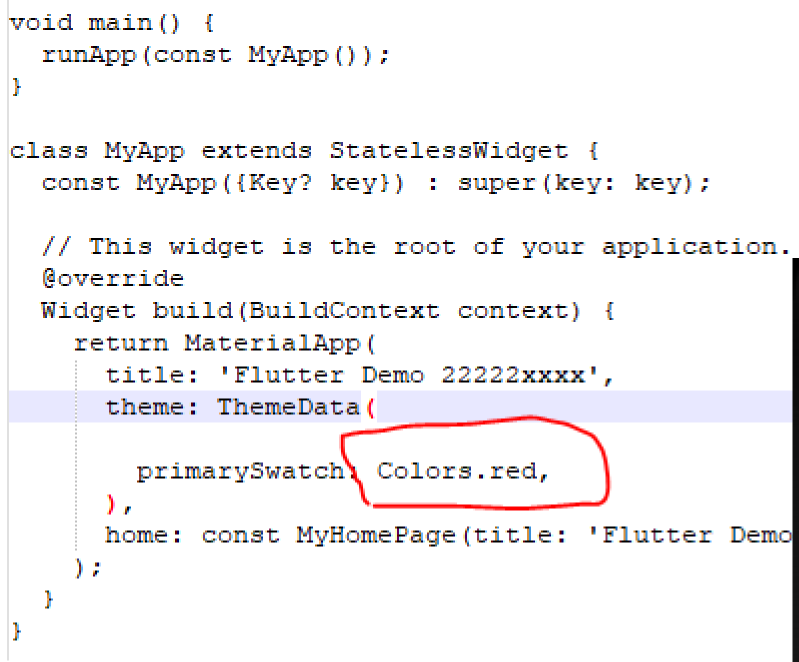
```
Flutter run key commands.
r Hot reload. ✓
R Hot restart. ✓
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).
```



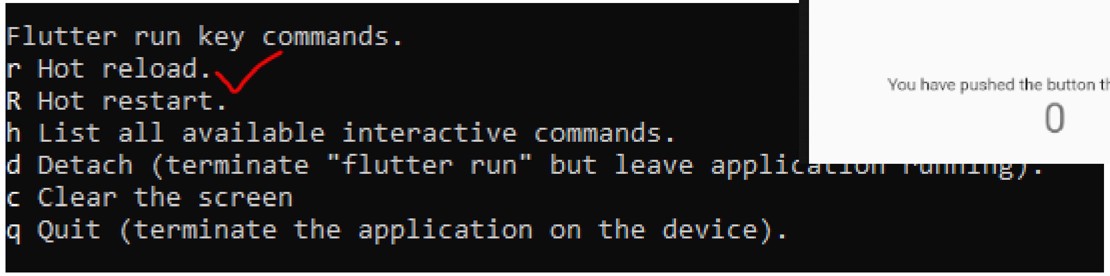
Hot Reload and Hot Restart

- Open main dart file from any editor. Modify anyhing and press key ‘r’ for hotreload. See the changes from emulator.

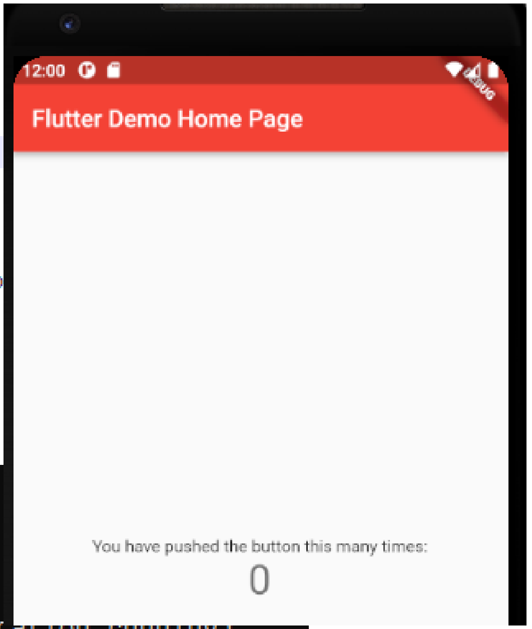




```
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo 22222xxxx',
      theme: ThemeData(

        primarySwatch: Colors.red,
      ),
      home: const MyHomePage(title: 'Flutter Demo
    );
  }
}
```

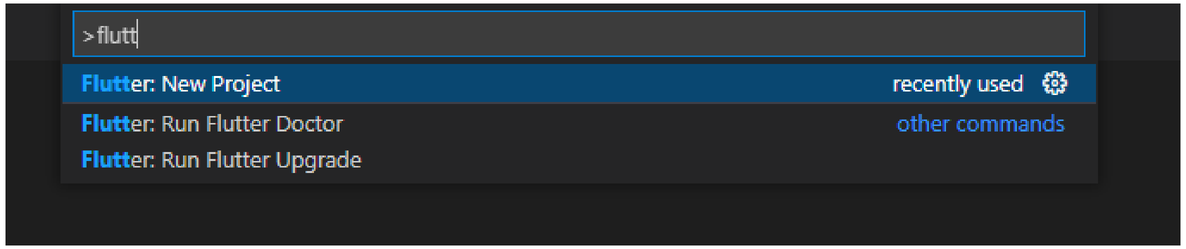


```
Flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave applic
c Clear the screen
q Quit (terminate the application on the device).
```



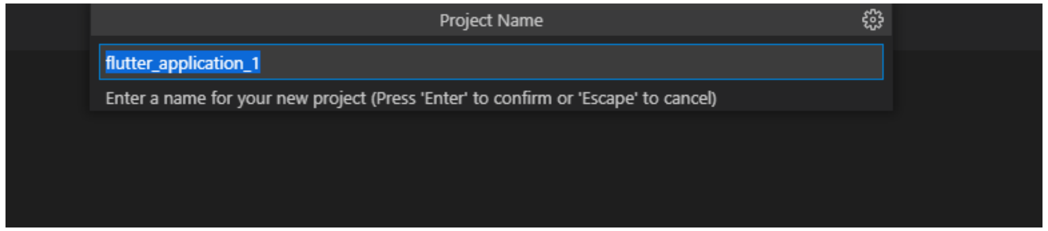
Creating a new flutter – Project FROM VSCode

- Step 1: Create an empty folder and open in VSCode
- Step 2: On VSCode press CTRL+ Shift + P , type flutter new project press enter, select Project type as Application and press again enter.

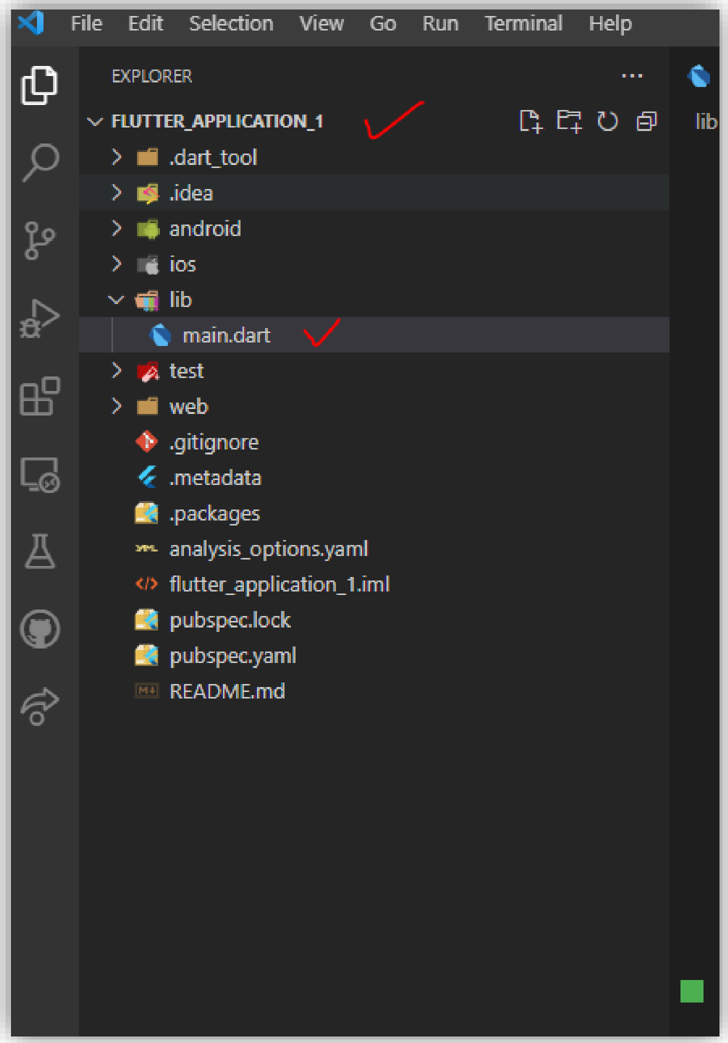


```
>flutt
Flutter: New Project                                  recently used
Flutter: Run Flutter Doctor                          other commands
Flutter: Run Flutter Upgrade
```

- Step 3: Select any other folder or current folder.
- Step 4: Enter new Project name and press enter.

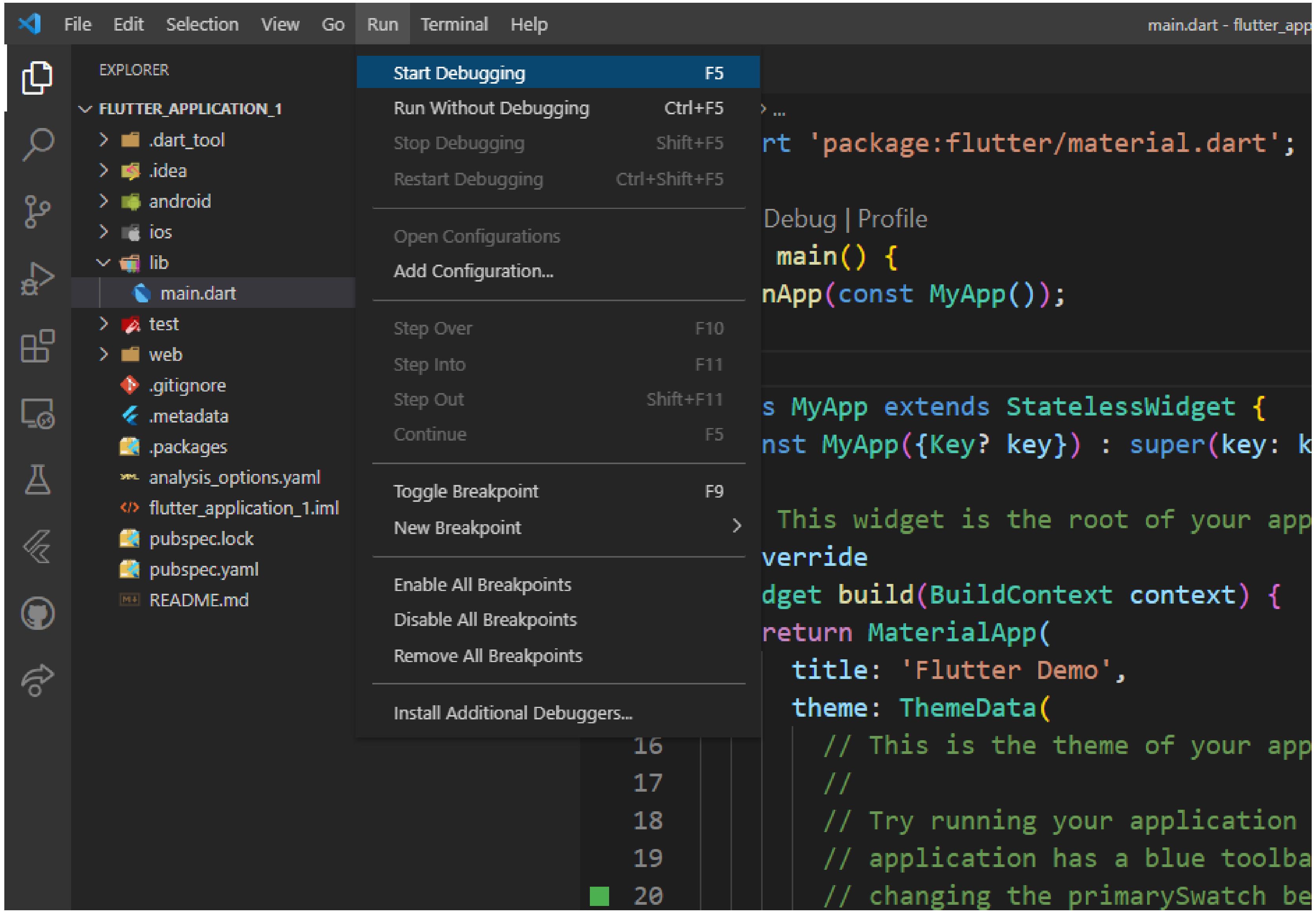


```
Project Name
flutter_application_1
Enter a name for your new project (Press 'Enter' to confirm or 'Escape' to cancel)
```

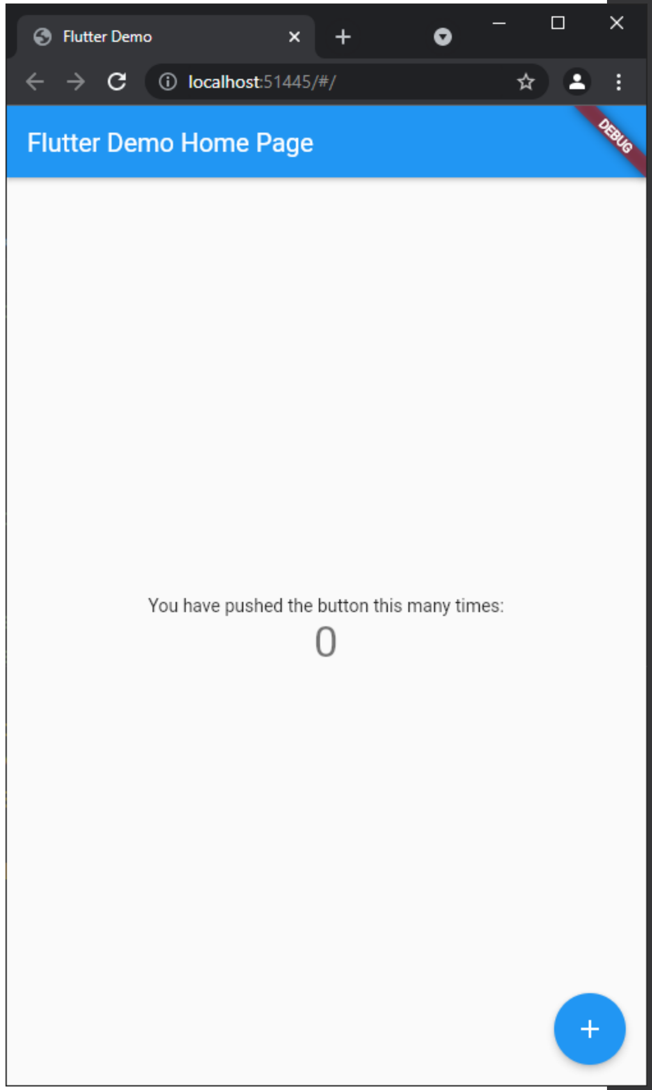


```
File  Edit  Selection  View  Go  Run  Terminal  Help
EXPLORER
FLUTTER_APPLICATION_1
  > .dart_tool
  > .idea
  > android
  > ios
  v lib
      main.dart
  > test
  > web
    .gitignore
    .metadata
    .packages
    analysis_options.yaml
    flutter_application_1.iml
    pubspec.lock
    pubspec.yaml
    README.md
```

Run and Debug Project from VSCode

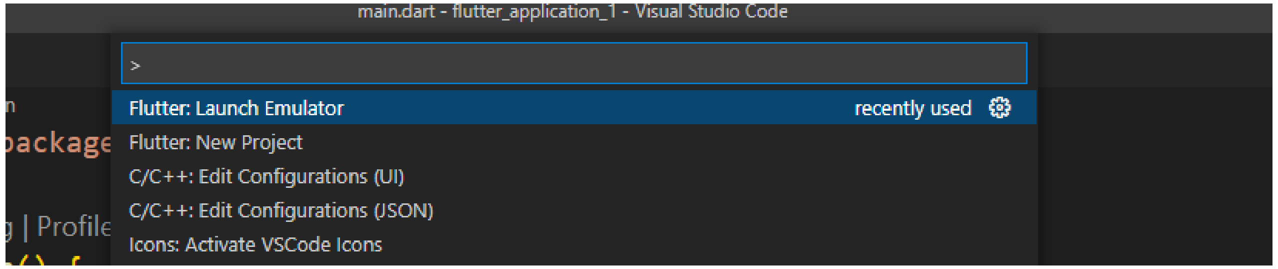


```
rt 'package:flutter/material.dart';

Debug | Profile
 main() {
nApp(const MyApp());

s MyApp extends StatelessWidget {
nst MyApp({Key? key}) : super(key: k

 This widget is the root of your app
verride
dget build(BuildContext context) {
return MaterialApp(
  title: 'Flutter Demo',
  theme: ThemeData(
    // This is the theme of your app
    //
    // Try running your application
    // application has a blue toolba
    // changing the primarySwatch be
```



Running Project on Emulator

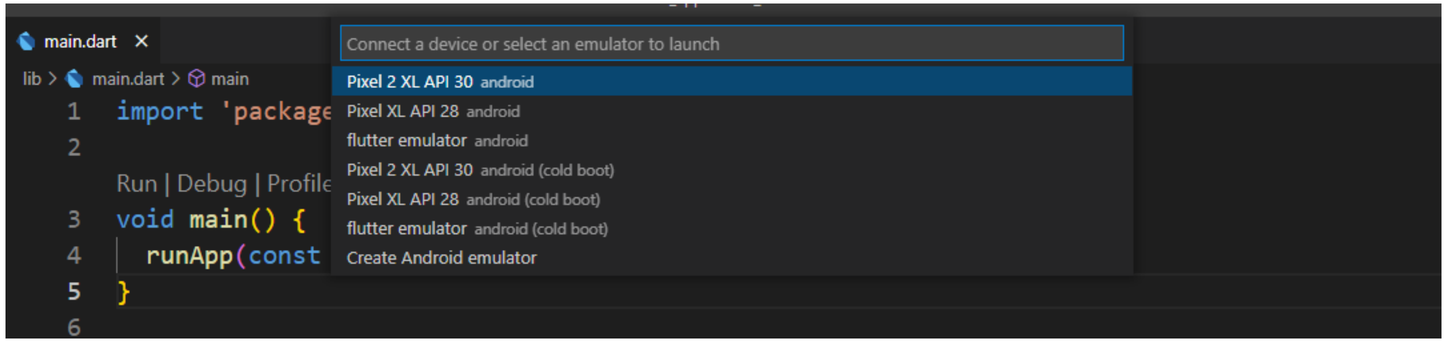
- Press CTLR+ Shift + P and type



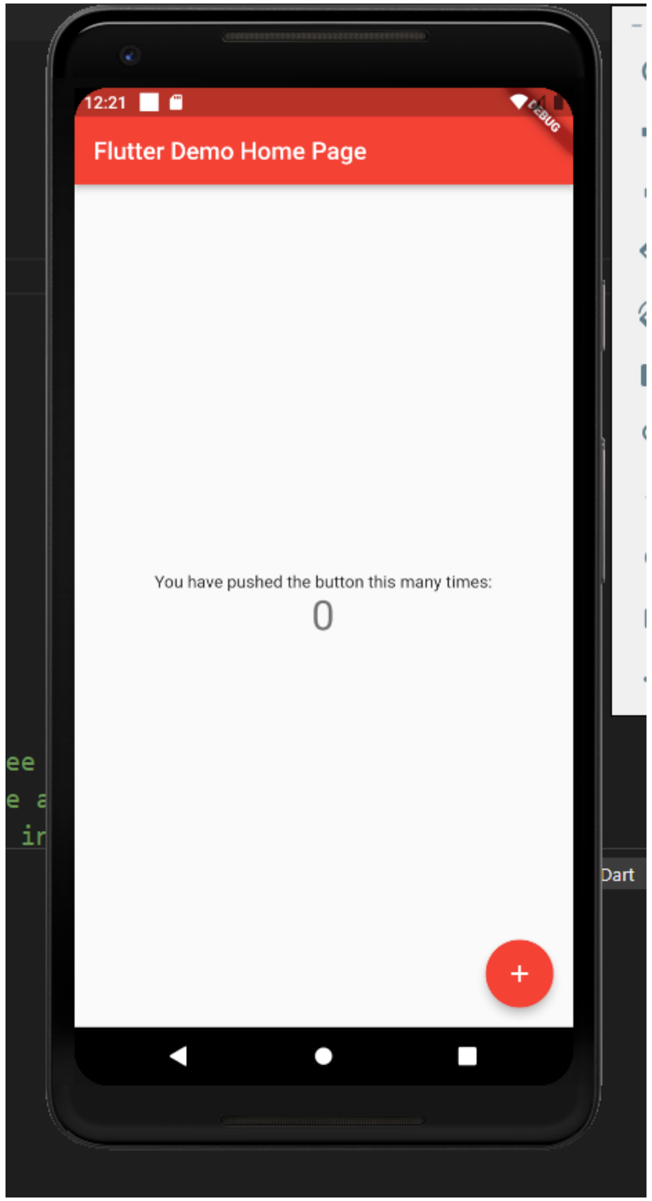
main.dart - flutter_application_1 - Visual Studio Code

```
>
Flutter: Launch Emulator                recently used
Flutter: New Project
C/C++: Edit Configurations (UI)
C/C++: Edit Configurations (JSON)
Icons: Activate VSCode Icons
```

- Choose previously installed Emulator (Android Studio or macOS)

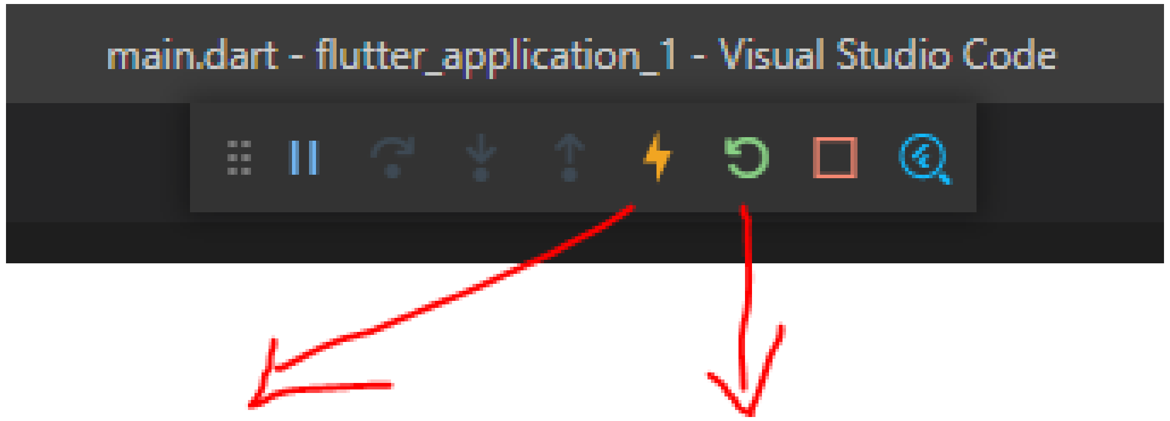


```
Connect a device or select an emulator to launch
Pixel 2 XL API 30 android
Pixel XL API 28 android
flutter emulator android
Pixel 2 XL API 30 android (cold boot)
Pixel XL API 28 android (cold boot)
flutter emulator android (cold boot)
Create Android emulator
```

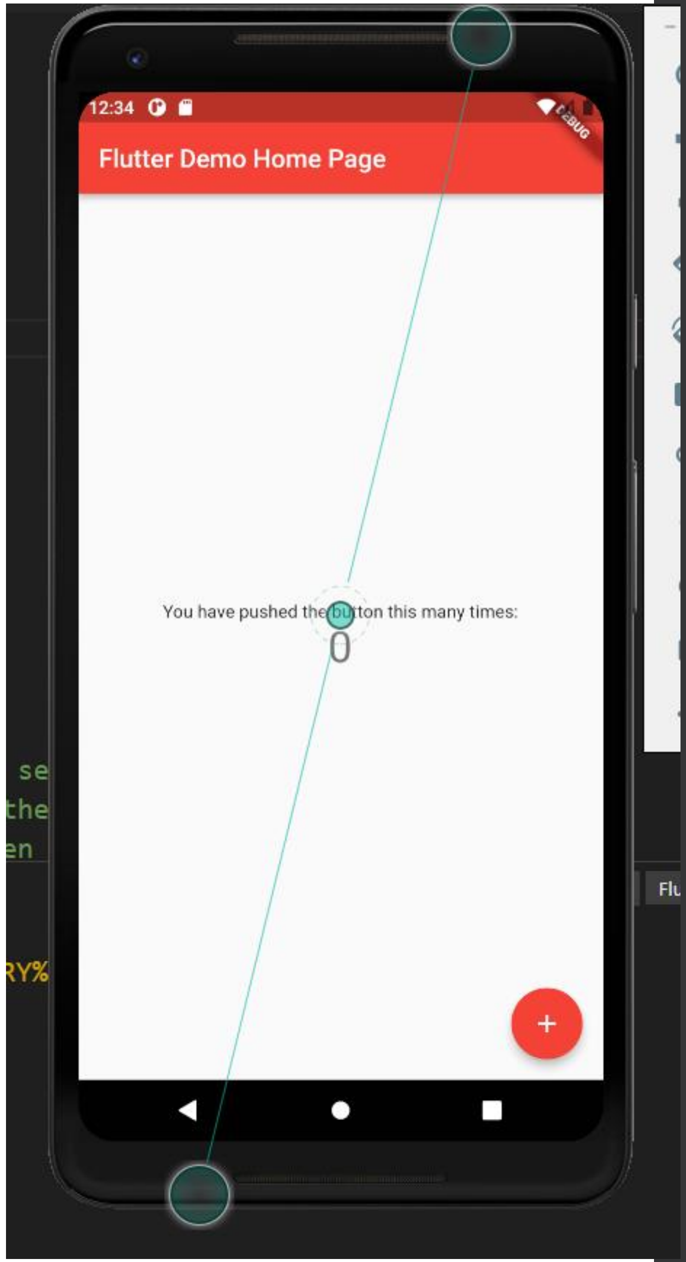


Hot Reload and Hot Restart –VSCode

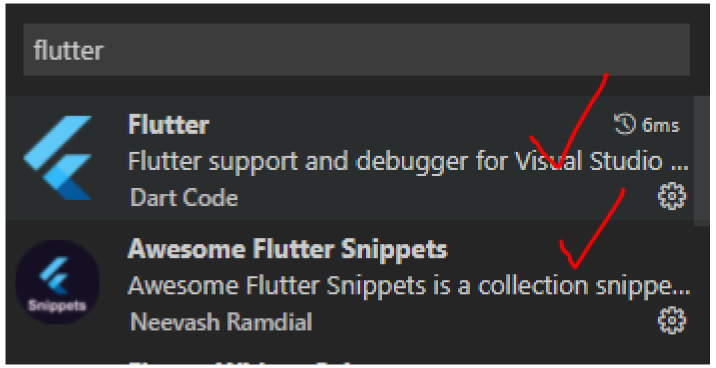
- Press F5 to debug Project



HOT RELOAD HOT RESTART → Directly Applied Changes



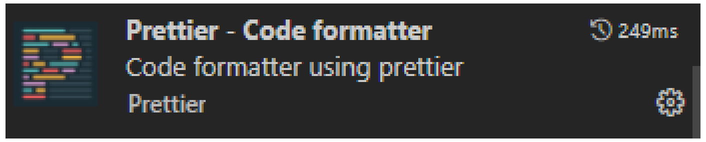
Additional Extension Packages For VSCode



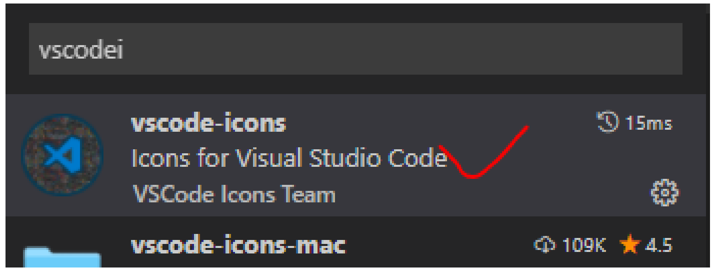
flutter

Flutter 6ms
Flutter support and debugger for Visual Studio ...
Dart Code

Awesome Flutter Snippets
Awesome Flutter Snippets is a collection snippe...
Neevash Ramdial



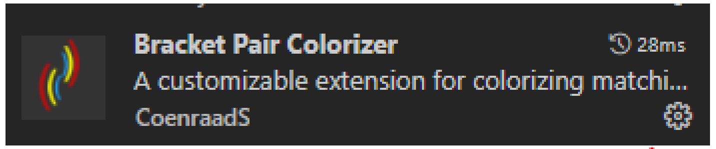
Prettier - Code formatter 249ms
Code formatter using prettier
Prettier



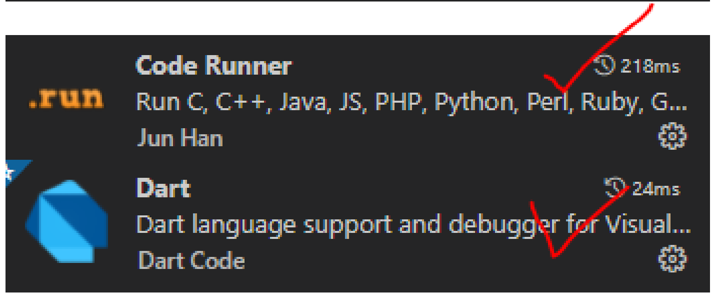
vscodei

vscode-icons 15ms
Icons for Visual Studio Code
VSCode Icons Team

vscode-icons-mac 109K ★ 4.5



Bracket Pair Colorizer 28ms
A customizable extension for colorizing matchi...
CoenraadS

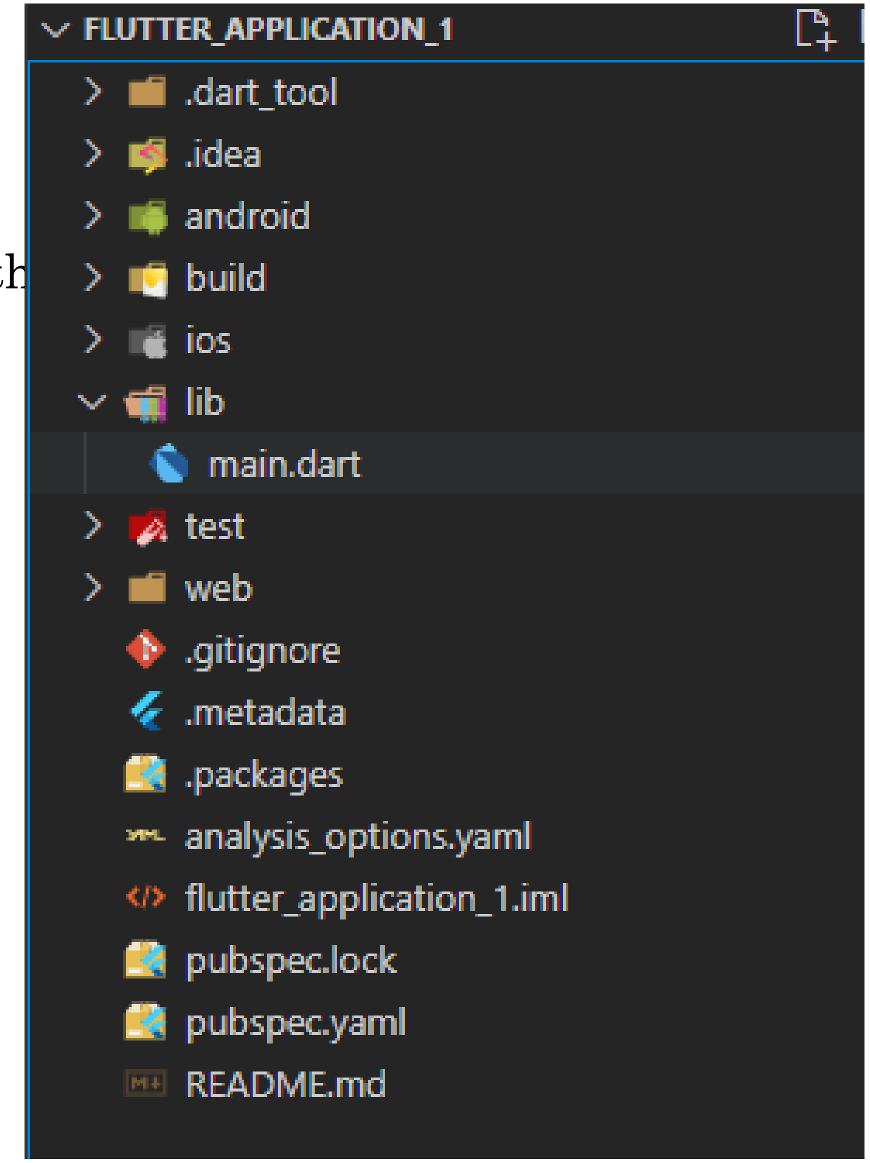


Code Runner 218ms
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, G...
Jun Han

Dart 24ms
Dart language support and debugger for Visual...
Dart Code

Flutter Project Folder Structure

- **Android, ios and web folders:** for different cross platforms
- **Lib folder:** all source code written for Project.
- **Pubspec.yaml file:** All Project references, sources path and other important settings configuration file for the Project.



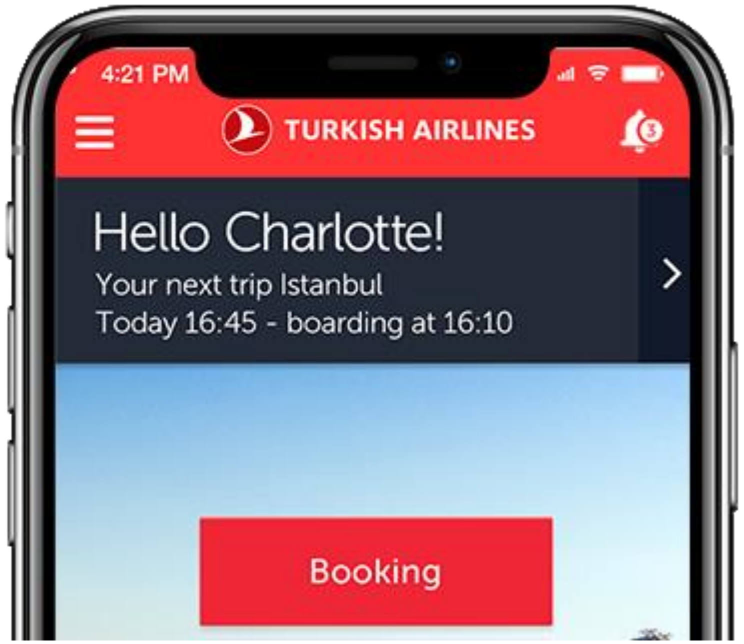
```
FLUTTER_APPLICATION_1
> .dart_tool
> .idea
> android
> build
> ios
v lib
    main.dart
> test
> web
  .gitignore
  .metadata
  .packages
  analysis_options.yaml
  flutter_application_1.iml
  pubspec.lock
  pubspec.yaml
  README.md
```

Before Starting: What is a Flutter Widget?

- https://flutter.dev/docs/development/ui/widgets
- Everthing is widget in flutter mobile platform, developed by Dart Language
  - Text boxes
  - Labels,
  - Buttons,
  - Containers
  - AppBars
  - Sliders
  - Scaffold containers
  - Columns,
  - Rows,
  - Check boxes
  - ..
  - ..
  - ..
  - And many

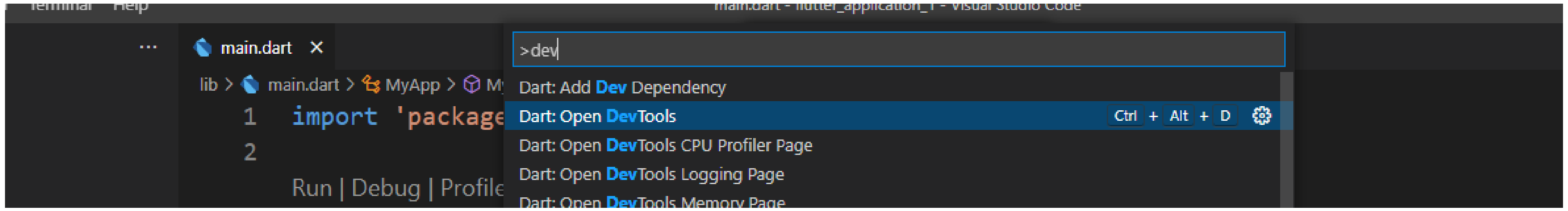
What is Flutter Mobile App Developmet

- Designing all required widgets under logical hierarchy and communicating these widgets together by using Dart language.



Widget Hierarchy- Widget Inspector

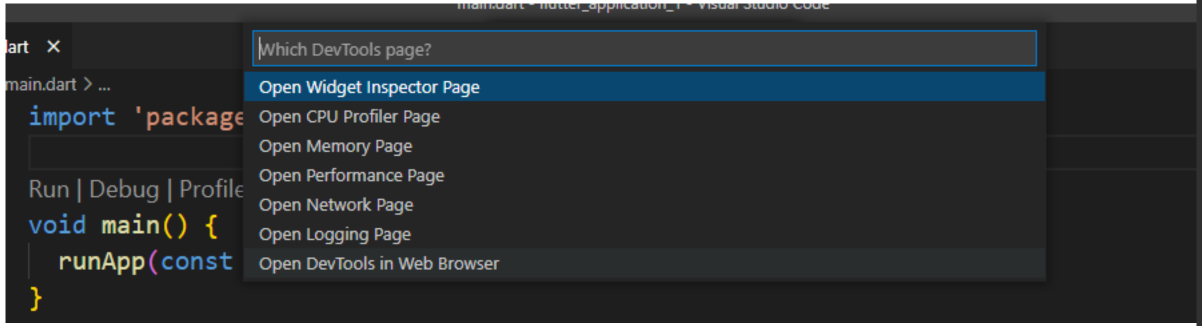
- A tool for examining and checking widget location hierarcyh.



```
>dev
Dart: Add Dev Dependency
Dart: Open DevTools                      Ctrl + Alt + D
Dart: Open DevTools CPU Profiler Page
Dart: Open DevTools Logging Page
Dart: Open DevTools Memory Page
```

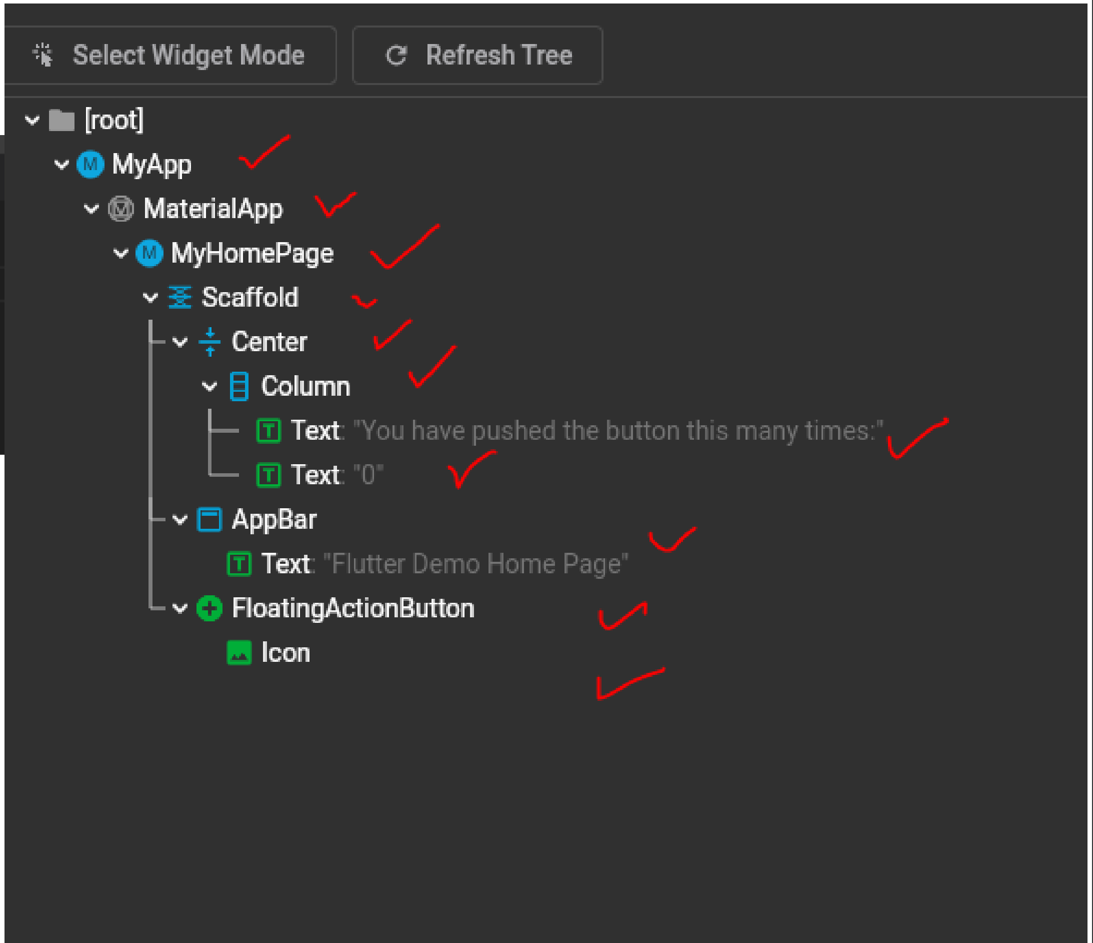
```
import 'package
Run | Debug | Profile
```

- Choose Open DevTools in Web Browser



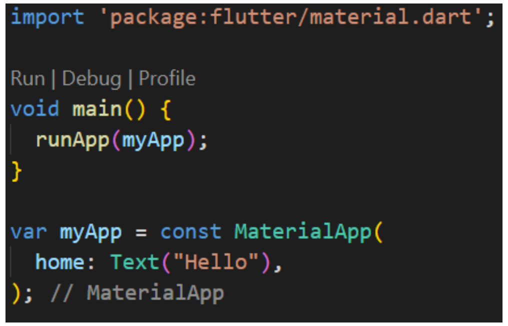
```
Which DevTools page?
Open Widget Inspector Page
Open CPU Profiler Page
Open Memory Page
Open Performance Page
Open Network Page
Open Logging Page
Open DevTools in Web Browser
```

```
import 'package
Run | Debug | Profile
void main() {
  runApp(const
}
```

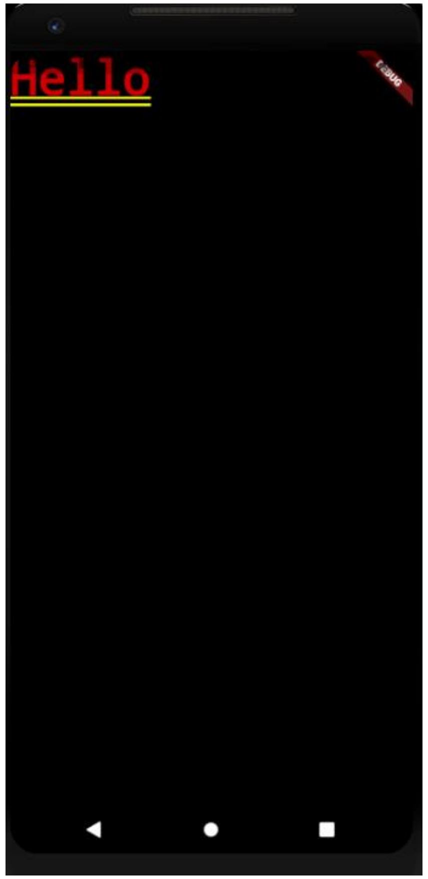


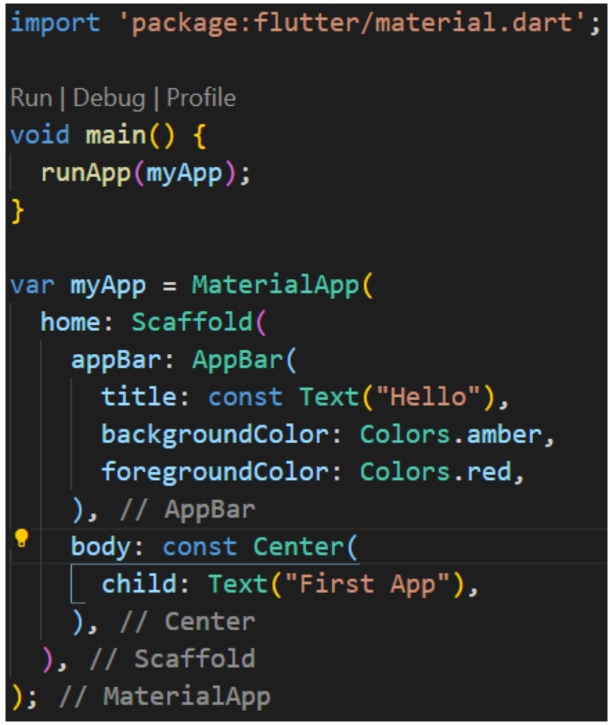
```
Select Widget Mode    Refresh Tree
[root]
  MyApp ✓
    MaterialApp ✓
      MyHomePage ✓
        Scaffold ✓
          Center ✓
            Column ✓
              Text: "You have pushed the button this many times:" ✓
              Text: "0" ✓
          AppBar ✓
            Text: "Flutter Demo Home Page"
          FloatingActionButton ✓
            Icon ✓
```

First Project, MaterialApp, AppBar, Scaffod

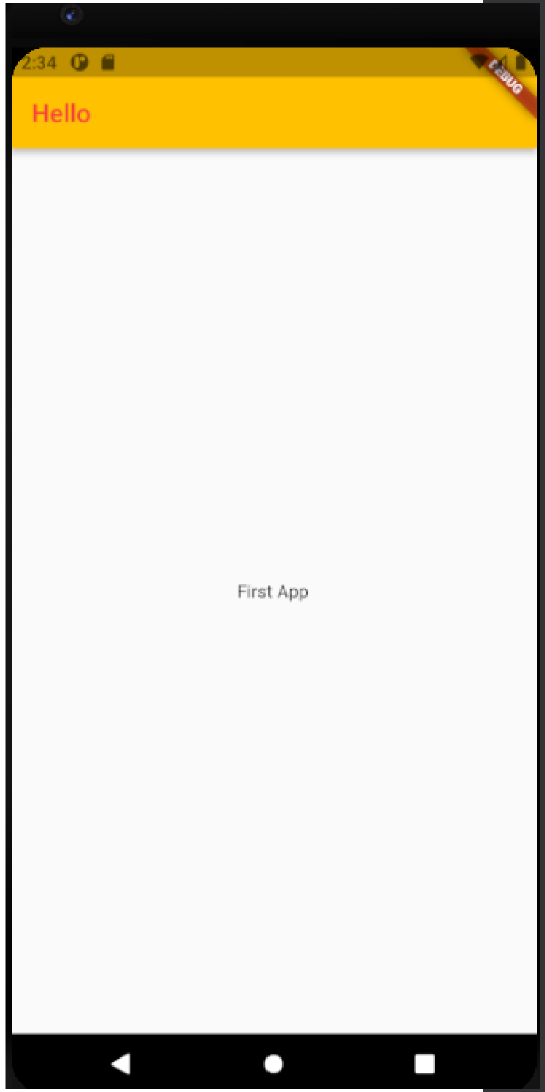


```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(myApp);
}

var myApp = const MaterialApp(
  home: Text("Hello"),
); // MaterialApp
```



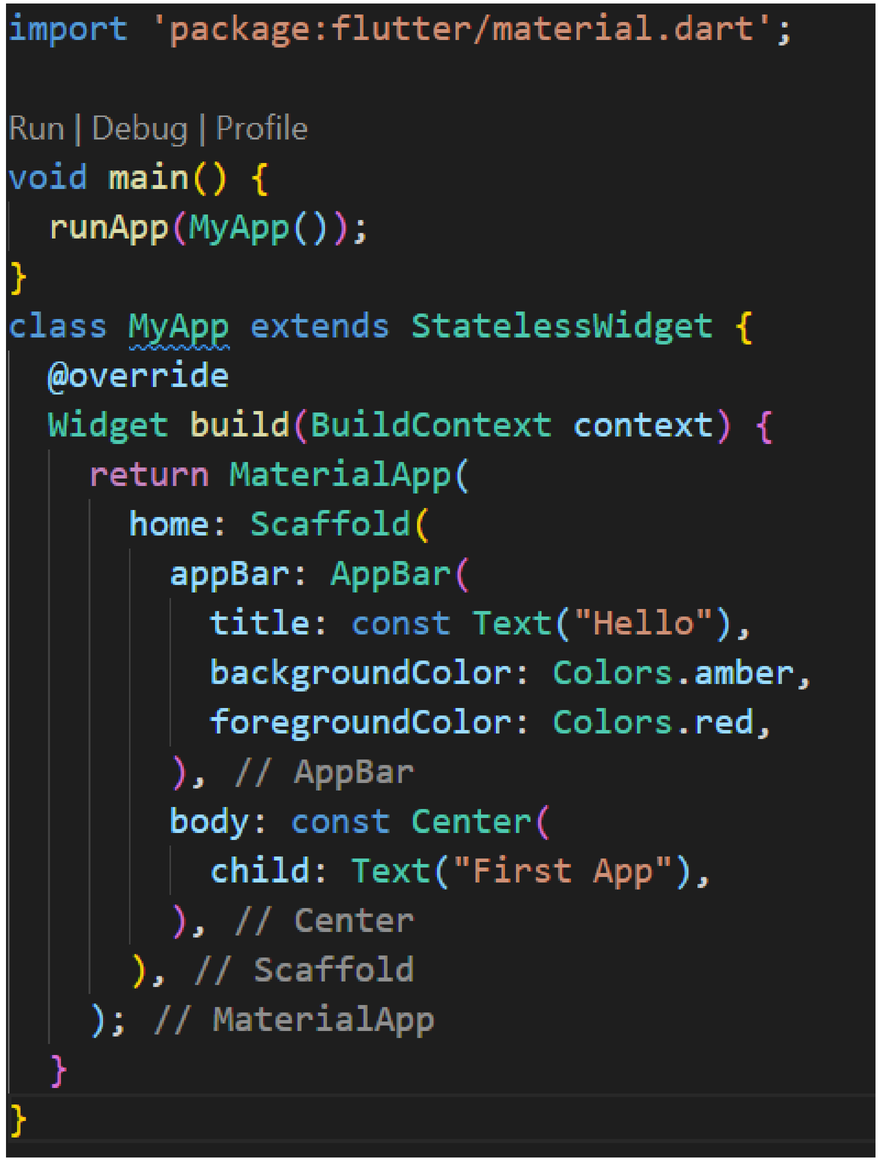


```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(myApp);
}

var myApp = MaterialApp(
  home: Scaffold(
    appBar: AppBar(
      title: const Text("Hello"),
      backgroundColor: Colors.amber,
      foregroundColor: Colors.red,
    ), // AppBar
    body: const Center(
      child: Text("First App"),
    ), // Center
  ), // Scaffold
); // MaterialApp
```

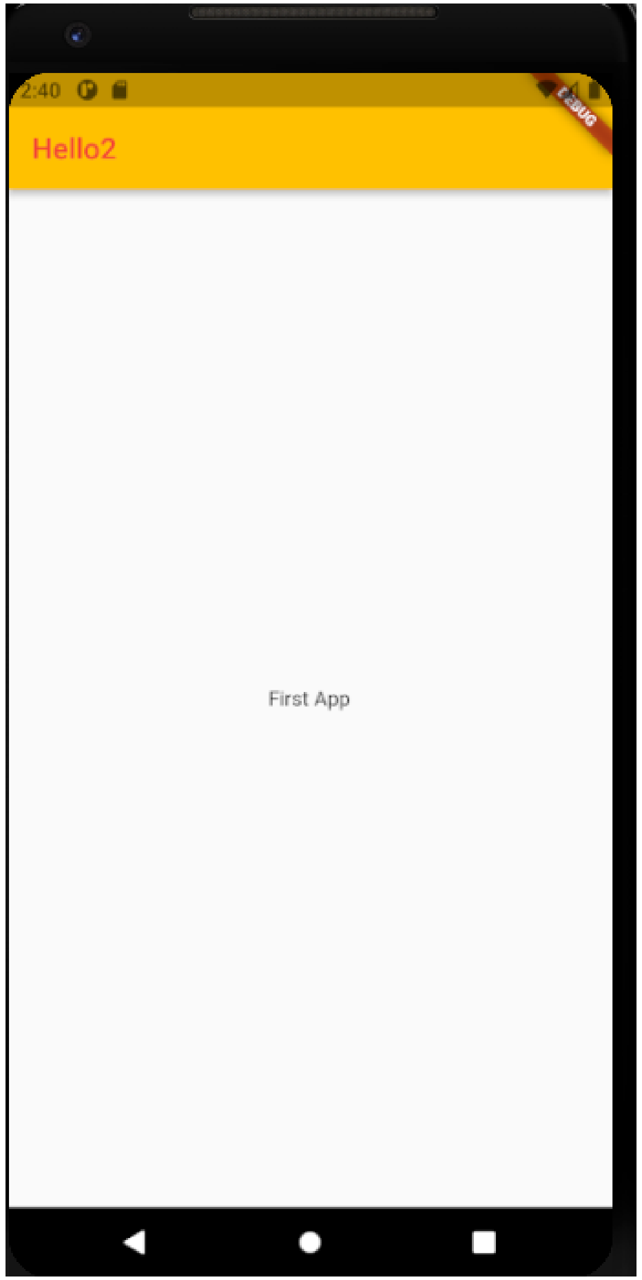


First Project with our custom widget class

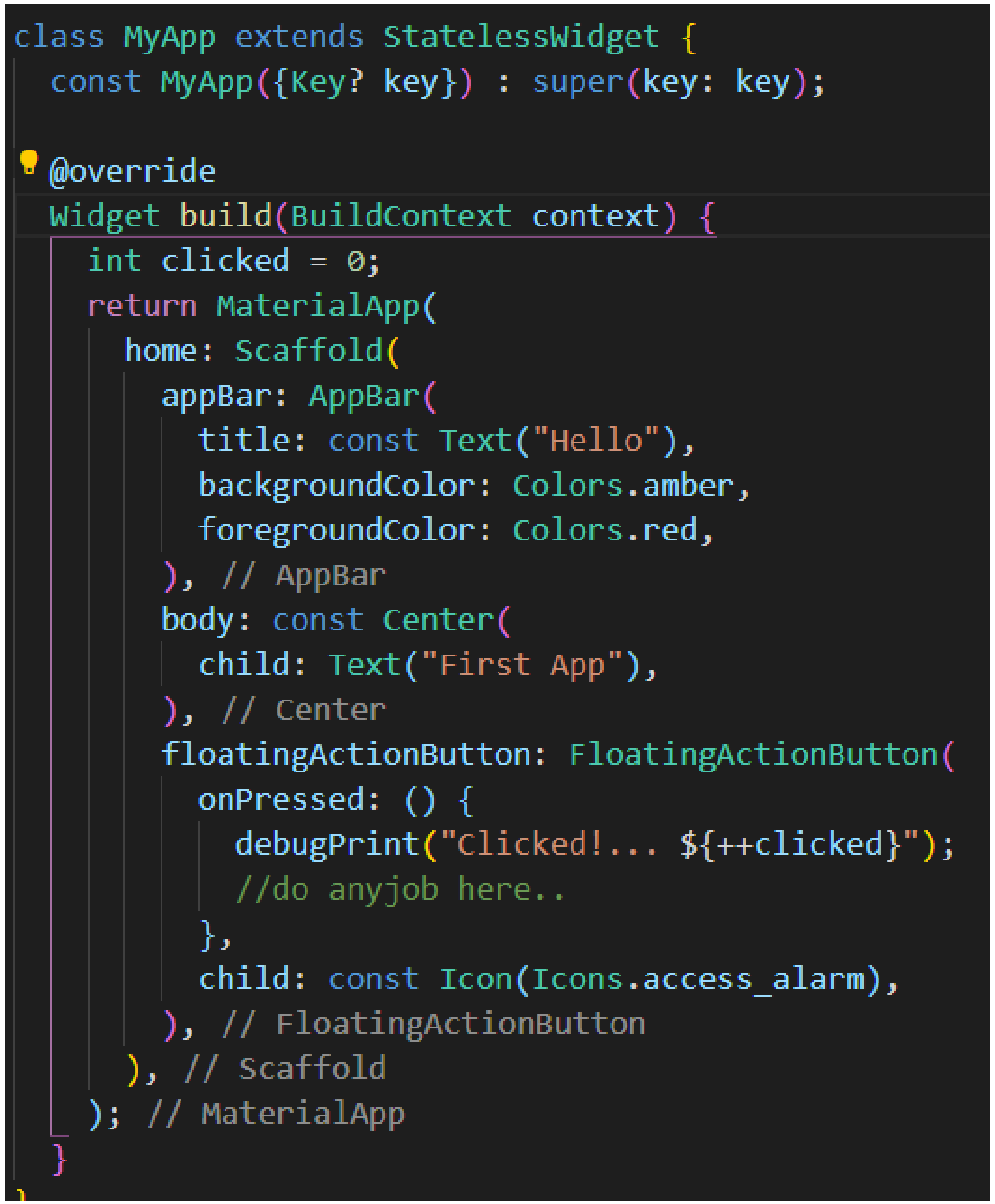
- Hot Reload Works with custom defined widgets.



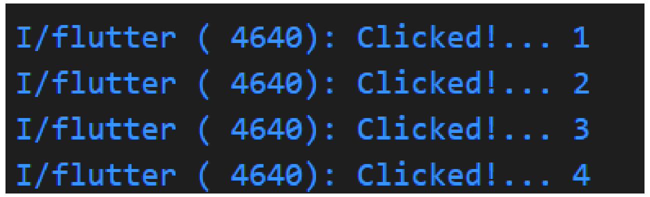
```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text("Hello"),
          backgroundColor: Colors.amber,
          foregroundColor: Colors.red,
        ), // AppBar
        body: const Center(
          child: Text("First App"),
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```



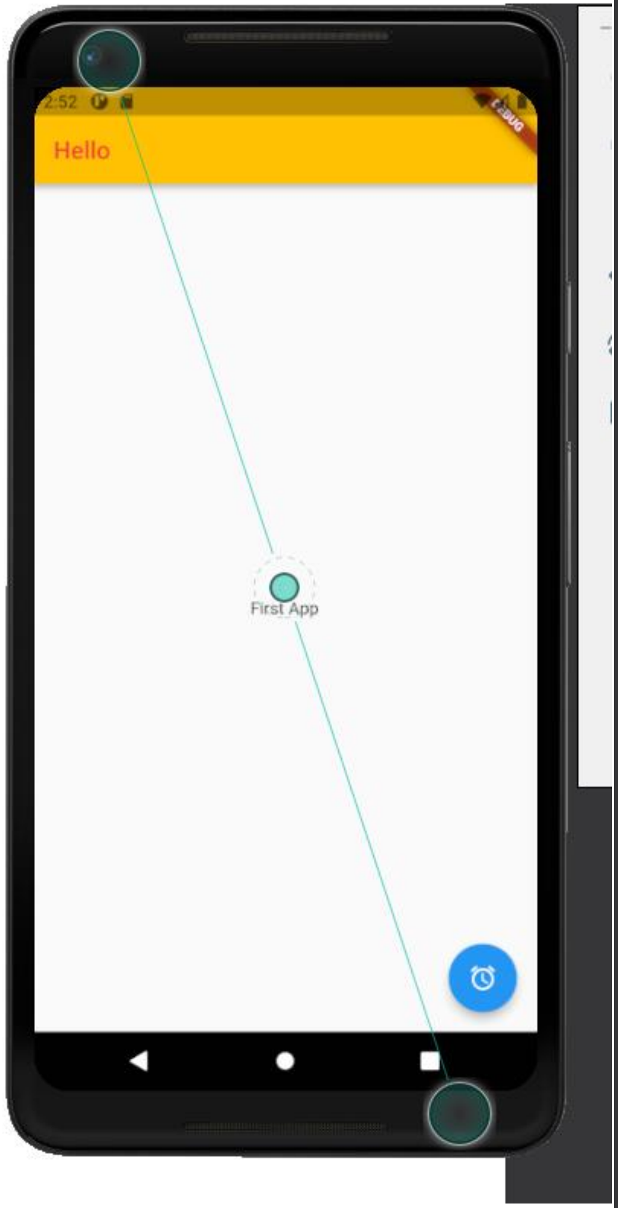
Floating Action Button Widged



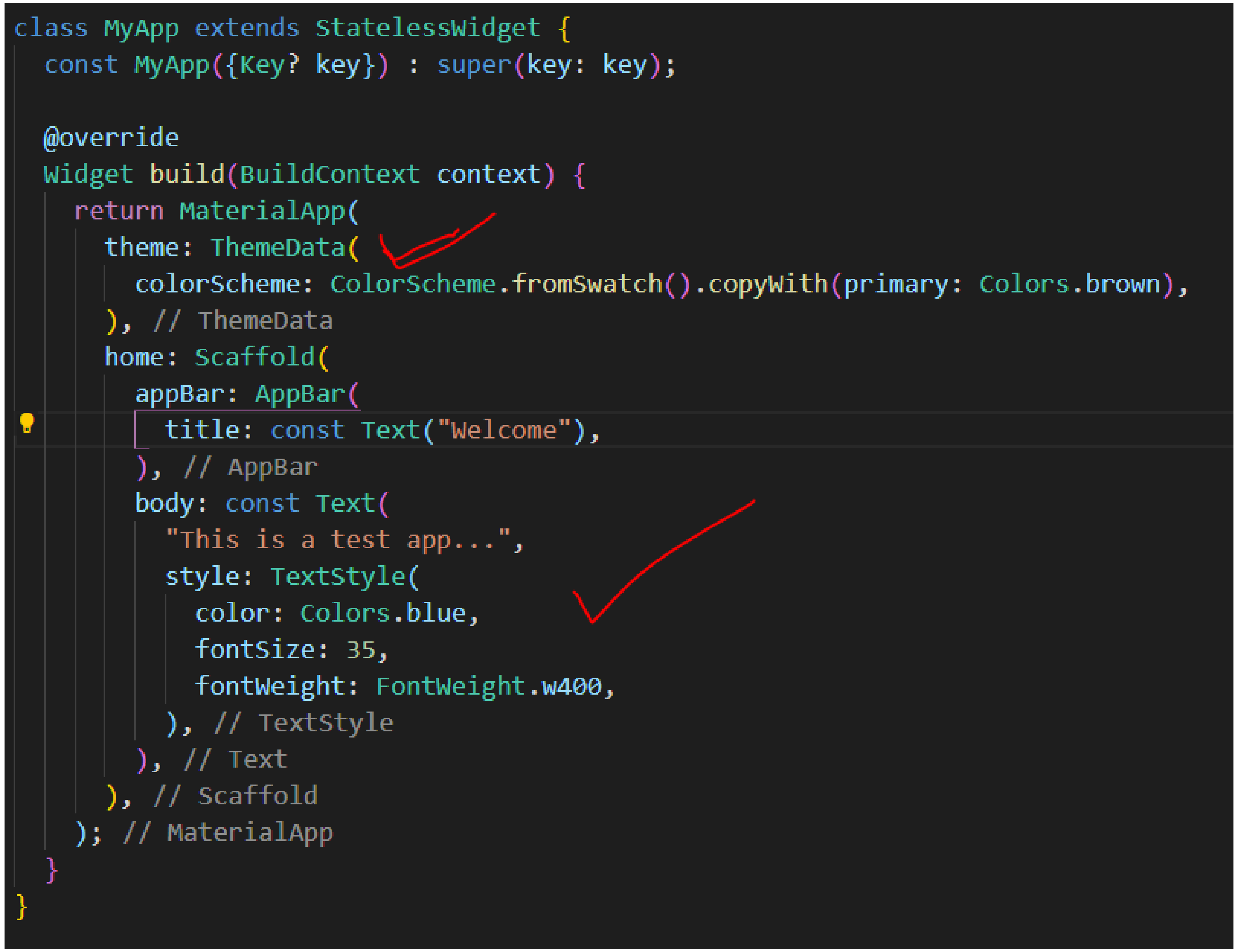
```dart
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    int clicked = 0;
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text("Hello"),
          backgroundColor: Colors.amber,
          foregroundColor: Colors.red,
        ), // AppBar
        body: const Center(
          child: Text("First App"),
        ), // Center
        floatingActionButton: FloatingActionButton(
          onPressed: () {
            debugPrint("Clicked!... ${++clicked}");
            //do anyjob here..
          },
          child: const Icon(Icons.access_alarm),
        ), // FloatingActionButton
      ), // Scaffold
    ); // MaterialApp
  }
```



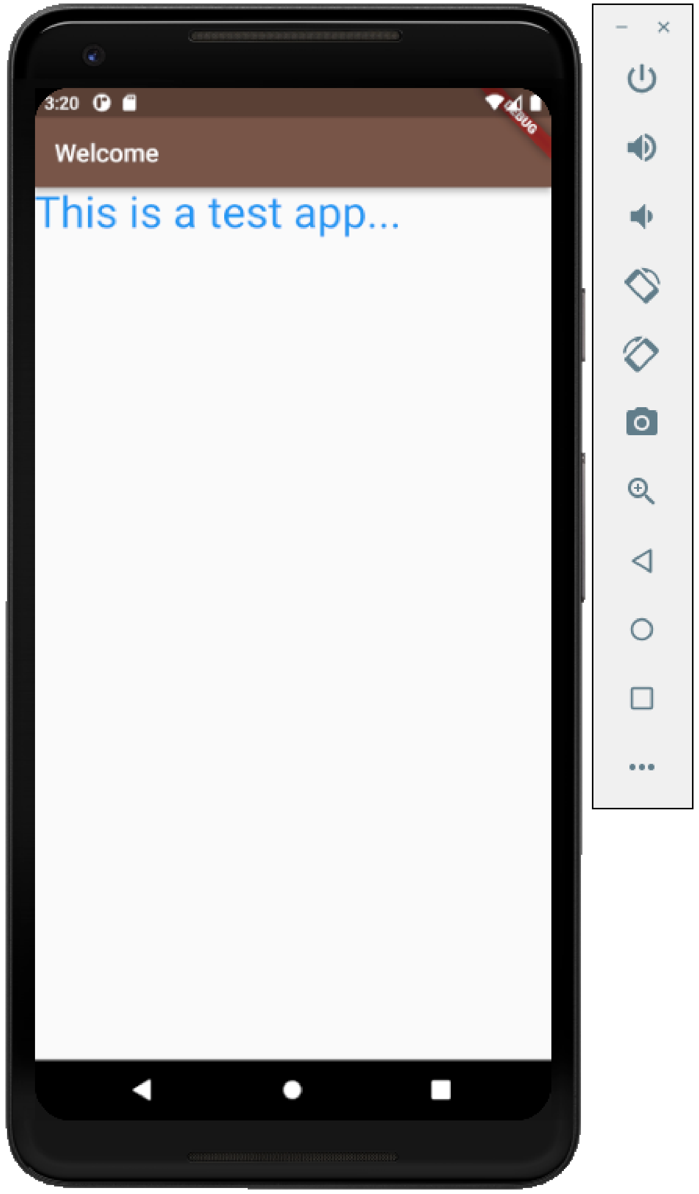
```
I/flutter ( 4640): Clicked!... 1
I/flutter ( 4640): Clicked!... 2
I/flutter ( 4640): Clicked!... 3
I/flutter ( 4640): Clicked!... 4
```



Theme and Styling

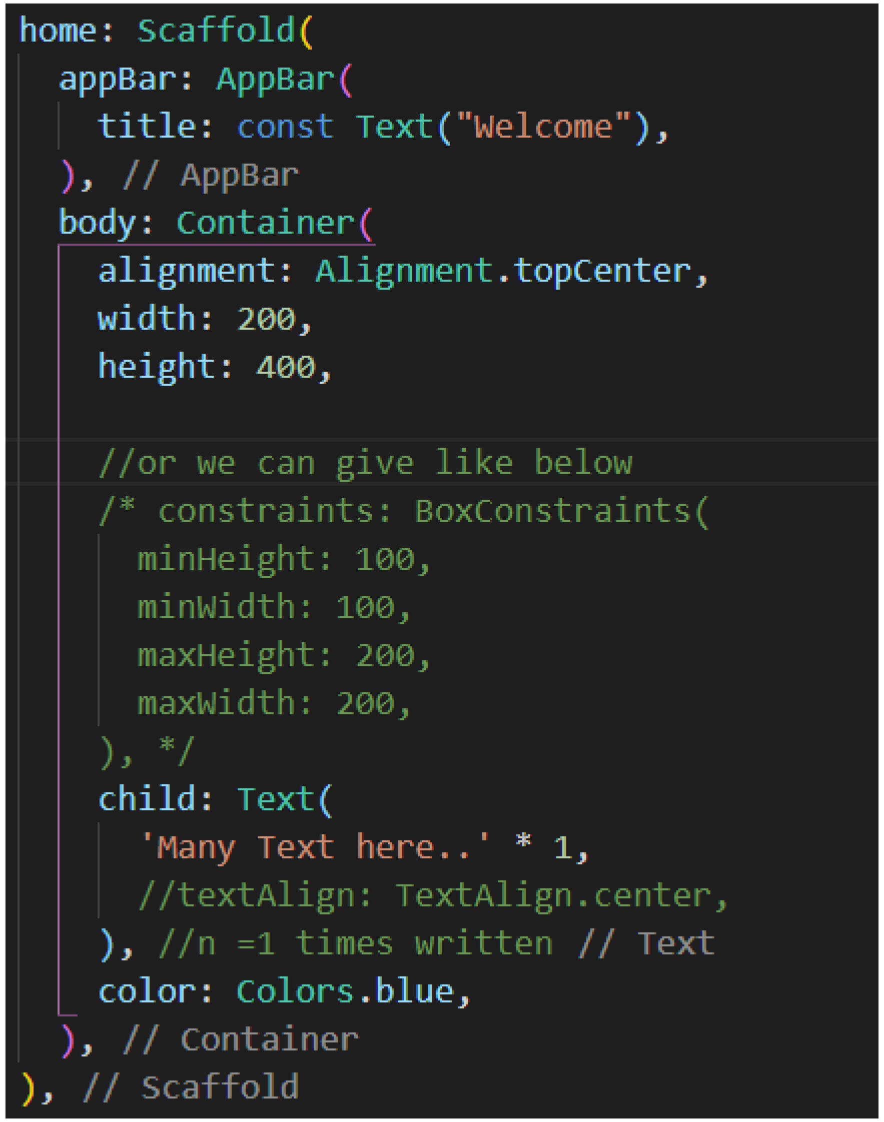


```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        colorScheme: ColorScheme.fromSwatch().copyWith(primary: Colors.brown),
      ), // ThemeData
      home: Scaffold(
        appBar: AppBar(
          title: const Text("Welcome"),
        ), // AppBar
        body: const Text(
          "This is a test app...",
          style: TextStyle(
            color: Colors.blue,
            fontSize: 35,
            fontWeight: FontWeight.w400,
          ), // TextStyle
        ), // Text
      ), // Scaffold
    ); // MaterialApp
  }
}
```

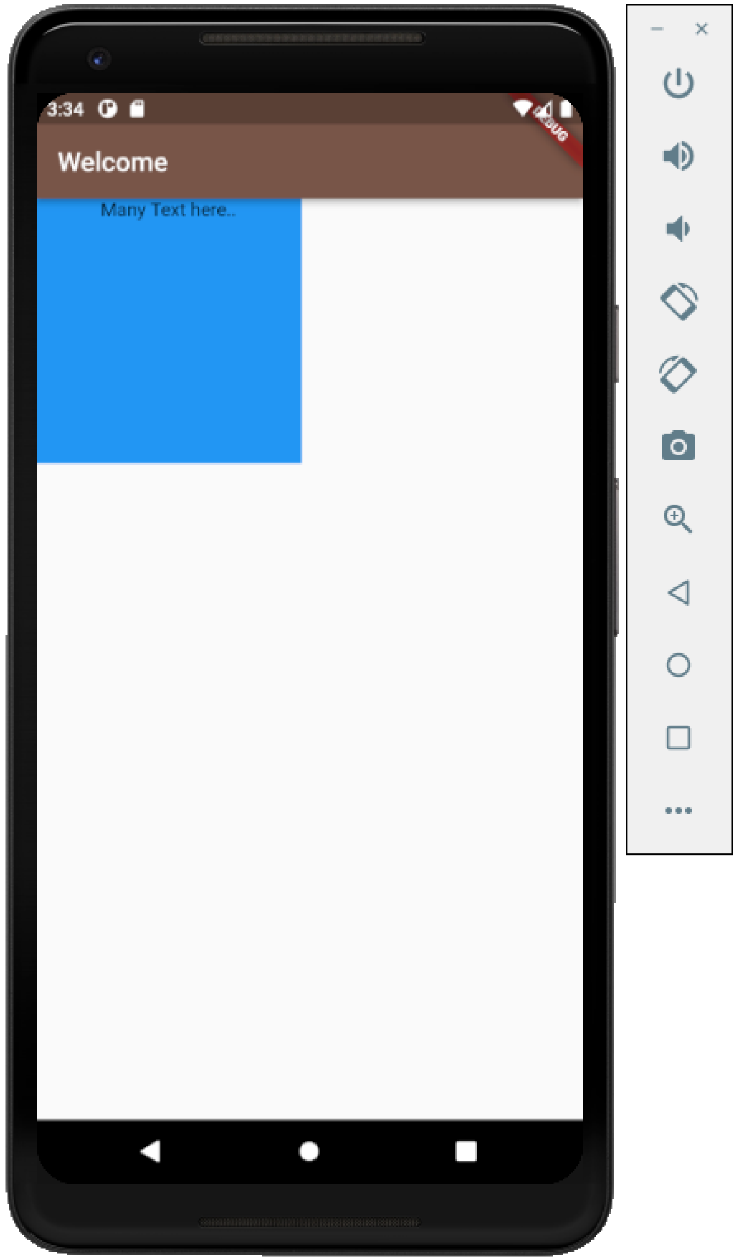


Container Widgets, Constraints, positions

- ..

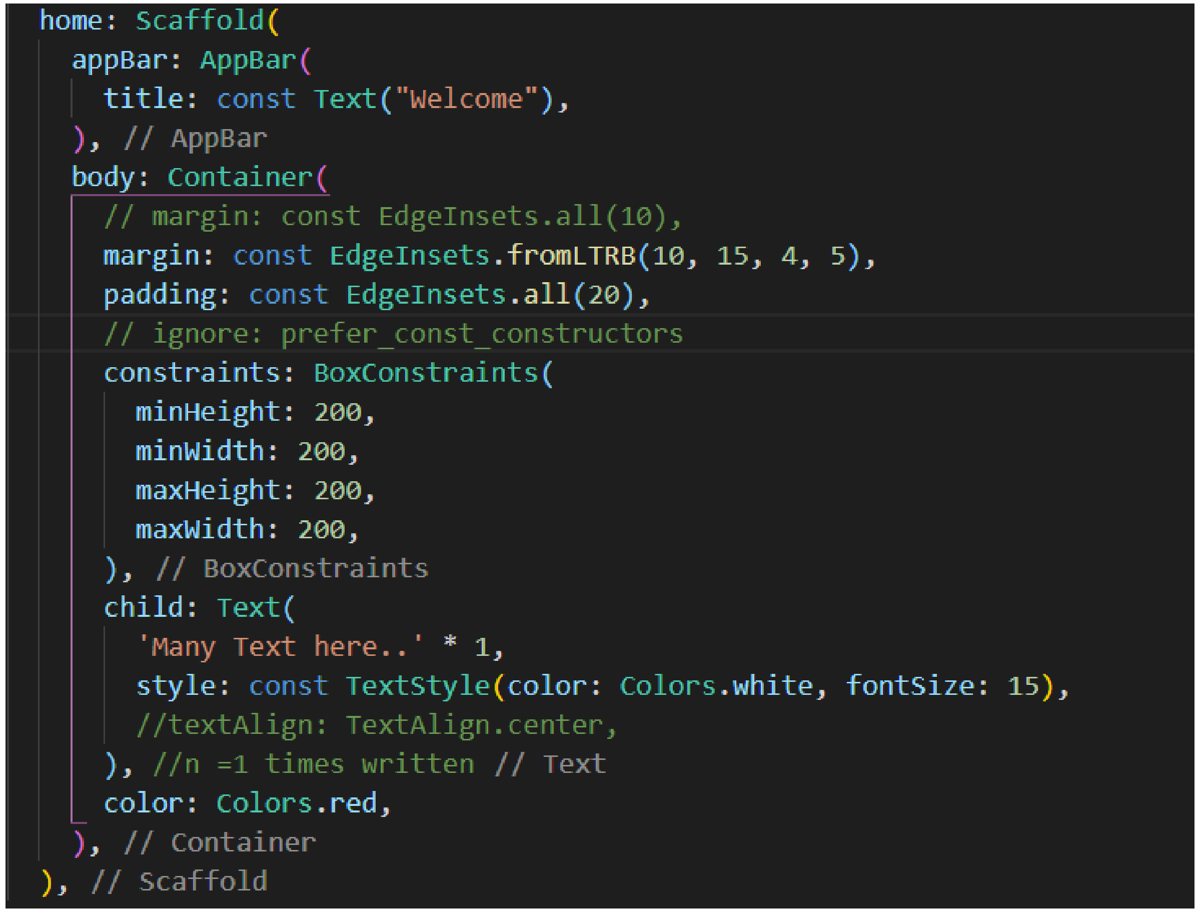


```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  body: Container(
    alignment: Alignment.topCenter,
    width: 200,
    height: 400,

    //or we can give like below
    /* constraints: BoxConstraints(
      minHeight: 100,
      minWidth: 100,
      maxHeight: 200,
      maxWidth: 200,
    ), */
    child: Text(
      'Many Text here..' * 1,
      //textAlign: TextAlign.center,
    ), //n =1 times written // Text
    color: Colors.blue,
  ), // Container
), // Scaffold
```

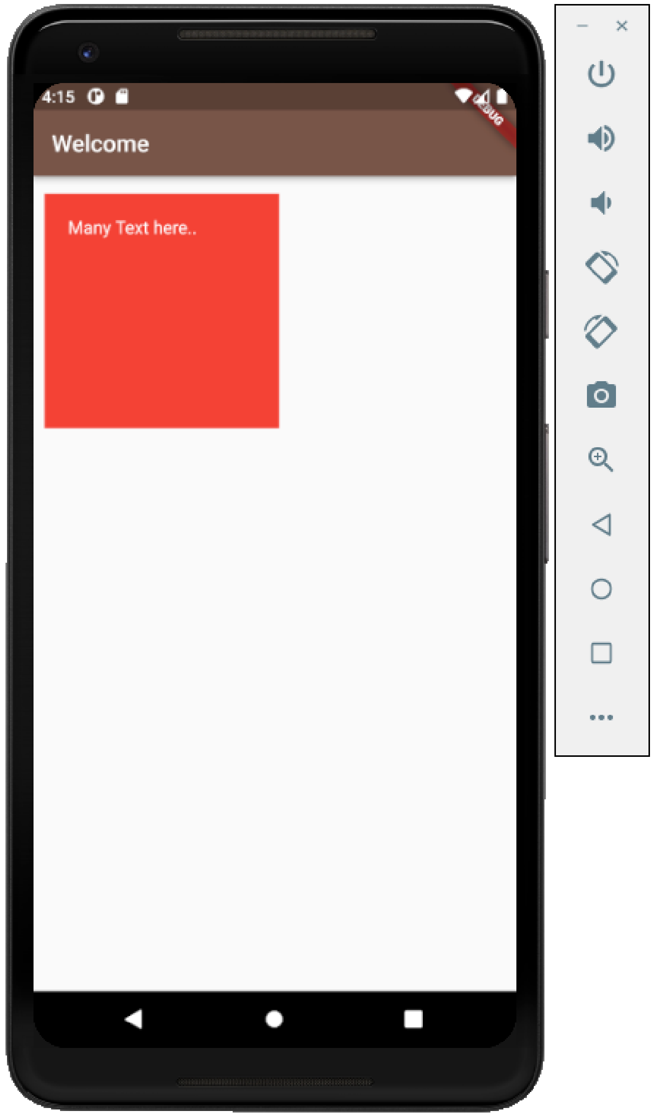


Container Alignment, Margin and Padding

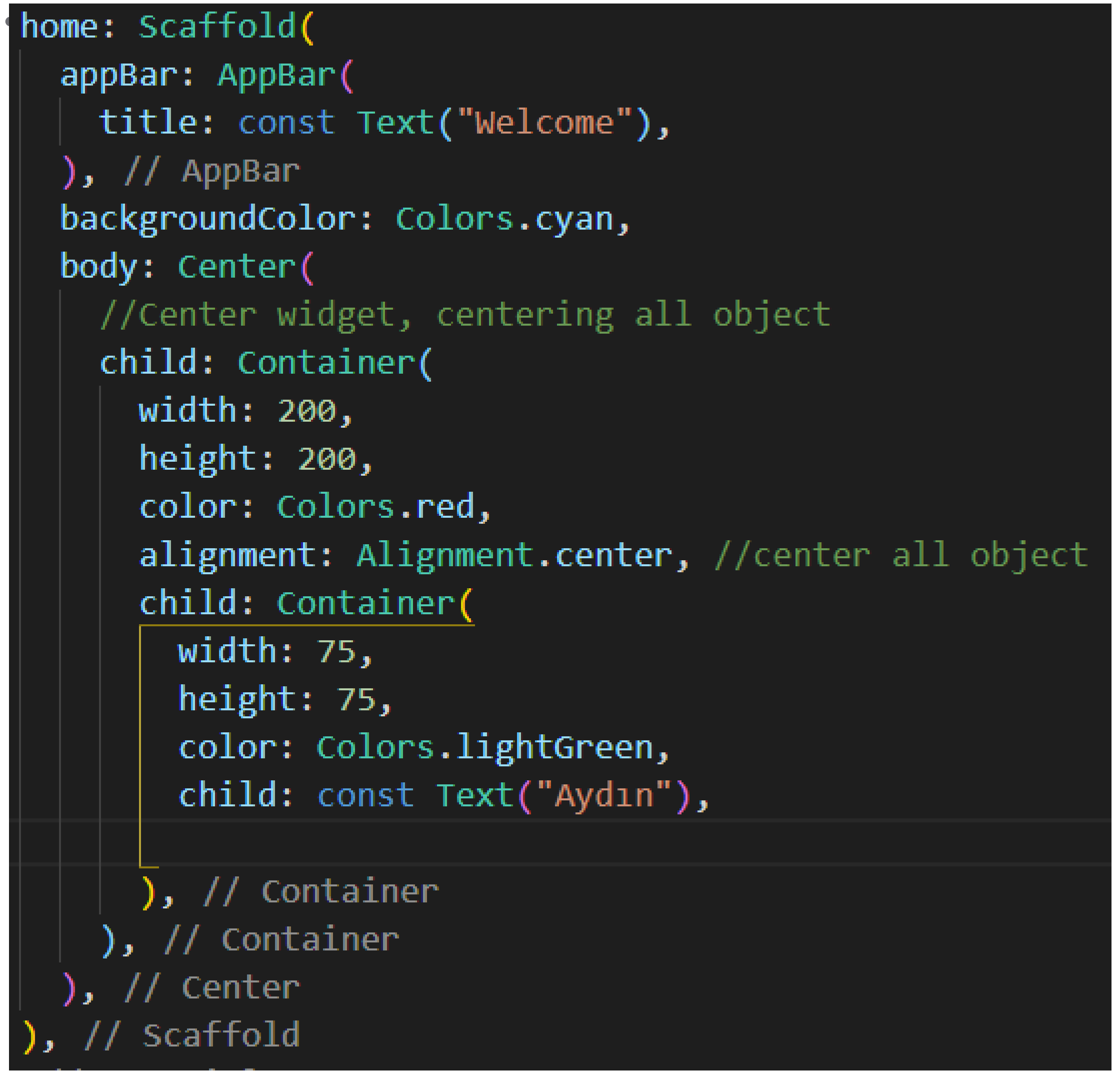
- ..



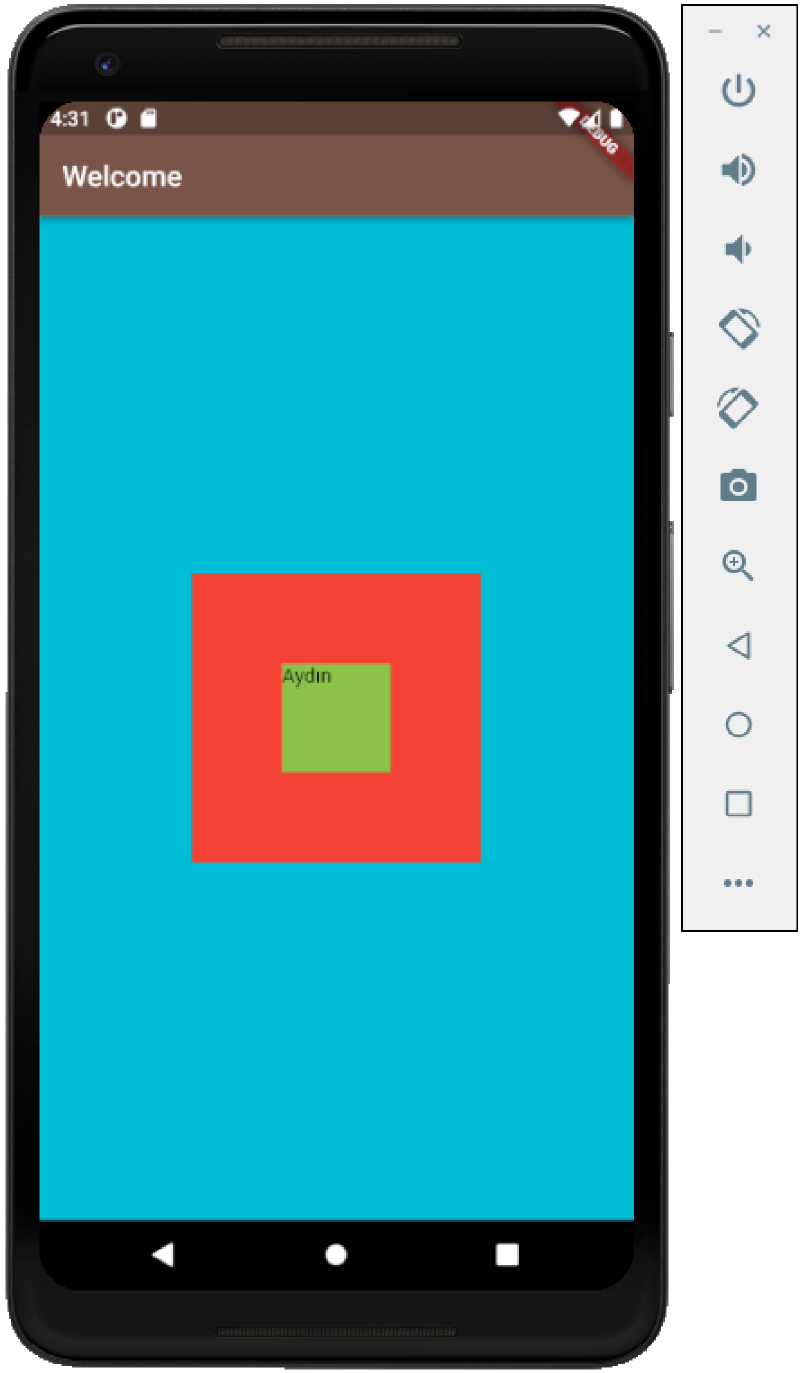
```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  body: Container(
    // margin: const EdgeInsets.all(10),
    margin: const EdgeInsets.fromLTRB(10, 15, 4, 5),
    padding: const EdgeInsets.all(20),
    // ignore: prefer_const_constructors
    constraints: BoxConstraints(
      minHeight: 200,
      minWidth: 200,
      maxHeight: 200,
      maxWidth: 200,
    ), // BoxConstraints
    child: Text(
      'Many Text here..' * 1,
      style: const TextStyle(color: Colors.white, fontSize: 15),
      //textAlign: TextAlign.center,
    ), //n =1 times written // Text
    color: Colors.red,
  ), // Container
), // Scaffold
```



Container Positioning

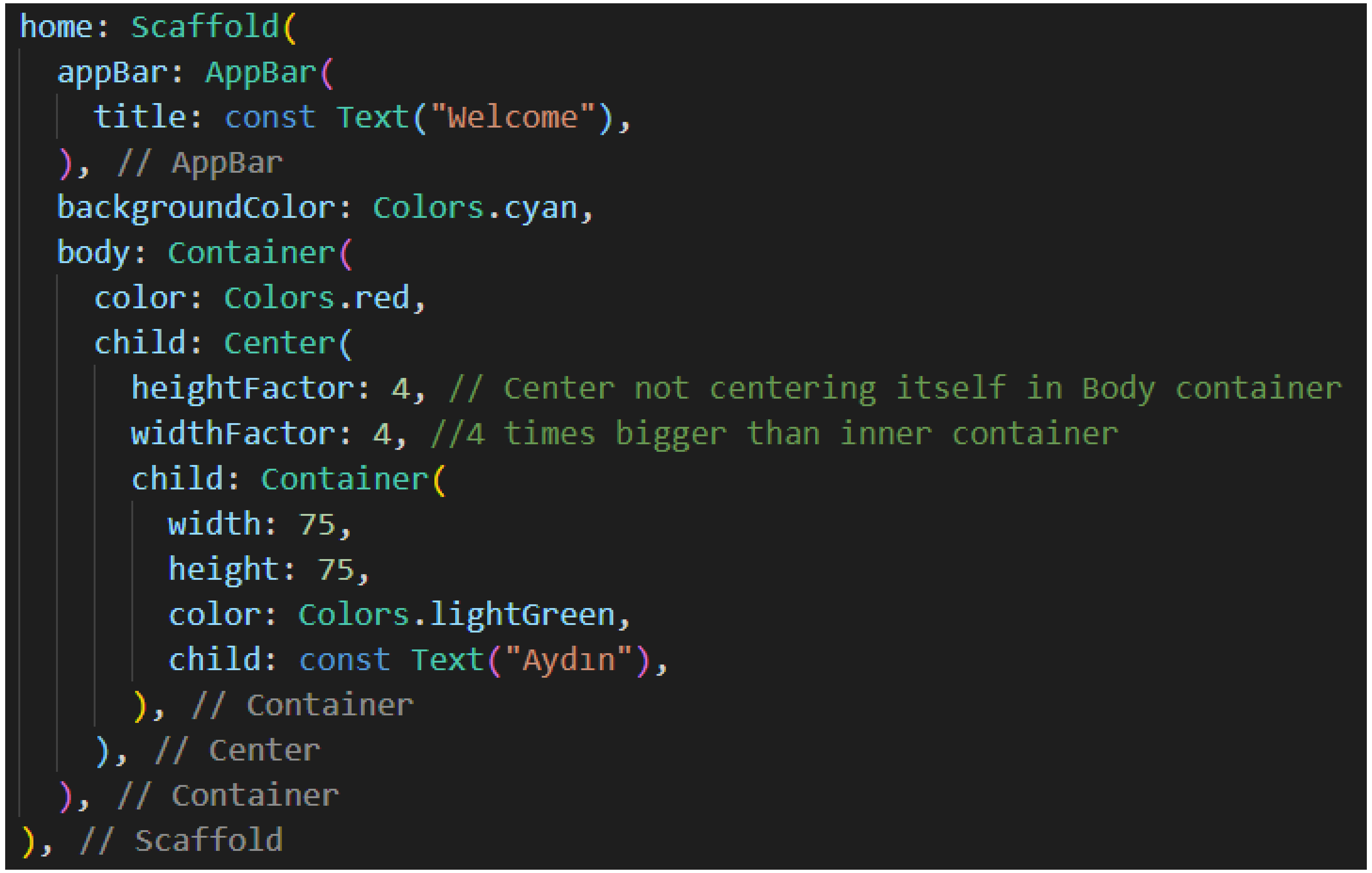


```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  backgroundColor: Colors.cyan,
  body: Center(
    //Center widget, centering all object
    child: Container(
      width: 200,
      height: 200,
      color: Colors.red,
      alignment: Alignment.center, //center all object
      child: Container(
        width: 75,
        height: 75,
        color: Colors.lightGreen,
        child: const Text("Aydın"),

      ), // Container
    ), // Container
  ), // Center
), // Scaffold
```

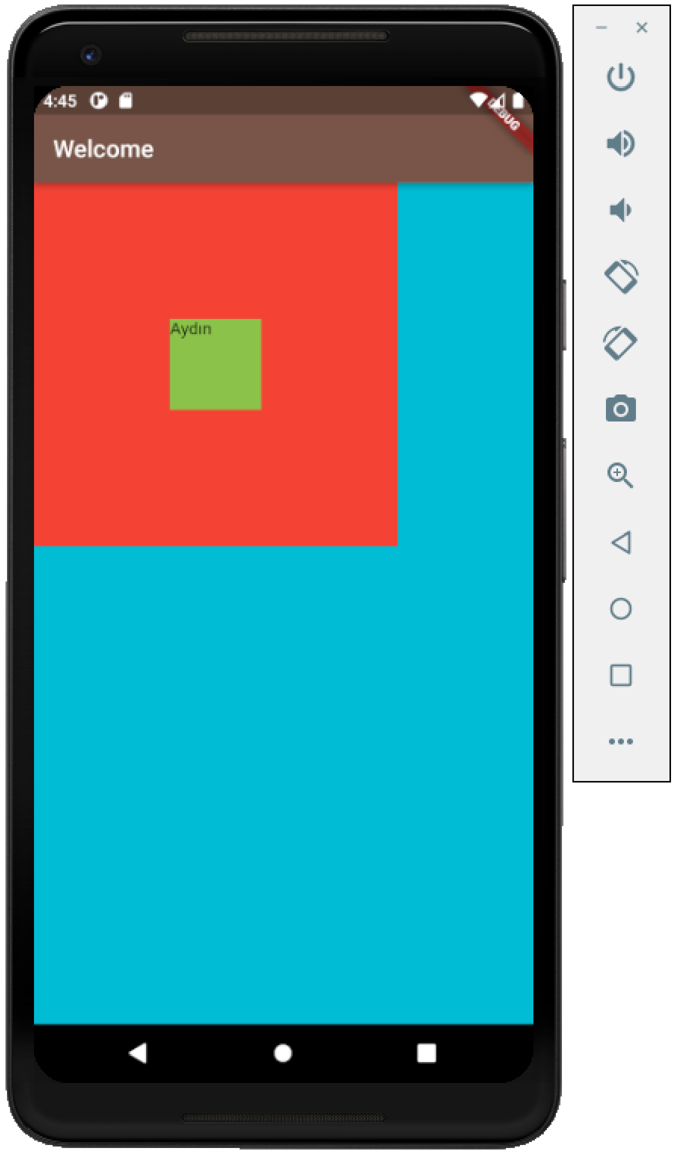


Widht and heigh factor, for free center object.

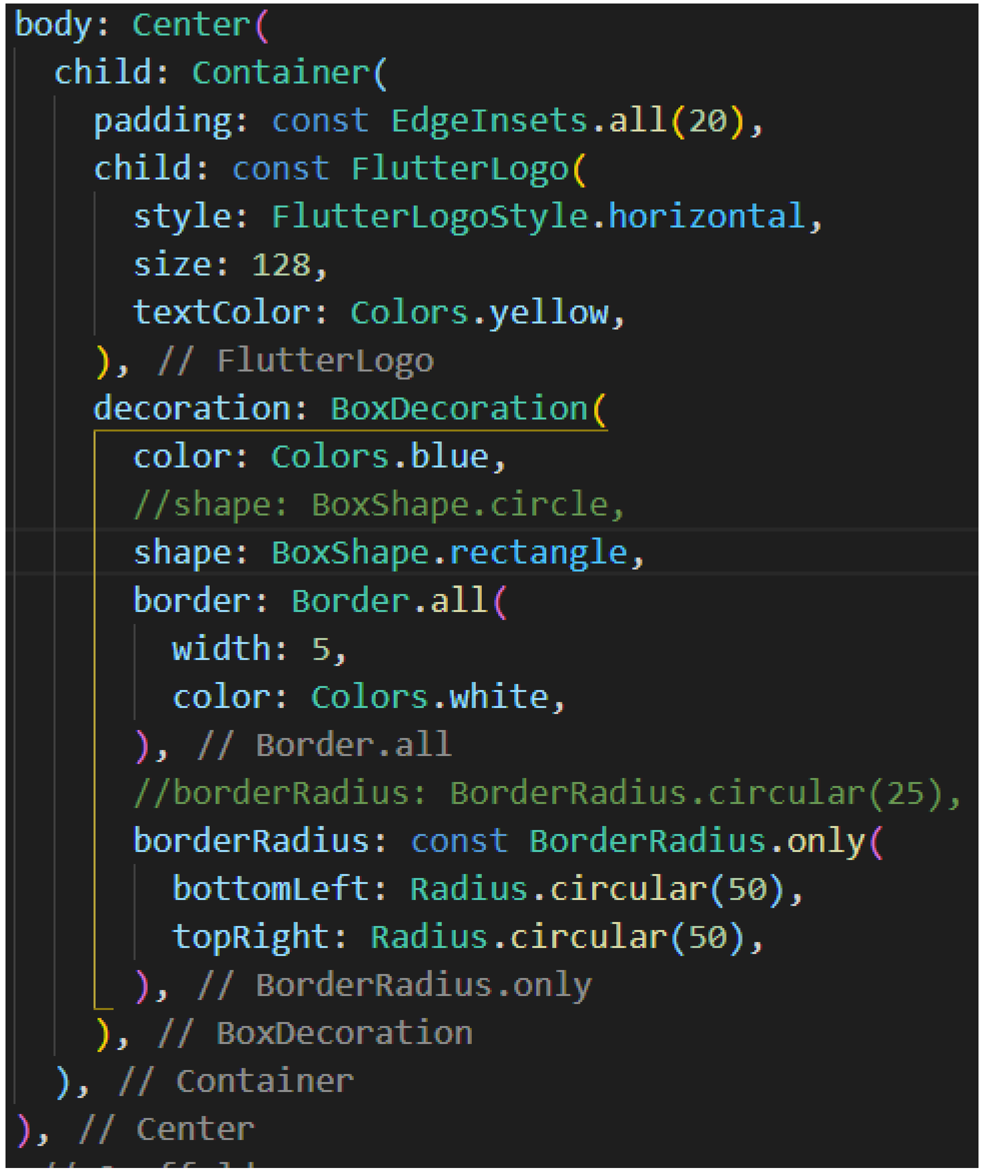
- ..



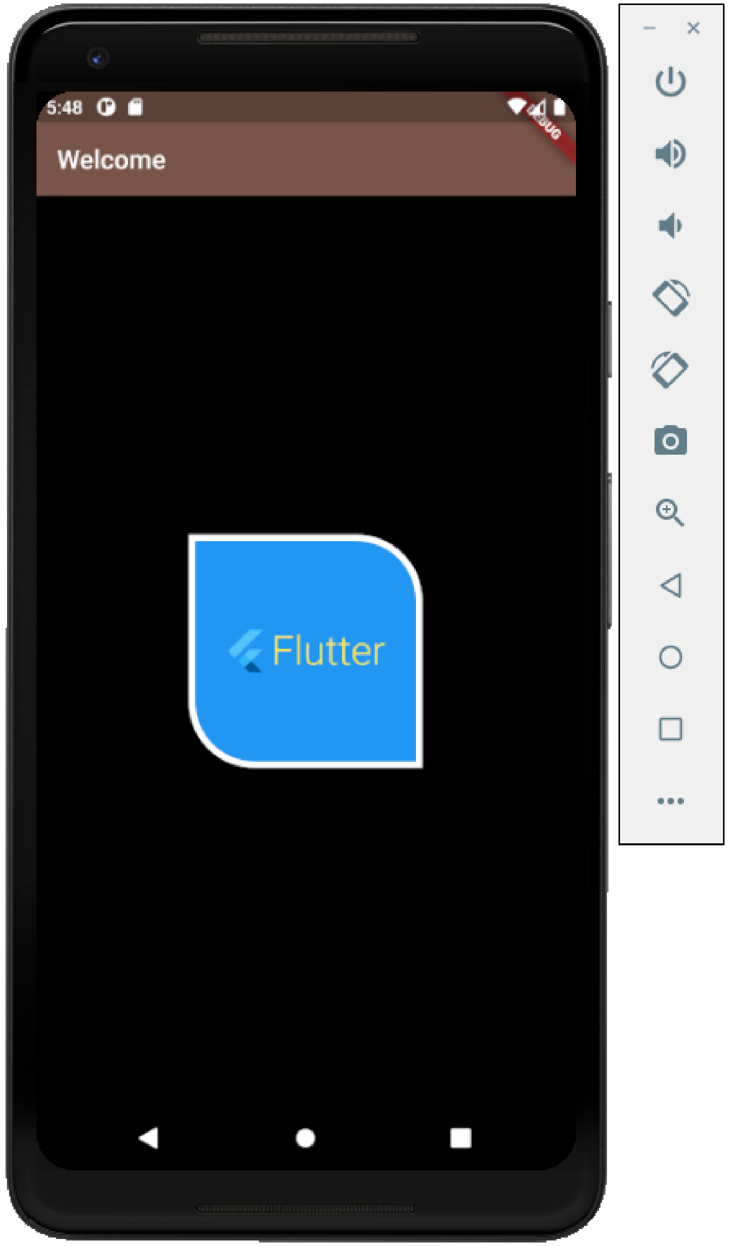
```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  backgroundColor: Colors.cyan,
  body: Container(
    color: Colors.red,
    child: Center(
      heightFactor: 4, // Center not centering itself in Body container
      widthFactor: 4, //4 times bigger than inner container
      child: Container(
        width: 75,
        height: 75,
        color: Colors.lightGreen,
        child: const Text("Aydın"),
      ), // Container
    ), // Center
  ), // Container
), // Scaffold
```



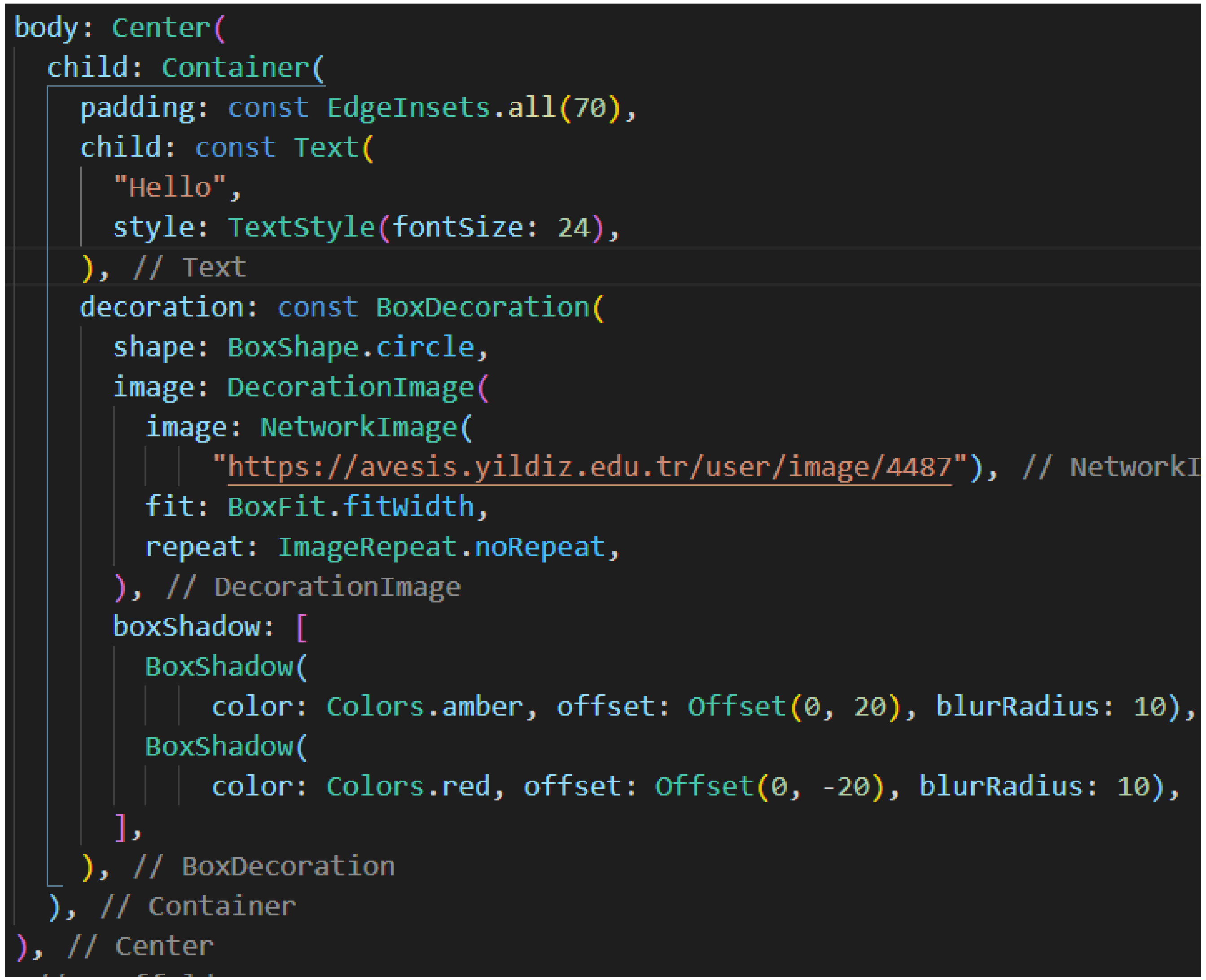
Container BoxDecoration, Borders



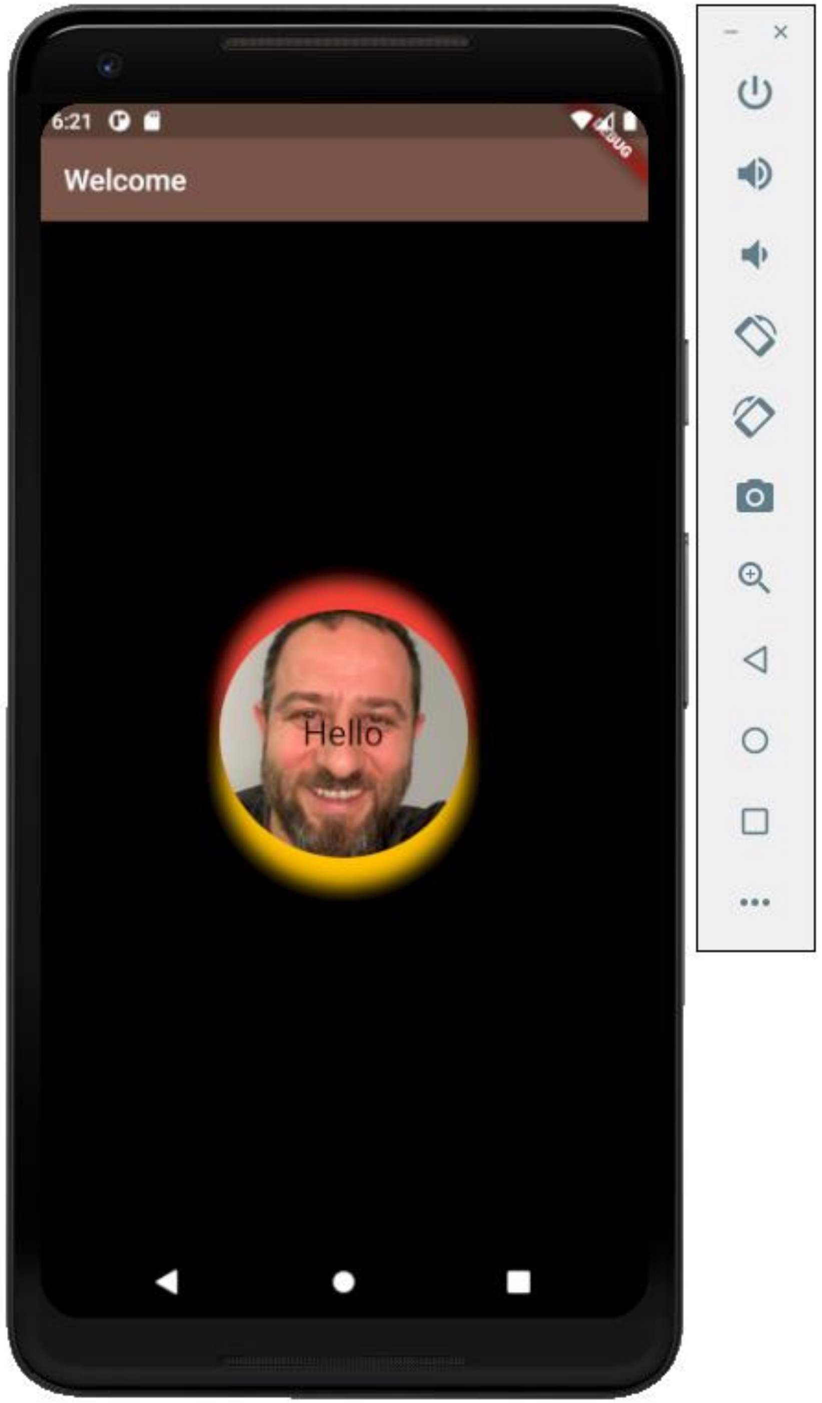
```
body: Center(
  child: Container(
    padding: const EdgeInsets.all(20),
    child: const FlutterLogo(
      style: FlutterLogoStyle.horizontal,
      size: 128,
      textColor: Colors.yellow,
    ), // FlutterLogo
    decoration: BoxDecoration(
      color: Colors.blue,
      //shape: BoxShape.circle,
      shape: BoxShape.rectangle,
      border: Border.all(
        width: 5,
        color: Colors.white,
      ), // Border.all
      //borderRadius: BorderRadius.circular(25),
      borderRadius: const BorderRadius.only(
        bottomLeft: Radius.circular(50),
        topRight: Radius.circular(50),
      ), // BorderRadius.only
    ), // BoxDecoration
  ), // Container
), // Center
```



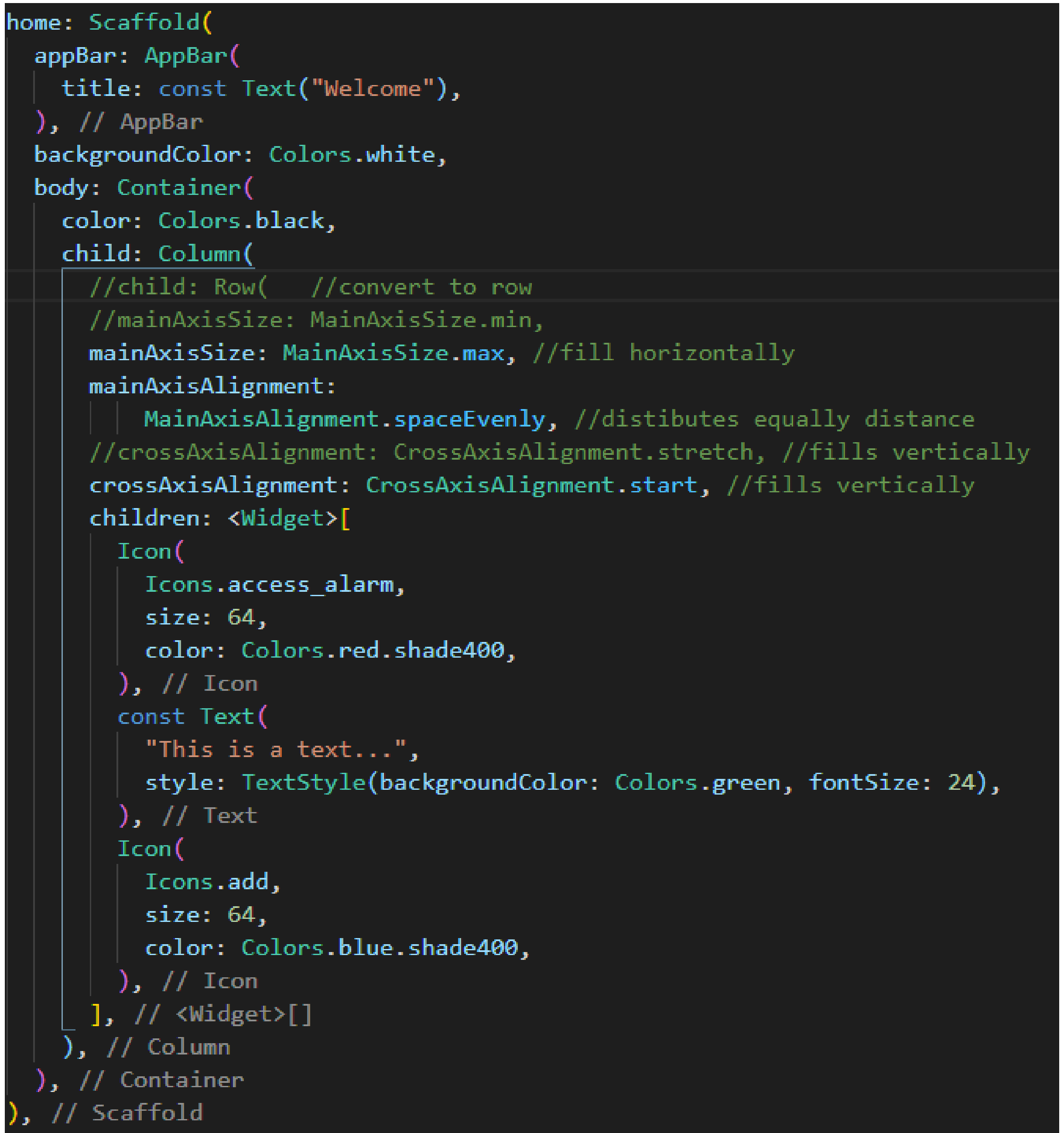
Images and Shadow



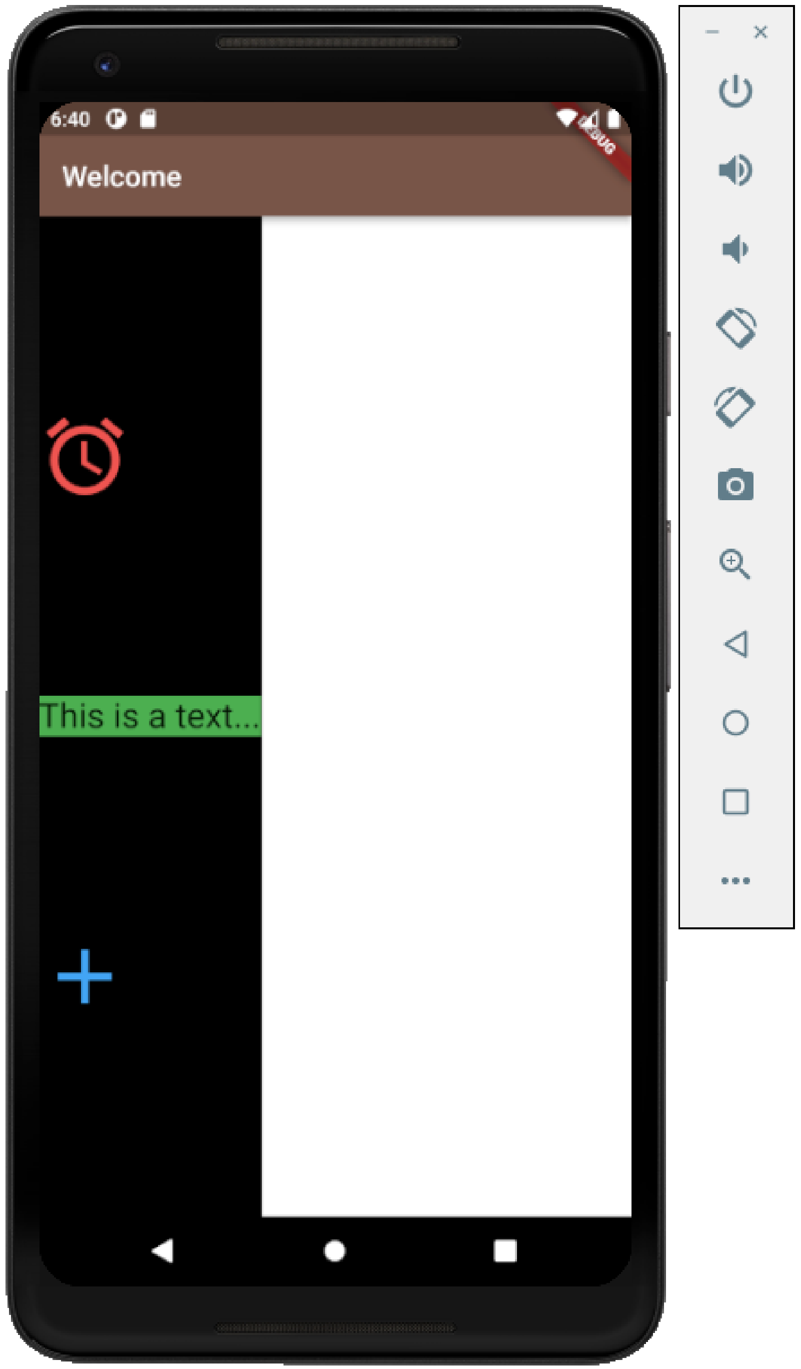
```
body: Center(
  child: Container(
    padding: const EdgeInsets.all(70),
    child: const Text(
      "Hello",
      style: TextStyle(fontSize: 24),
    ), // Text
    decoration: const BoxDecoration(
      shape: BoxShape.circle,
      image: DecorationImage(
        image: NetworkImage(
            "https://avesis.yildiz.edu.tr/user/image/4487"), // NetworkI
        fit: BoxFit.fitWidth,
        repeat: ImageRepeat.noRepeat,
      ), // DecorationImage
      boxShadow: [
        BoxShadow(
            color: Colors.amber, offset: Offset(0, 20), blurRadius: 10),
        BoxShadow(
            color: Colors.red, offset: Offset(0, -20), blurRadius: 10),
      ],
    ), // BoxDecoration
  ), // Container
), // Center
```



Row and Column Widgets

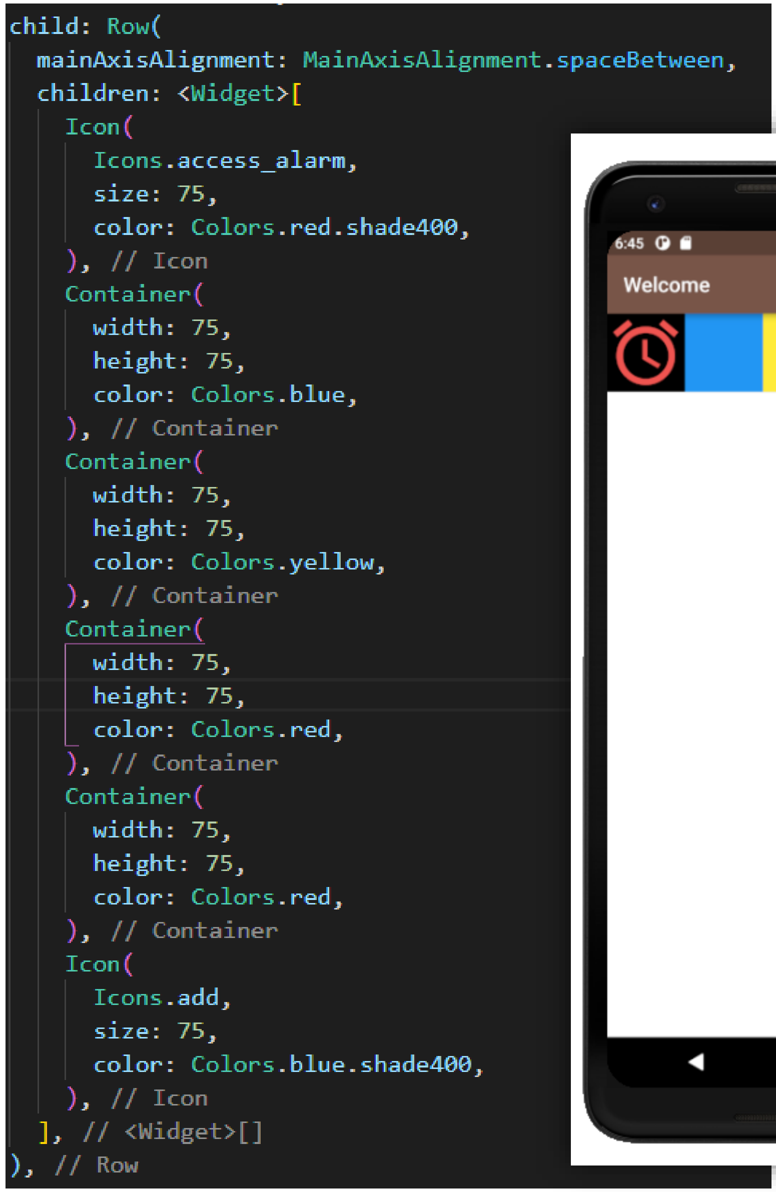


```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  backgroundColor: Colors.white,
  body: Container(
    color: Colors.black,
    child: Column(
      //child: Row(   //convert to row
      //mainAxisSize: MainAxisSize.min,
      mainAxisSize: MainAxisSize.max, //fill horizontally
      mainAxisAlignment:
          MainAxisAlignment.spaceEvenly, //distibutes equally distance
      //crossAxisAlignment: CrossAxisAlignment.stretch, //fills vertically
      crossAxisAlignment: CrossAxisAlignment.start, //fills vertically
      children: <Widget>[
        Icon(
          Icons.access_alarm,
          size: 64,
          color: Colors.red.shade400,
        ), // Icon
        const Text(
          "This is a text...",
          style: TextStyle(backgroundColor: Colors.green, fontSize: 24),
        ), // Text
        Icon(
          Icons.add,
          size: 64,
          color: Colors.blue.shade400,
        ), // Icon
      ], // <Widget>[]
    ), // Column
  ), // Container
), // Scaffold
```

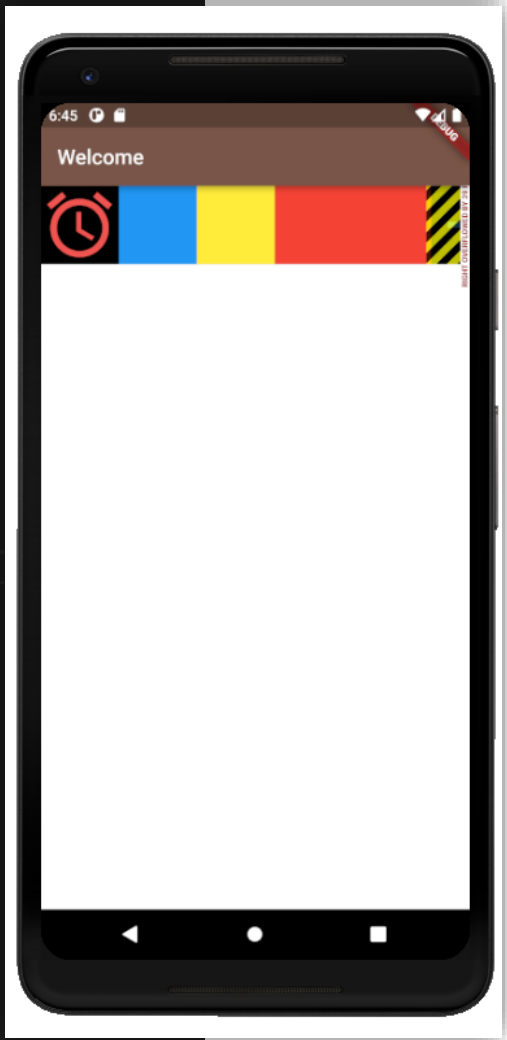


Expanded and Flexible Widgeds

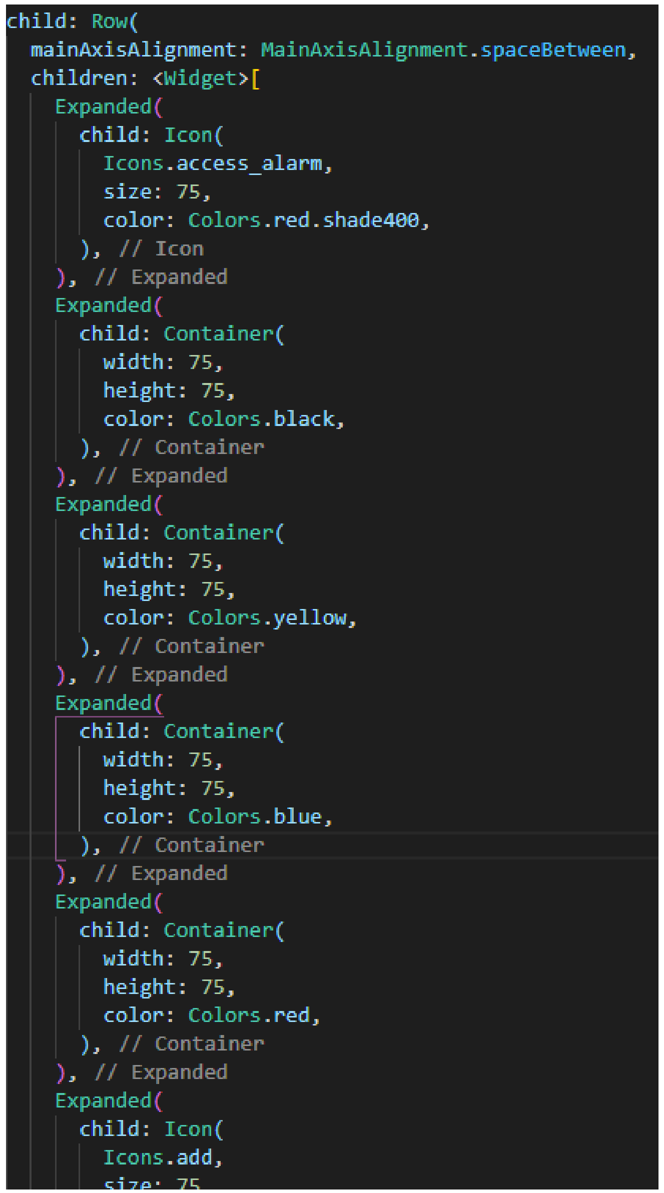
- Space Problem



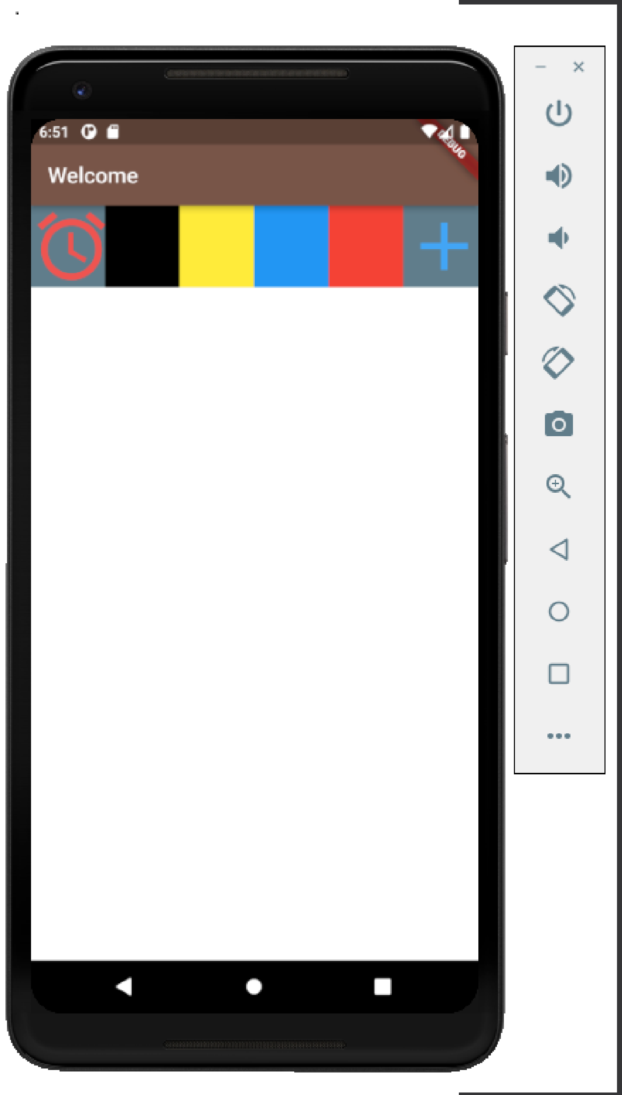
```
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: <Widget>[
    Icon(
      Icons.access_alarm,
      size: 75,
      color: Colors.red.shade400,
    ), // Icon
    Container(
      width: 75,
      height: 75,
      color: Colors.blue,
    ), // Container
    Container(
      width: 75,
      height: 75,
      color: Colors.yellow,
    ), // Container
    Container(
      width: 75,
      height: 75,
      color: Colors.red,
    ), // Container
    Container(
      width: 75,
      height: 75,
      color: Colors.red,
    ), // Container
    Icon(
      Icons.add,
      size: 75,
      color: Colors.blue.shade400,
    ), // Icon
  ], // <Widget>[]
), // Row
```



Solution:

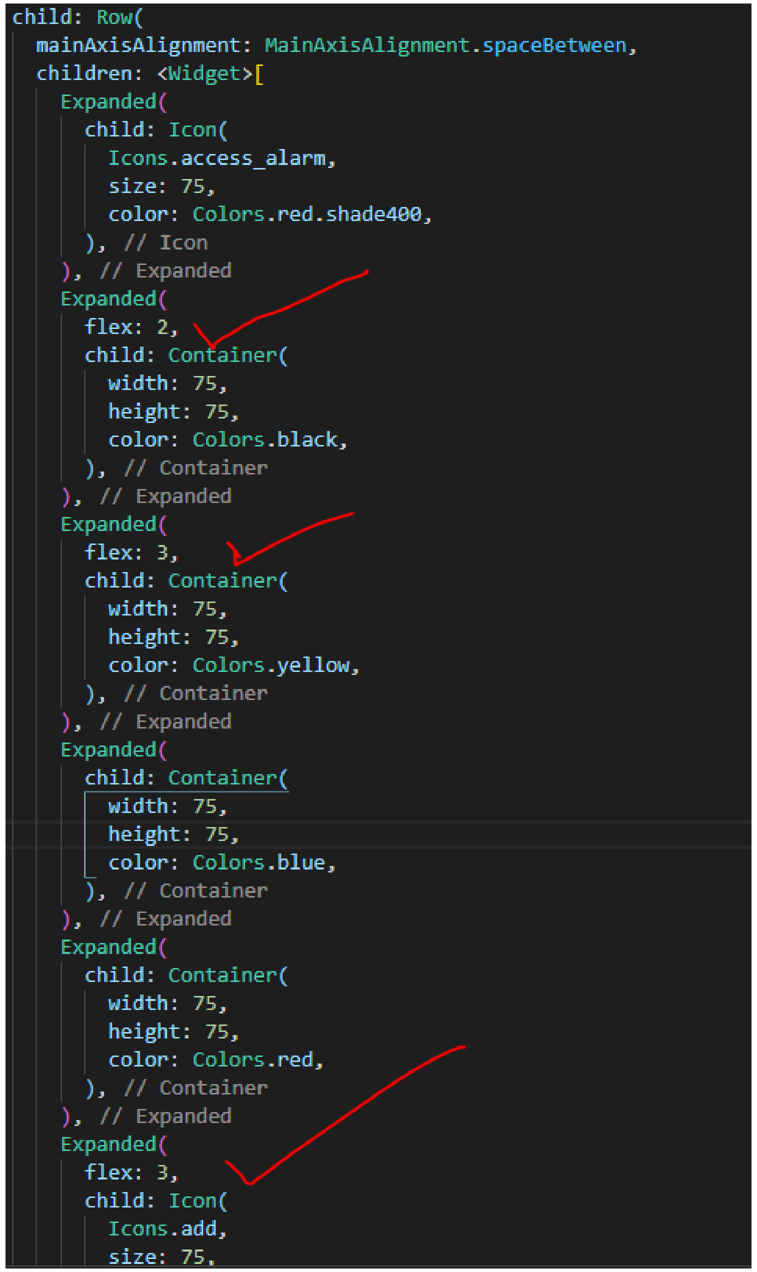


```
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: <Widget>[
    Expanded(
      child: Icon(
        Icons.access_alarm,
        size: 75,
        color: Colors.red.shade400,
      ), // Icon
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.black,
      ), // Container
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.yellow,
      ), // Container
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.blue,
      ), // Container
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.red,
      ), // Container
    ), // Expanded
    Expanded(
      child: Icon(
        Icons.add,
        size: 75
```

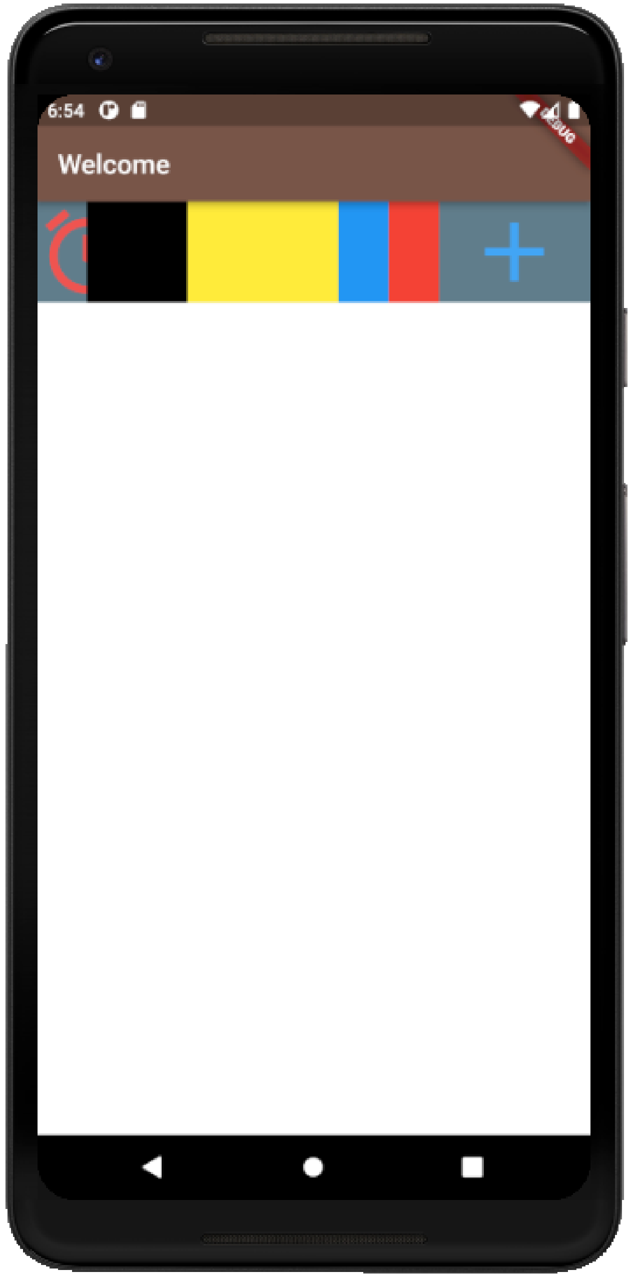


Flex attribute

- ..

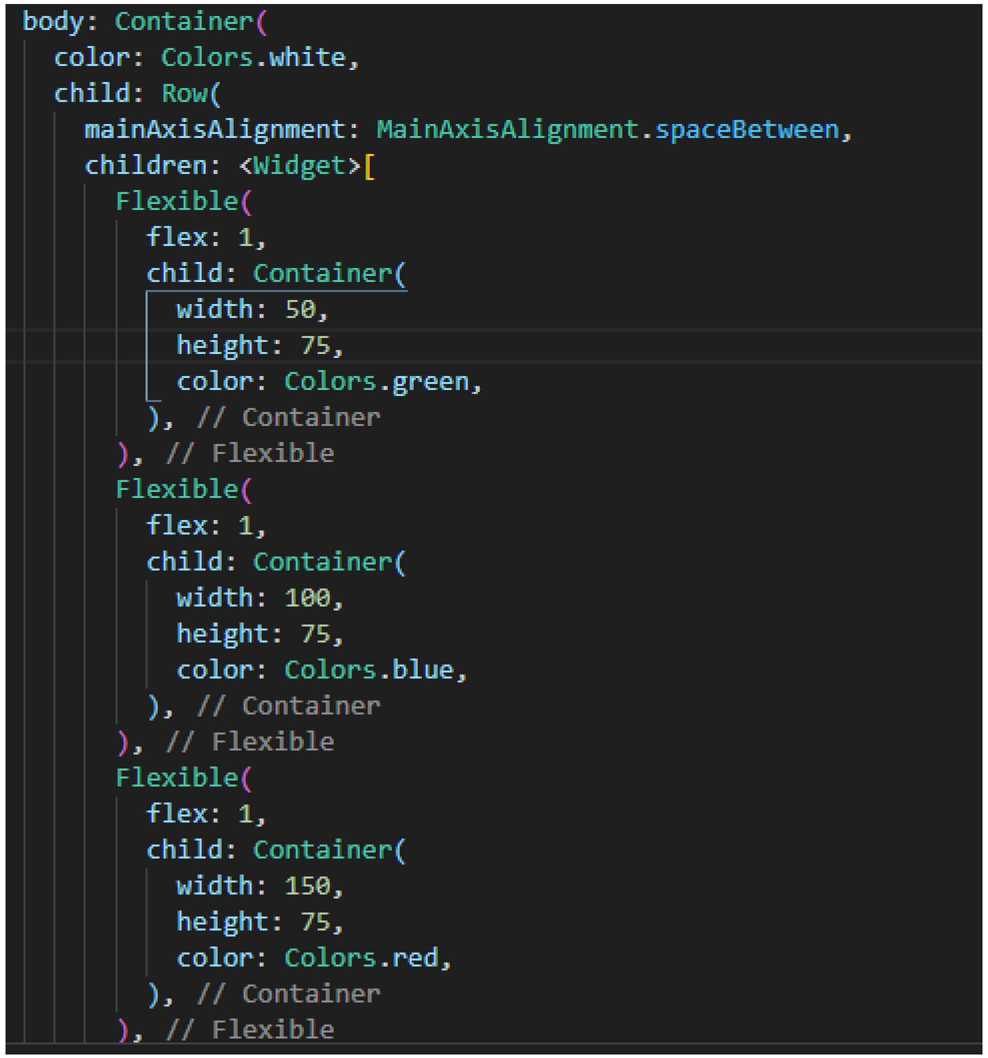


```
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: <Widget>[
    Expanded(
      child: Icon(
        Icons.access_alarm,
        size: 75,
        color: Colors.red.shade400,
      ), // Icon
    ), // Expanded
    Expanded(
      flex: 2,
      child: Container(
        width: 75,
        height: 75,
        color: Colors.black,
      ), // Container
    ), // Expanded
    Expanded(
      flex: 3,
      child: Container(
        width: 75,
        height: 75,
        color: Colors.yellow,
      ), // Container
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.blue,
      ), // Container
    ), // Expanded
    Expanded(
      child: Container(
        width: 75,
        height: 75,
        color: Colors.red,
      ), // Container
    ), // Expanded
    Expanded(
      flex: 3,
      child: Icon(
        Icons.add,
        size: 75,
```

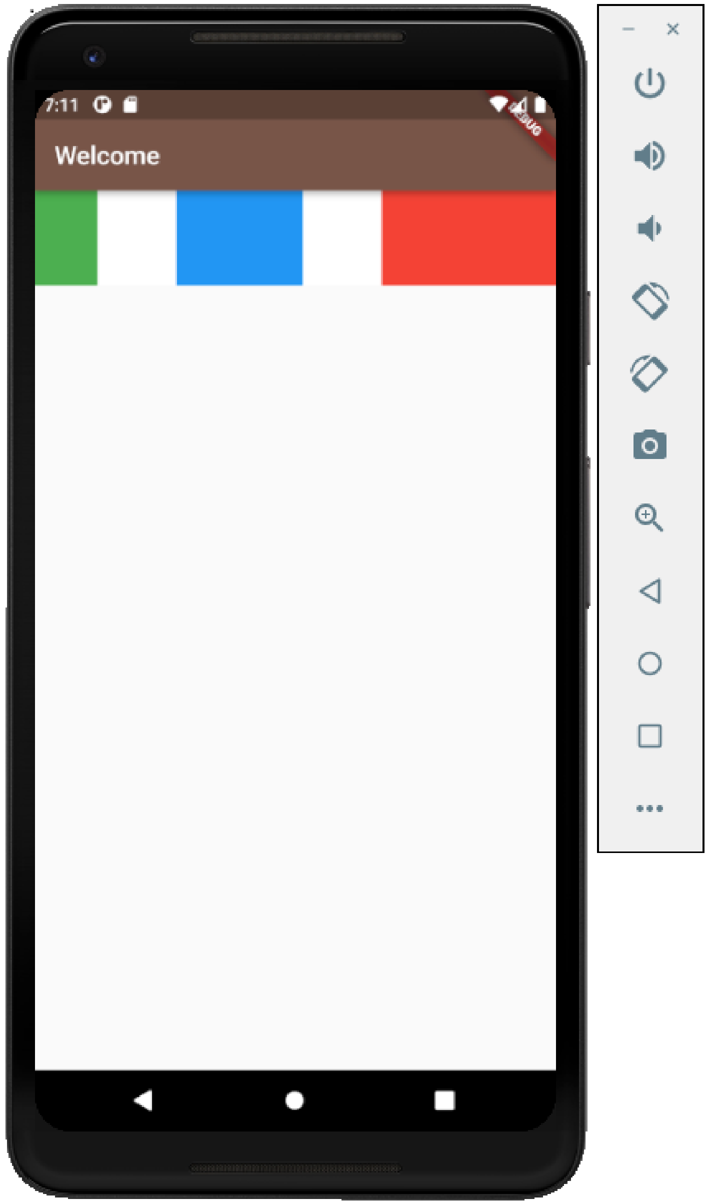


Flexible Container Widget

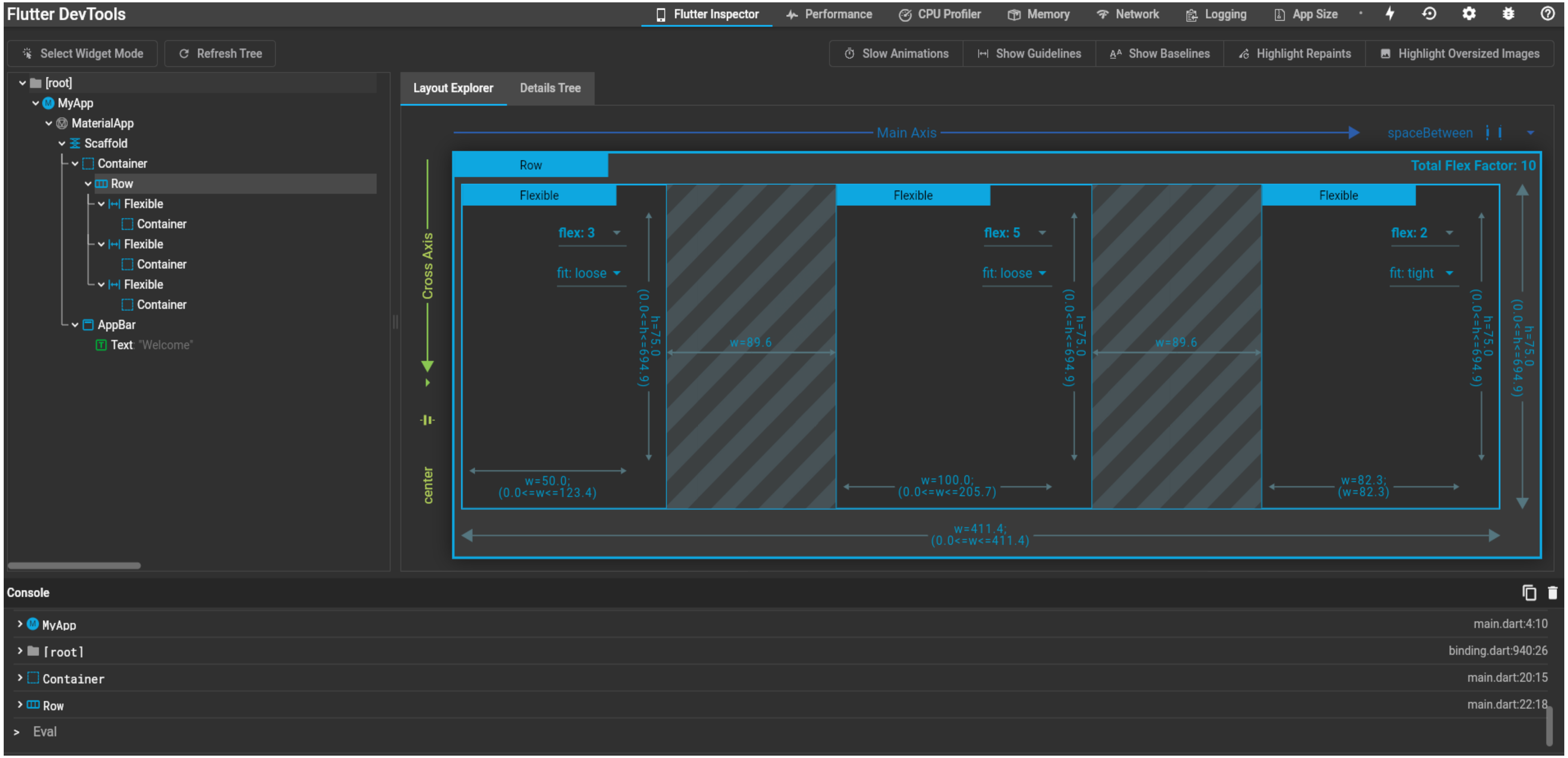
- ..



```
body: Container(
  color: Colors.white,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: <Widget>[
      Flexible(
        flex: 1,
        child: Container(
          width: 50,
          height: 75,
          color: Colors.green,
        ), // Container
      ), // Flexible
      Flexible(
        flex: 1,
        child: Container(
          width: 100,
          height: 75,
          color: Colors.blue,
        ), // Container
      ), // Flexible
      Flexible(
        flex: 1,
        child: Container(
          width: 150,
          height: 75,
          color: Colors.red,
        ), // Container
      ), // Flexible
```



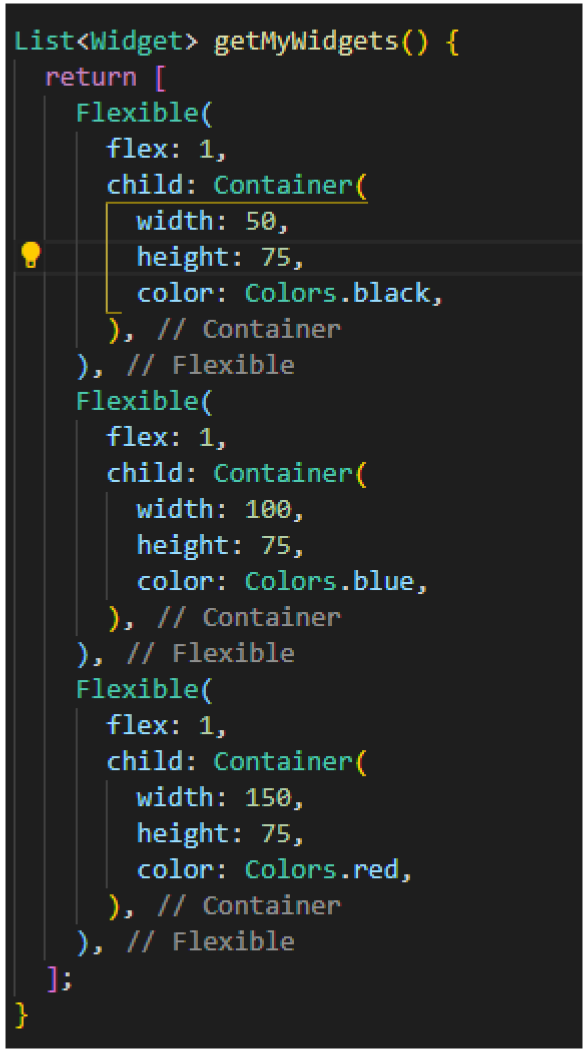
Page Inspector to find design problems.



```
> MyApp                                          main.dart:4:10
> [root]                                    binding.dart:940:26
> Container                                      main.dart:20:15
> Row                                            main.dart:22:18
> Eval
```

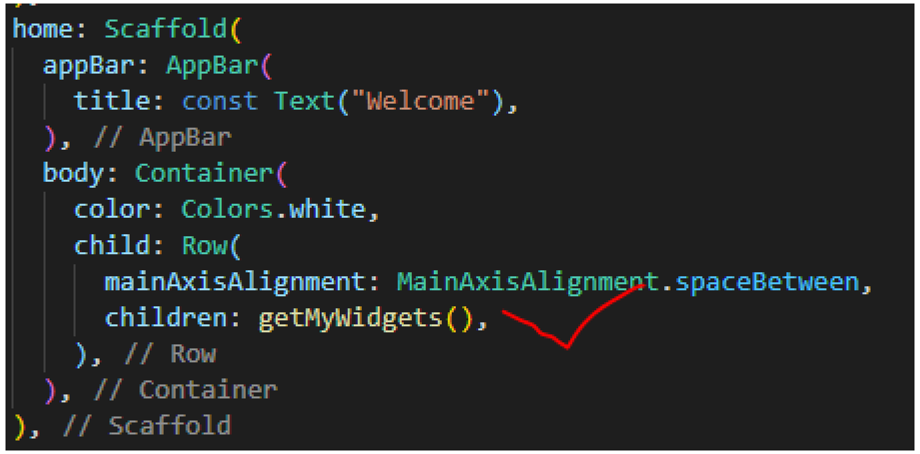
Separating Widgets by using Methods

- Method



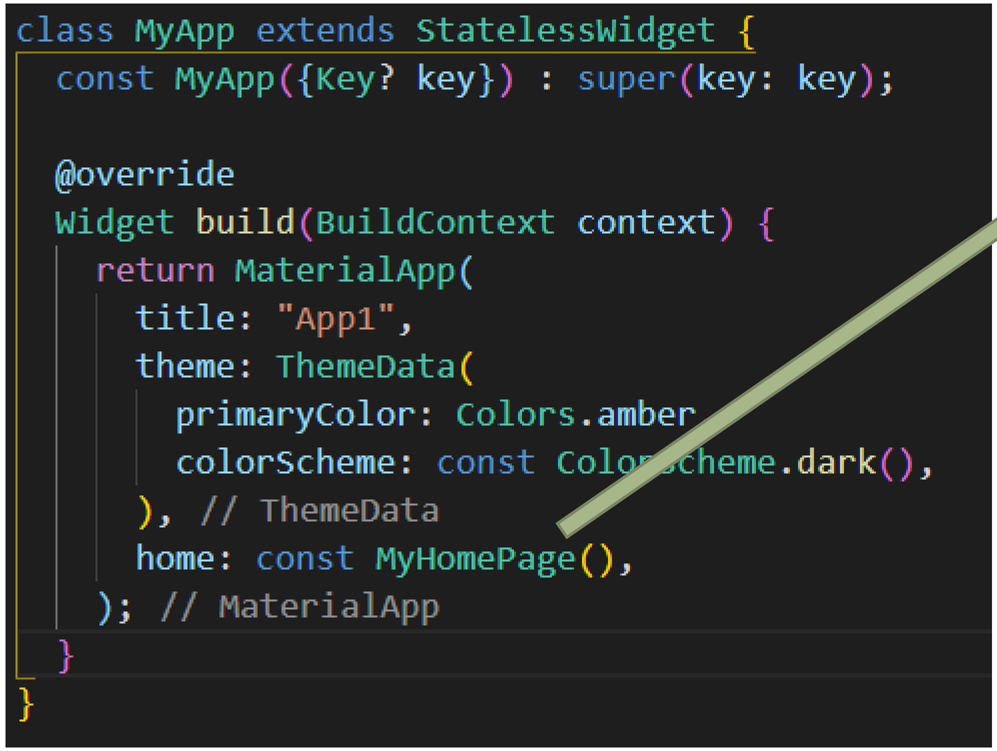
```
List<Widget> getMyWidgets() {
  return [
    Flexible(
      flex: 1,
      child: Container(
        width: 50,
        height: 75,
        color: Colors.black,
      ), // Container
    ), // Flexible
    Flexible(
      flex: 1,
      child: Container(
        width: 100,
        height: 75,
        color: Colors.blue,
      ), // Container
    ), // Flexible
    Flexible(
      flex: 1,
      child: Container(
        width: 150,
        height: 75,
        color: Colors.red,
      ), // Container
    ), // Flexible
  ];
}
```

Usage:

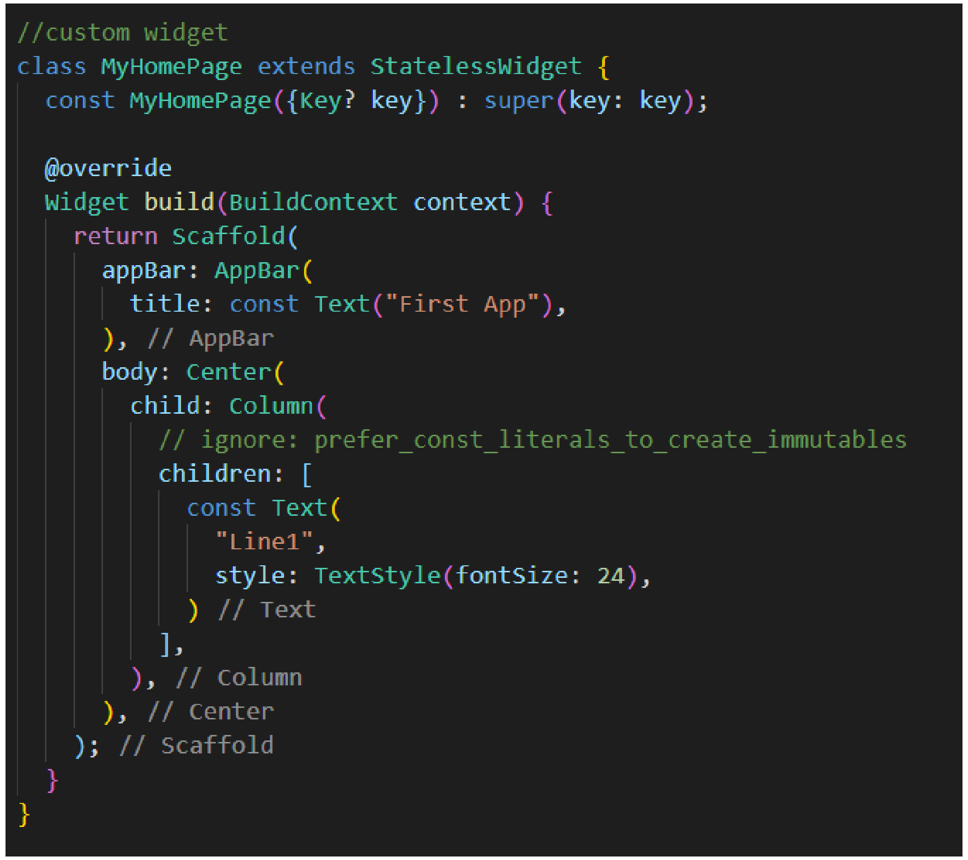


```
home: Scaffold(
  appBar: AppBar(
    title: const Text("Welcome"),
  ), // AppBar
  body: Container(
    color: Colors.white,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: getMyWidgets(),
    ), // Row
  ), // Container
), // Scaffold
```

Custom Widgets



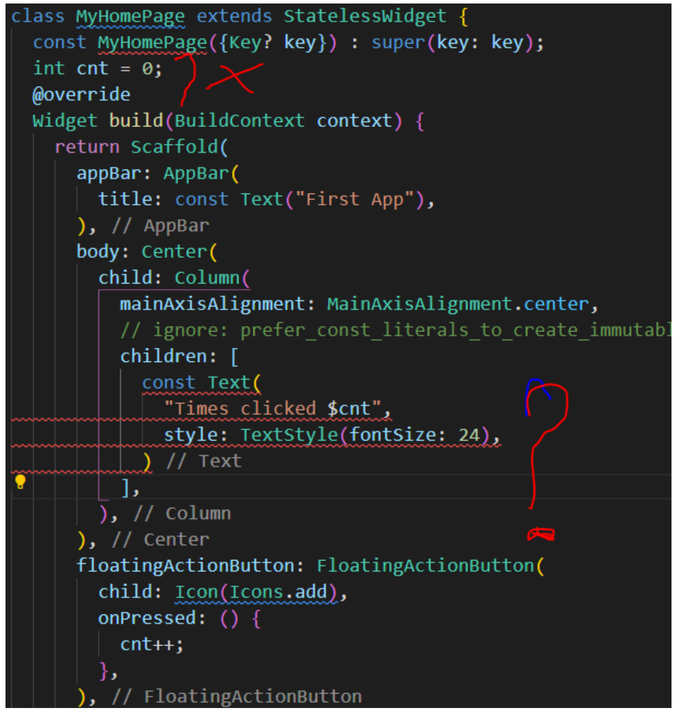
```dart
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "App1",
      theme: ThemeData(
        primaryColor: Colors.amber,
        colorScheme: const ColorScheme.dark(),
      ), // ThemeData
      home: const MyHomePage(),
    ); // MaterialApp
  }
}
```



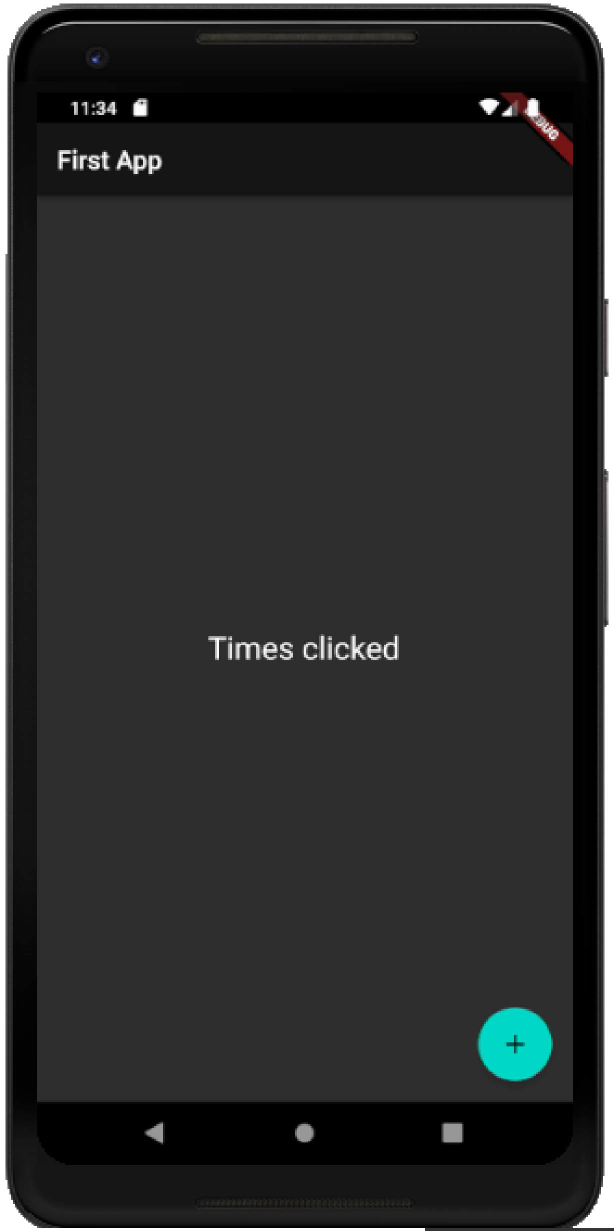
```dart
//custom widget
class MyHomePage extends StatelessWidget {
  const MyHomePage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("First App"),
      ), // AppBar
      body: Center(
        child: Column(
          // ignore: prefer_const_literals_to_create_immutables
          children: [
            const Text(
              "Line1",
              style: TextStyle(fontSize: 24),
            ) // Text
          ],
        ), // Column
      ), // Center
    ); // Scaffold
  }
}
```

Stateless Widgets

- Drawn on the screen one time and never changed.
- No user interaction,
- Cannot change remove or resize object once widget rendered
- Forexample: We cannot change text value and counter increment in the right example.
- Everything must be finished before bulding widget.

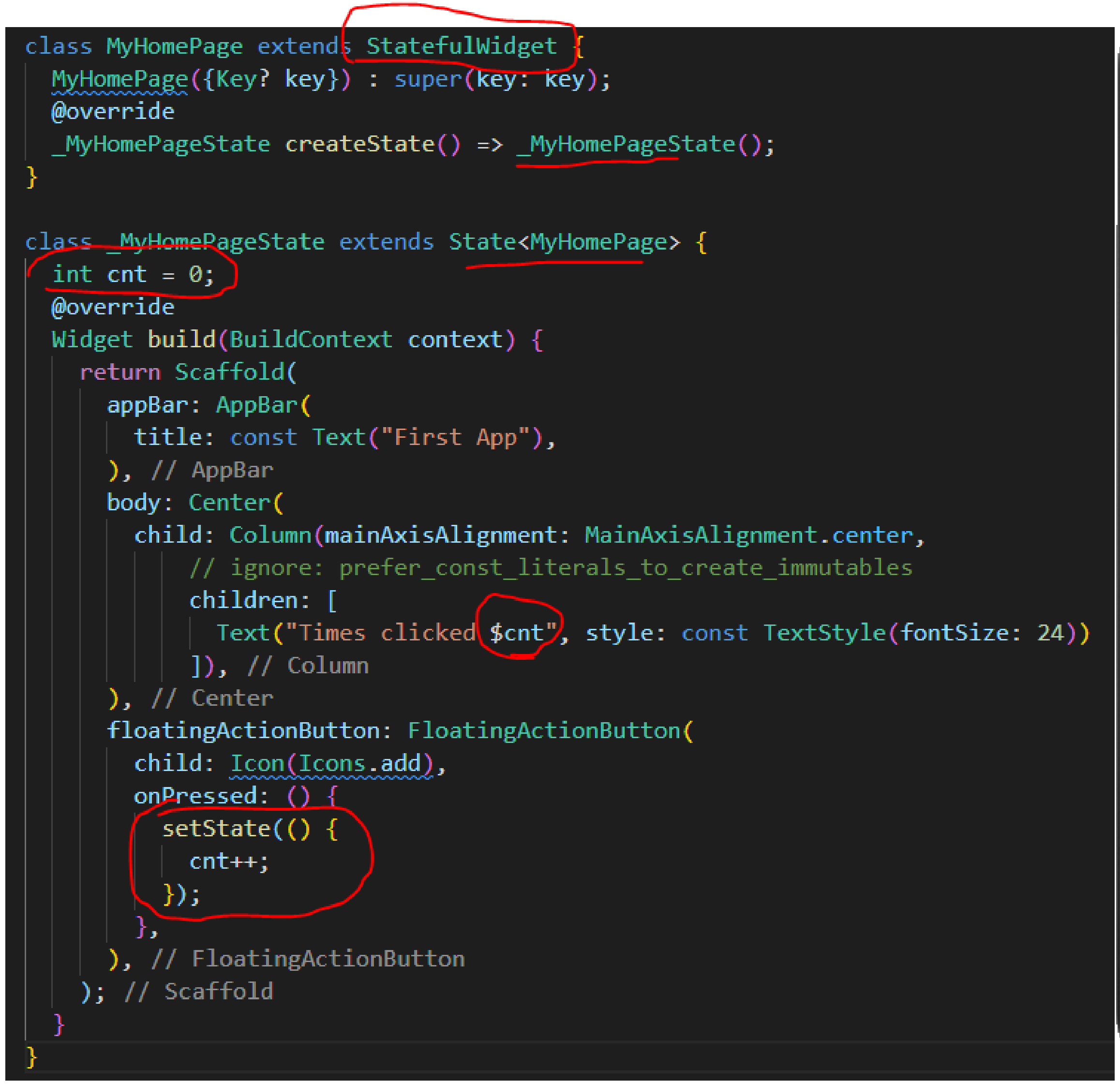


```
class MyHomePage extends StatelessWidget {
  const MyHomePage({Key? key}) : super(key: key);
  int cnt = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("First App"),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          // ignore: prefer_const_literals_to_create_immutabl
          children: [
            const Text(
              "Times clicked $cnt",
              style: TextStyle(fontSize: 24),
            ) // Text
          ],
        ), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.add),
        onPressed: () {
          cnt++;
        },
      ), // FloatingActionButton
```

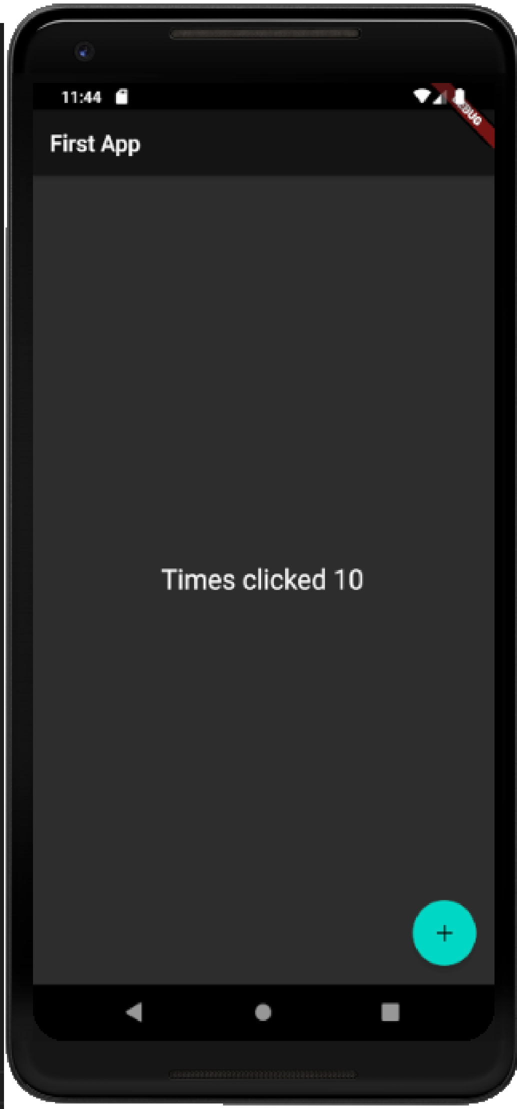


Statefull Widget

- This widwet is interactive Widget,
- We can change everthing on it.
- Used setState(); callback for the state changes

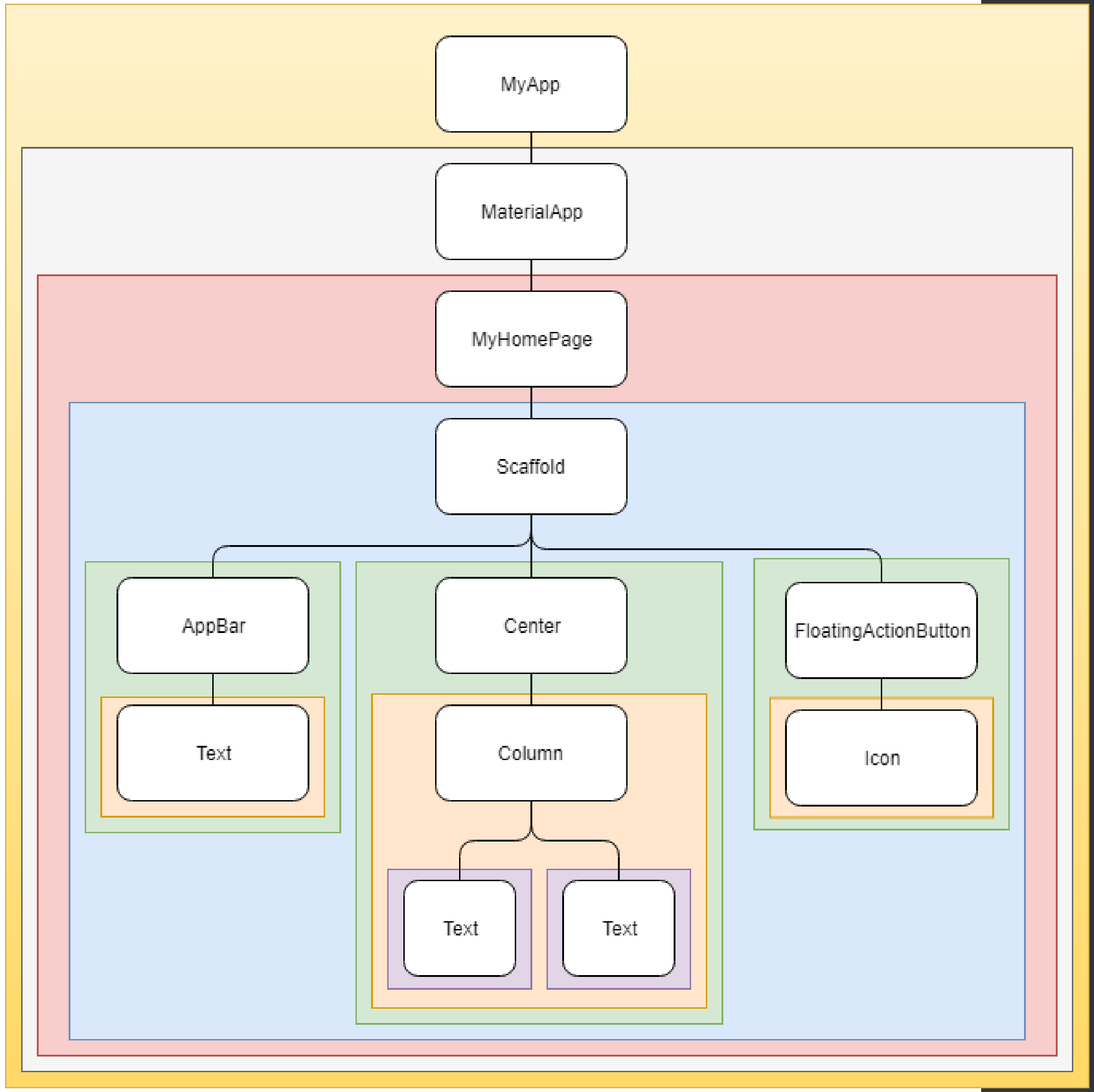


```dart
class MyHomePage extends StatefulWidget {
  MyHomePage({Key? key}) : super(key: key);
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int cnt = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("First App"),
      ), // AppBar
      body: Center(
        child: Column(mainAxisAlignment: MainAxisAlignment.center,
            // ignore: prefer_const_literals_to_create_immutables
            children: [
              Text("Times clicked $cnt", style: const TextStyle(fontSize: 24))
            ]), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.add),
        onPressed: () {
          setState(() {
            cnt++;
          });
        },
      ), // FloatingActionButton
    ); // Scaffold
  }
}
```

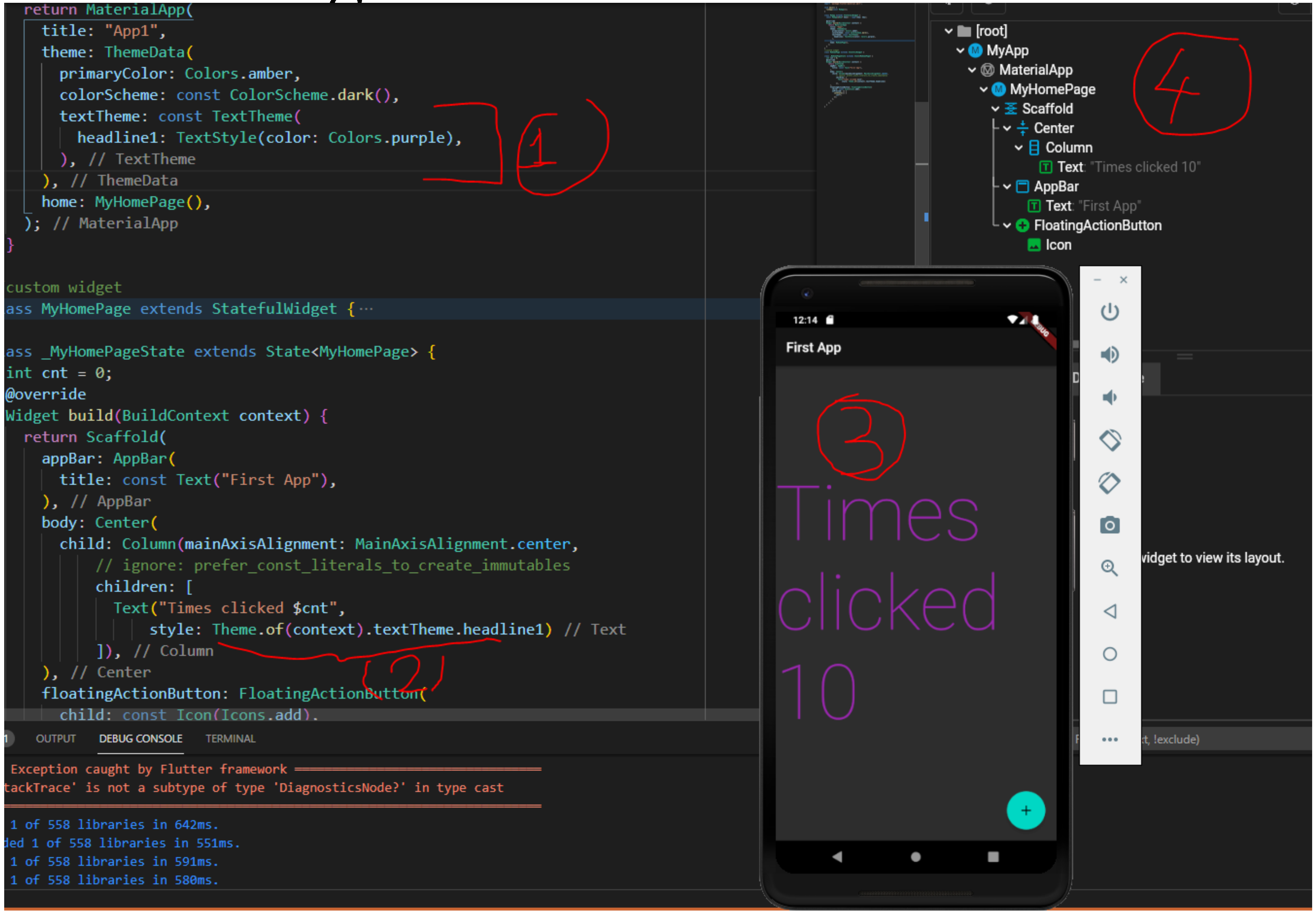


BuildContext

- A BuildContext is nothing else but a reference to the location of a Widget within the tree structure of all the Widgets which are built.
- In short, think of a BuildContext as the part of Widgets tree where the Widget is attached to this tree.
- A BuildContext only belongs to one widget.
- If a widget ‘A’ has children widgets, the BuildContext of widget ‘A’ will become the parent BuildContext of the direct children BuildContexts.
- BuildContexts are chained and are composing a tree of BuildContexts (parent-children relationship).
- From this statement we can derive that from a child BuildContext, it is easily possible to find an ancestor (= parent) Widget.
- An example is, considering the Scaffold > Center > Column > Text:
- context.ancestorWidgetOfExactType(Scaffold) => returns the first Scaffold by going up to tree structure from the Text context.
- From a parent BuildContext, it is also possible to find a descendant (= child) Widget but it is not advised to do so (we will discuss this later).



Traversing Contexts



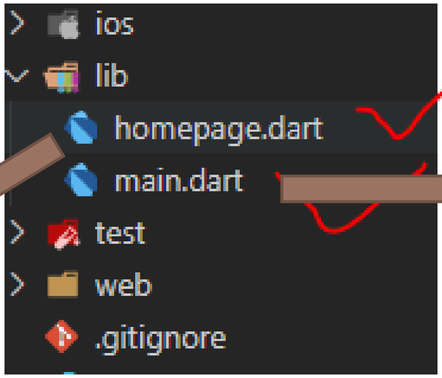
```
  return MaterialApp(
    title: "App1",
    theme: ThemeData(
      primaryColor: Colors.amber,
      colorScheme: const ColorScheme.dark(),
      textTheme: const TextTheme(
        headline1: TextStyle(color: Colors.purple),
      ), // TextTheme
    ), // ThemeData
    home: MyHomePage(),
  ); // MaterialApp
}

custom widget
ass MyHomePage extends StatefulWidget {···

ass _MyHomePageState extends State<MyHomePage> {
int cnt = 0;
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("First App"),
    ), // AppBar
    body: Center(
      child: Column(mainAxisAlignment: MainAxisAlignment.center,
          // ignore: prefer_const_literals_to_create_immutables
          children: [
            Text("Times clicked $cnt",
                style: Theme.of(context).textTheme.headline1) // Text
          ]), // Column
    ), // Center
    floatingActionButton: FloatingActionButton(
      child: const Icon(Icons.add),
```

```
[root]
  MyApp
    MaterialApp
      MyHomePage
        Scaffold
          Center
            Column
              Text: "Times clicked 10"
          AppBar
            Text: "First App"
          FloatingActionButton
            Icon
```

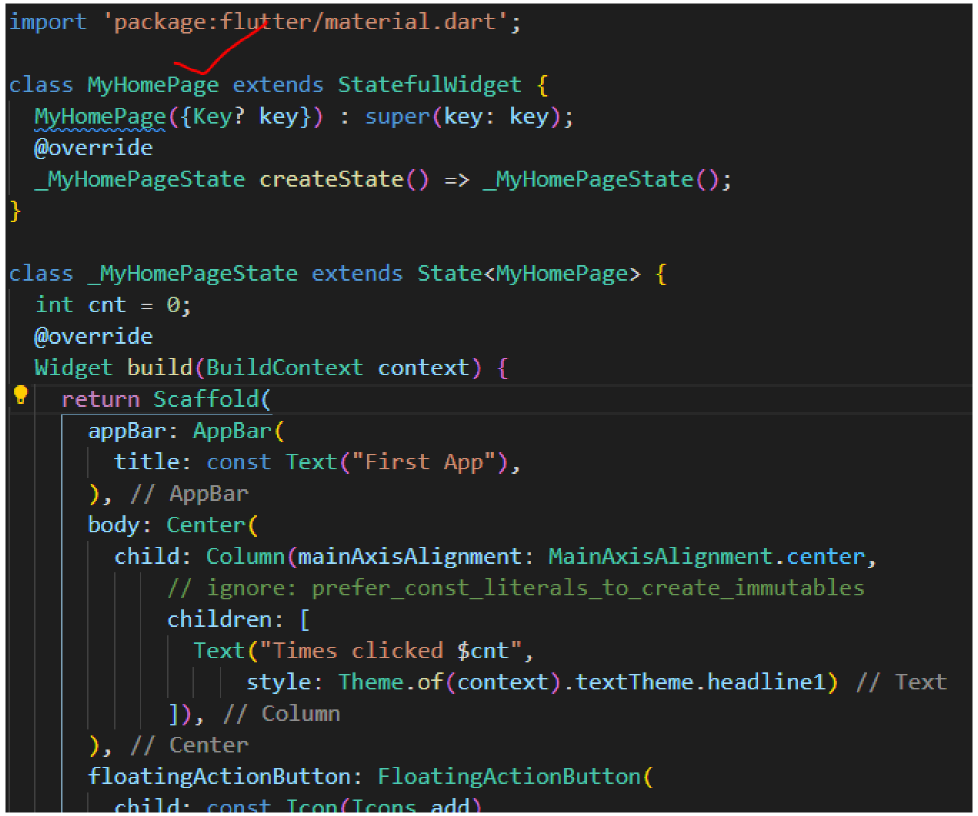
```
Exception caught by Flutter framework
tackTrace' is not a subtype of type 'DiagnosticsNode?' in type cast

 1 of 558 libraries in 642ms.
ded 1 of 558 libraries in 551ms.
 1 of 558 libraries in 591ms.
 1 of 558 libraries in 580ms.
```

Separating long dart files to additional files

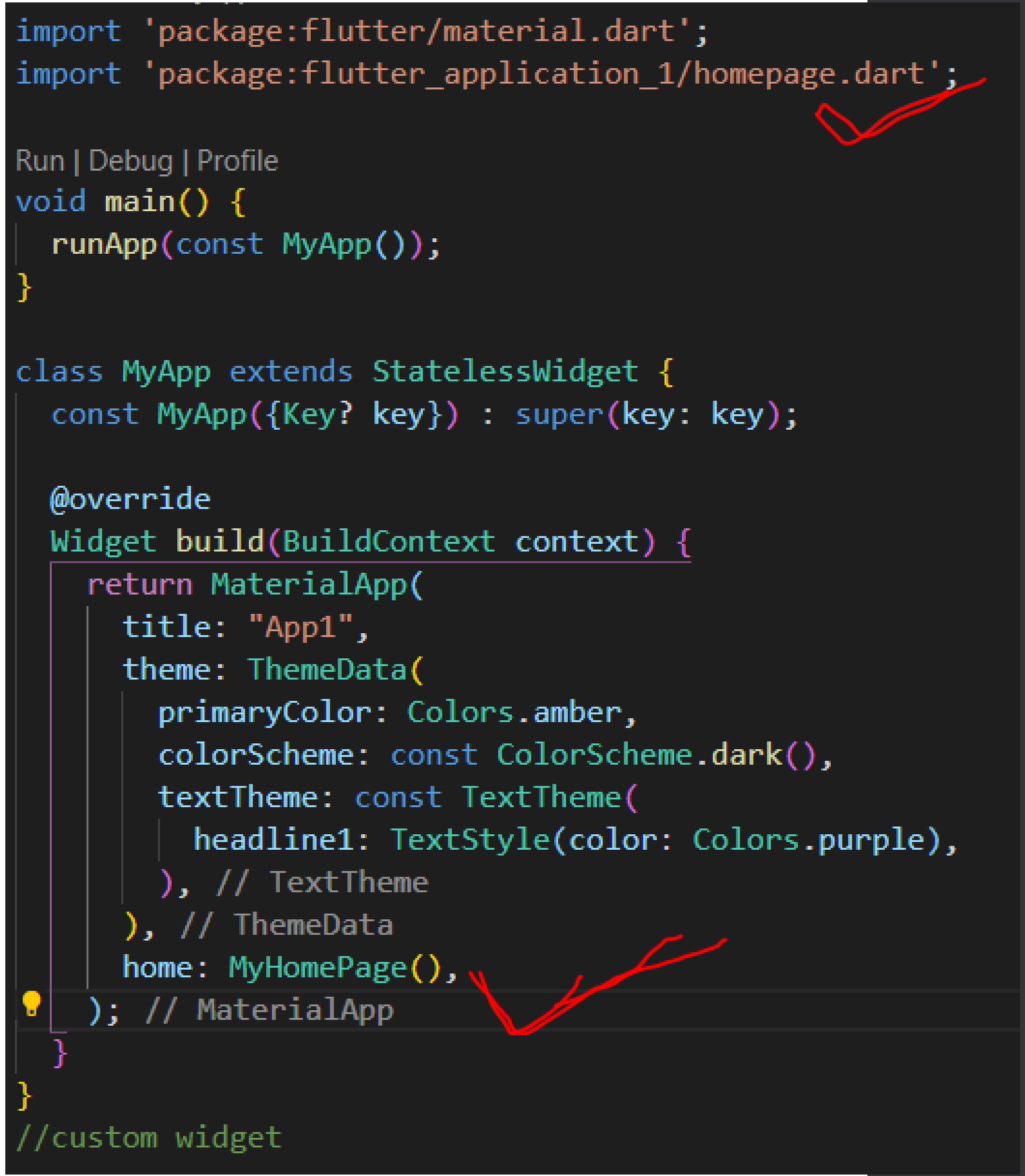
- ..



```
> ios
v lib
    homepage.dart
    main.dart
> test
> web
  .gitignore
```



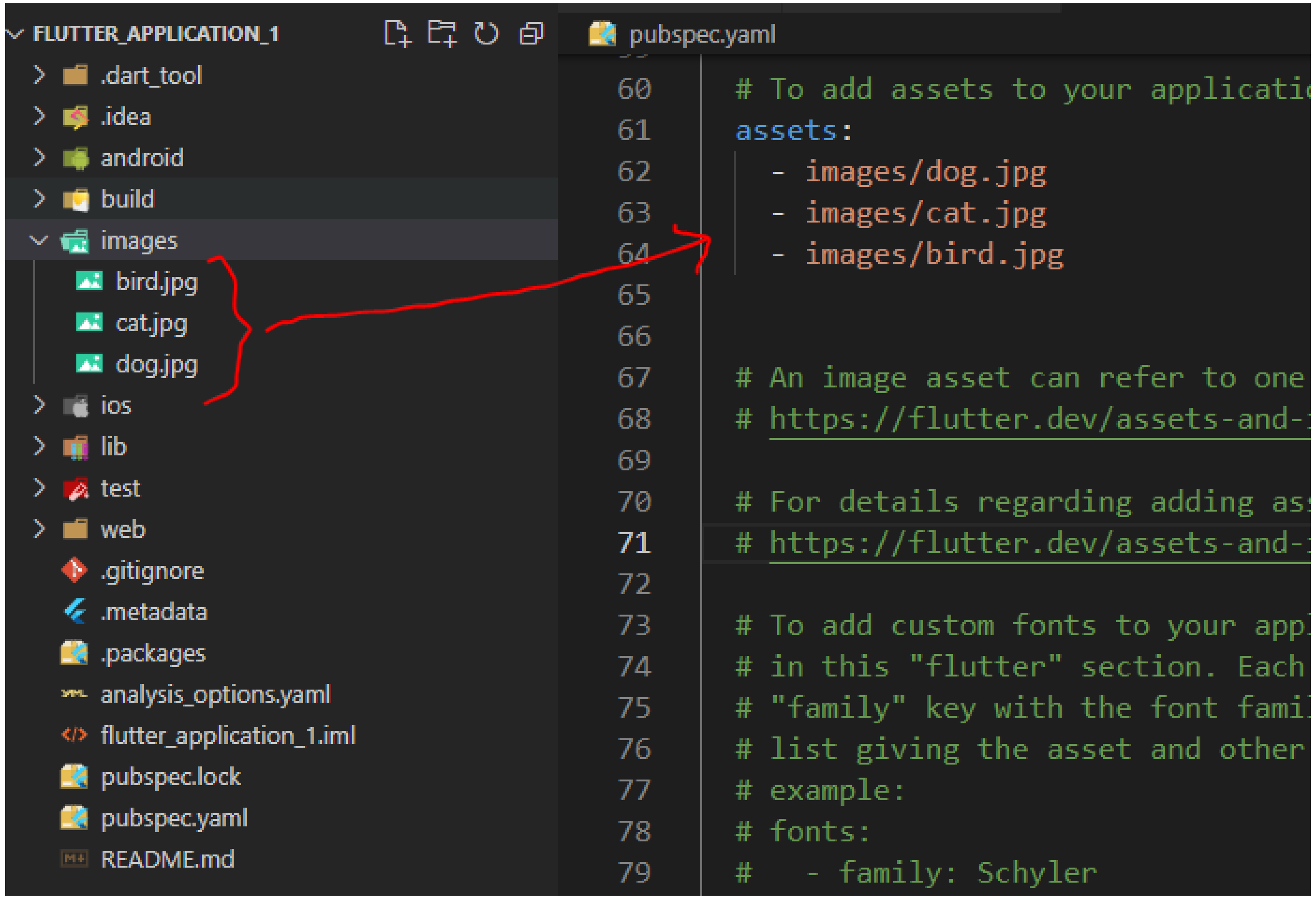
```dart
import 'package:flutter/material.dart';

class MyHomePage extends StatefulWidget {
  MyHomePage({Key? key}) : super(key: key);
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int cnt = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("First App"),
      ), // AppBar
      body: Center(
        child: Column(mainAxisAlignment: MainAxisAlignment.center,
            // ignore: prefer_const_literals_to_create_immutables
            children: [
              Text("Times clicked $cnt",
                  style: Theme.of(context).textTheme.headline1) // Text
            ]), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        child: const Icon(Icons.add)
```



```dart
import 'package:flutter/material.dart';
import 'package:flutter_application_1/homepage.dart';

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "App1",
      theme: ThemeData(
        primaryColor: Colors.amber,
        colorScheme: const ColorScheme.dark(),
        textTheme: const TextTheme(
          headline1: TextStyle(color: Colors.purple),
        ), // TextTheme
      ), // ThemeData
      home: MyHomePage(),
    ); // MaterialApp
  }
}
//custom widget
```

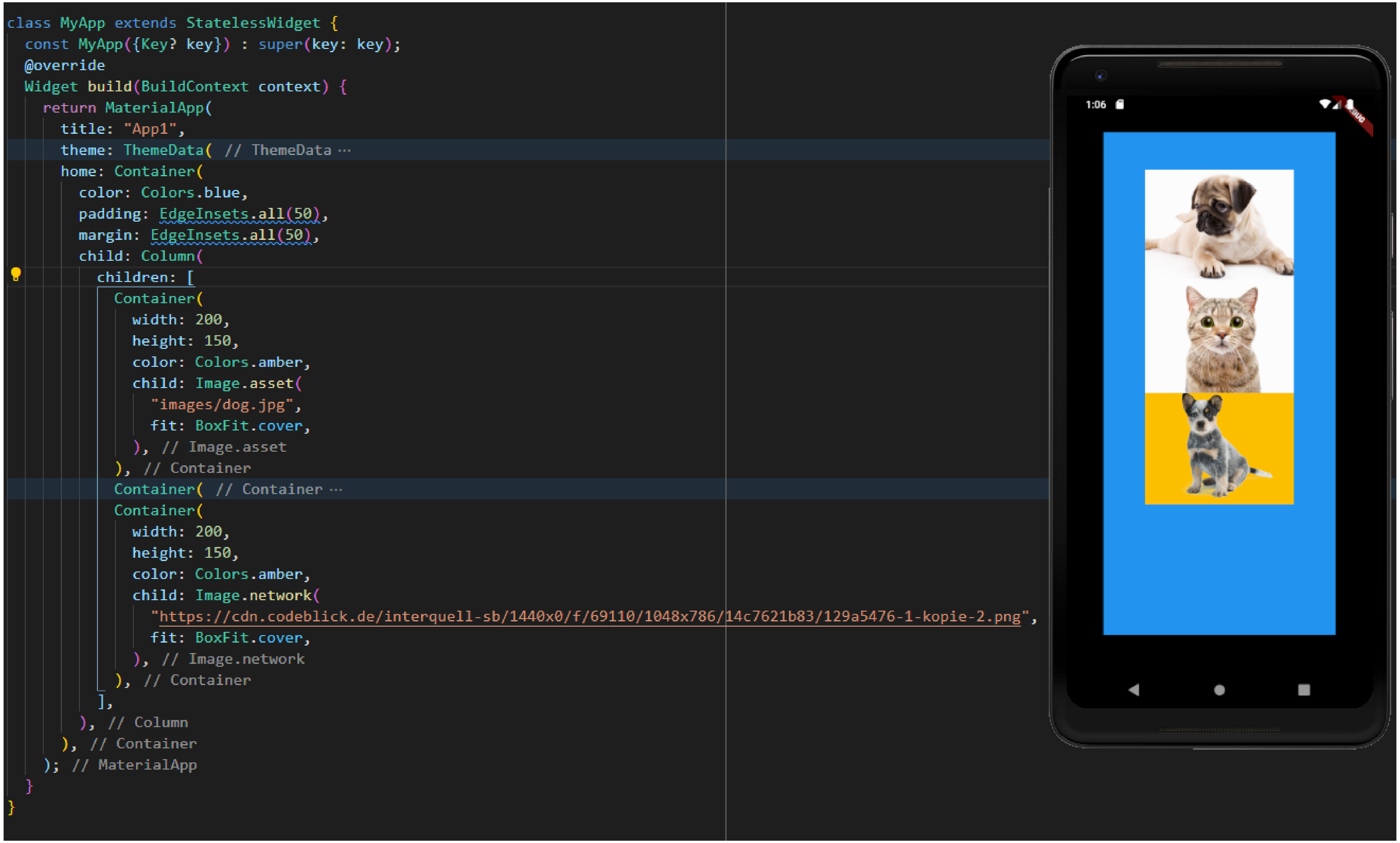
Using Assests

- Assets configurations is specified in **pubspec.yaml** file.
- We can create new folders and files in Project.
- To reach this file, full paths must be specified in pubspec.yaml file.



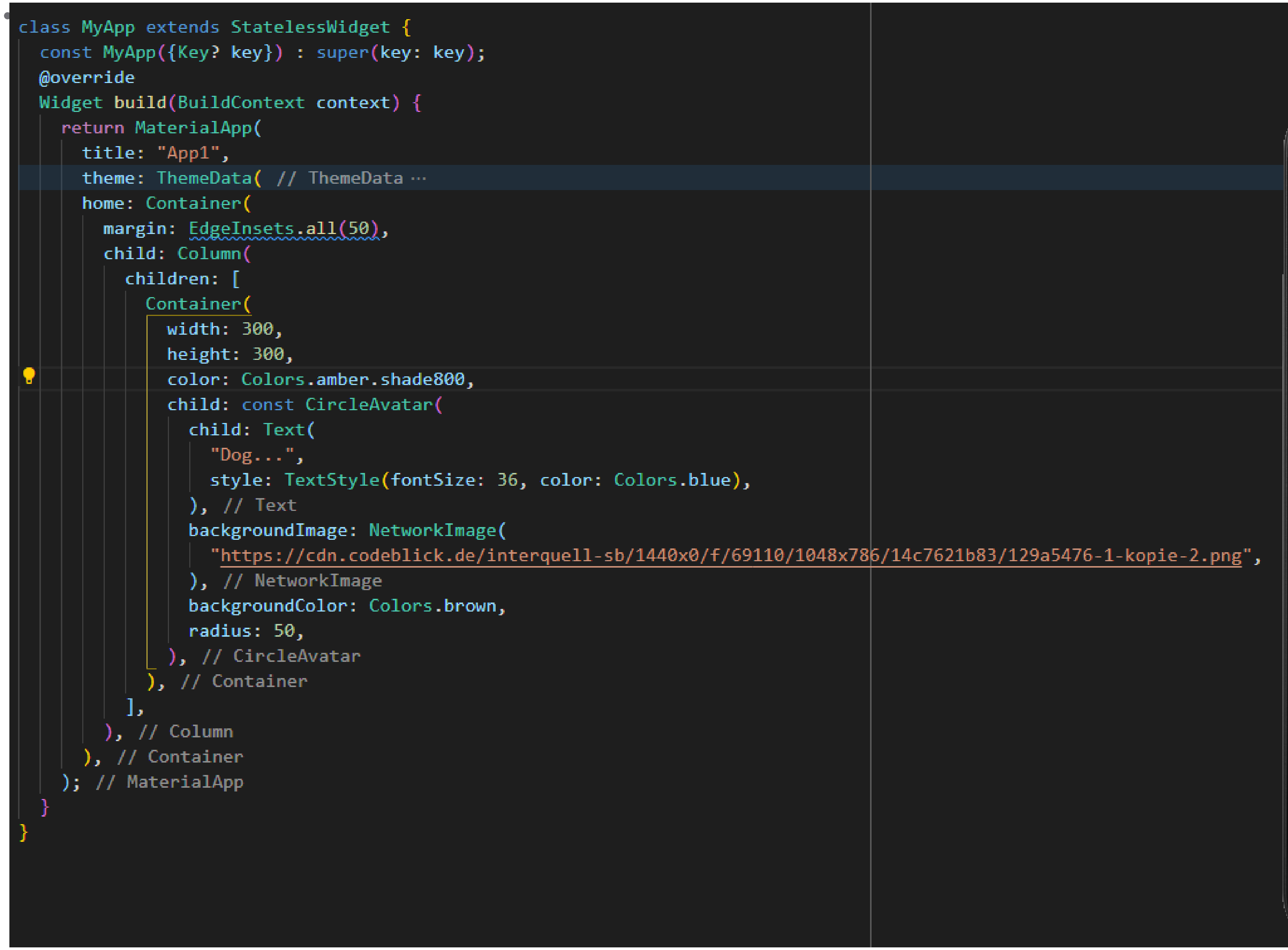
```yaml
  # To add assets to your applicati
  assets:
    - images/dog.jpg
    - images/cat.jpg
    - images/bird.jpg


  # An image asset can refer to one
  # https://flutter.dev/assets-and-

  # For details regarding adding as
  # https://flutter.dev/assets-and-

  # To add custom fonts to your app
  # in this "flutter" section. Each
  # "family" key with the font fami
  # list giving the asset and other
  # example:
  # fonts:
  #   - family: Schyler
```

Asset and Network Images and Styling

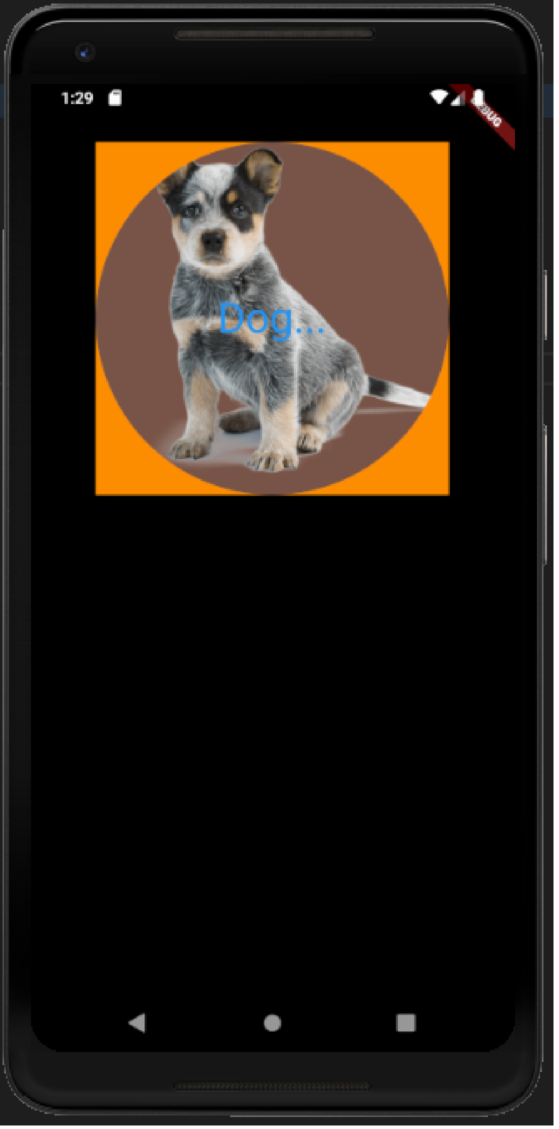


```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "App1",
      theme: ThemeData( // ThemeData ···
      home: Container(
        color: Colors.blue,
        padding: EdgeInsets.all(50),
        margin: EdgeInsets.all(50),
        child: Column(
          children: [
            Container(
              width: 200,
              height: 150,
              color: Colors.amber,
              child: Image.asset(
                "images/dog.jpg",
                fit: BoxFit.cover,
              ), // Image.asset
            ), // Container
            Container( // Container ···
            Container(
              width: 200,
              height: 150,
              color: Colors.amber,
              child: Image.network(
                "https://cdn.codeblick.de/interquell-sb/1440x0/f/69110/1048x786/14c7621b83/129a5476-1-kopie-2.png",
                fit: BoxFit.cover,
              ), // Image.network
            ), // Container
          ],
        ), // Column
      ), // Container
    ); // MaterialApp
  }
}
```

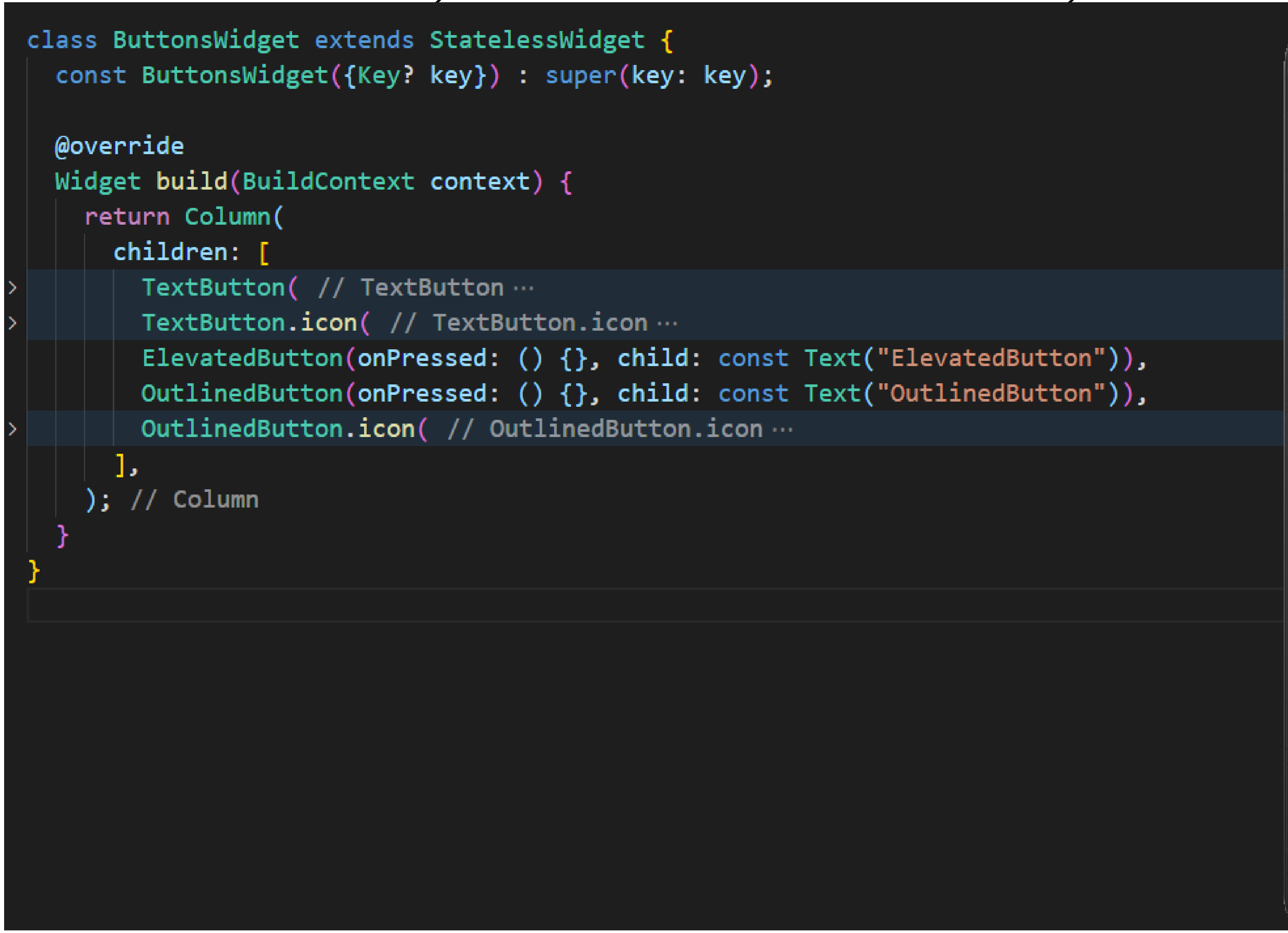
CircleAvatar and Shading



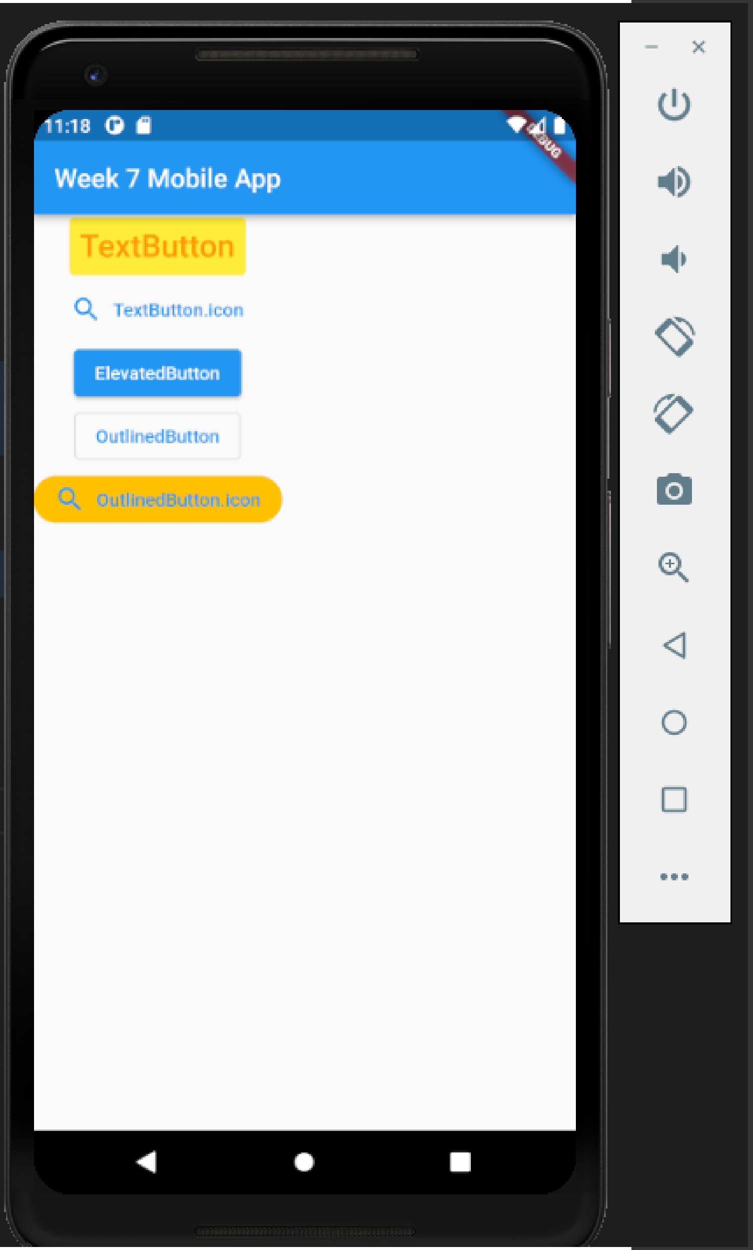
```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "App1",
      theme: ThemeData( // ThemeData ···
      home: Container(
        margin: EdgeInsets.all(50),
        child: Column(
          children: [
            Container(
              width: 300,
              height: 300,
              color: Colors.amber.shade800,
              child: const CircleAvatar(
                child: Text(
                  "Dog...",
                  style: TextStyle(fontSize: 36, color: Colors.blue),
                ), // Text
                backgroundImage: NetworkImage(
                  "https://cdn.codeblick.de/interquell-sb/1440x0/f/69110/1048x786/14c7621b83/129a5476-1-kopie-2.png",
                ), // NetworkImage
                backgroundColor: Colors.brown,
                radius: 50,
              ), // CircleAvatar
            ), // Container
          ],
        ), // Column
      ), // Container
    ); // MaterialApp
  }
}
```



Text Button, Elevated Button, Outlined Button

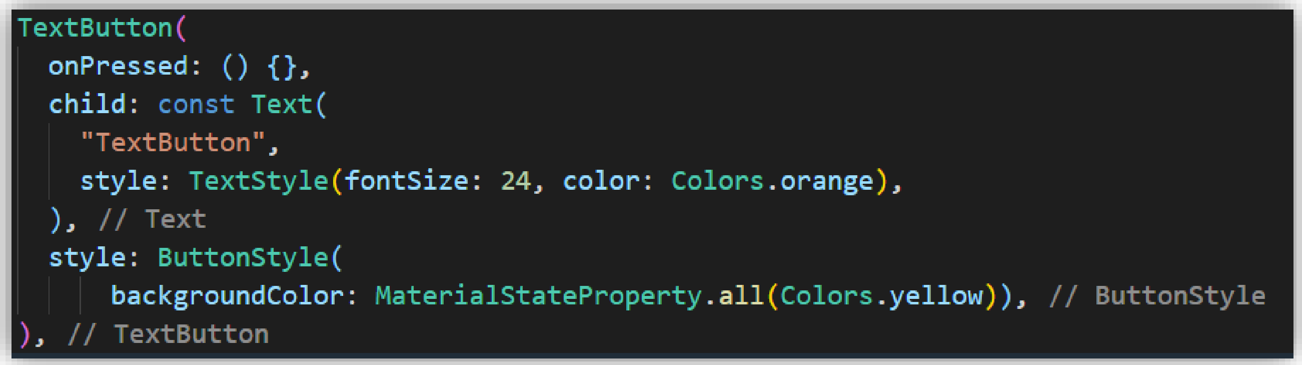


```
class ButtonsWidget extends StatelessWidget {
  const ButtonsWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        TextButton( // TextButton ...
        TextButton.icon( // TextButton.icon ...
        ElevatedButton(onPressed: () {}, child: const Text("ElevatedButton")),
        OutlinedButton(onPressed: () {}, child: const Text("OutlinedButton")),
        OutlinedButton.icon( // OutlinedButton.icon ...
      ],
    ); // Column
  }
}
```

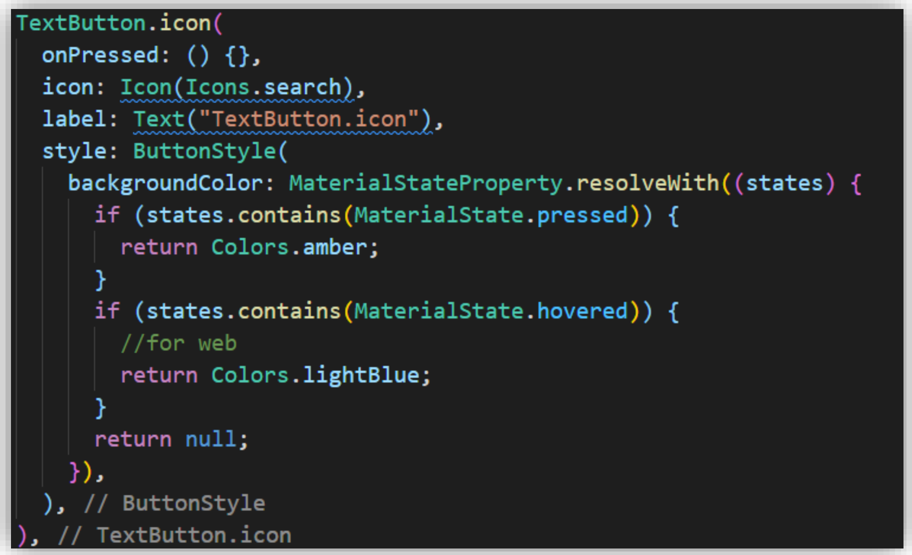


Details

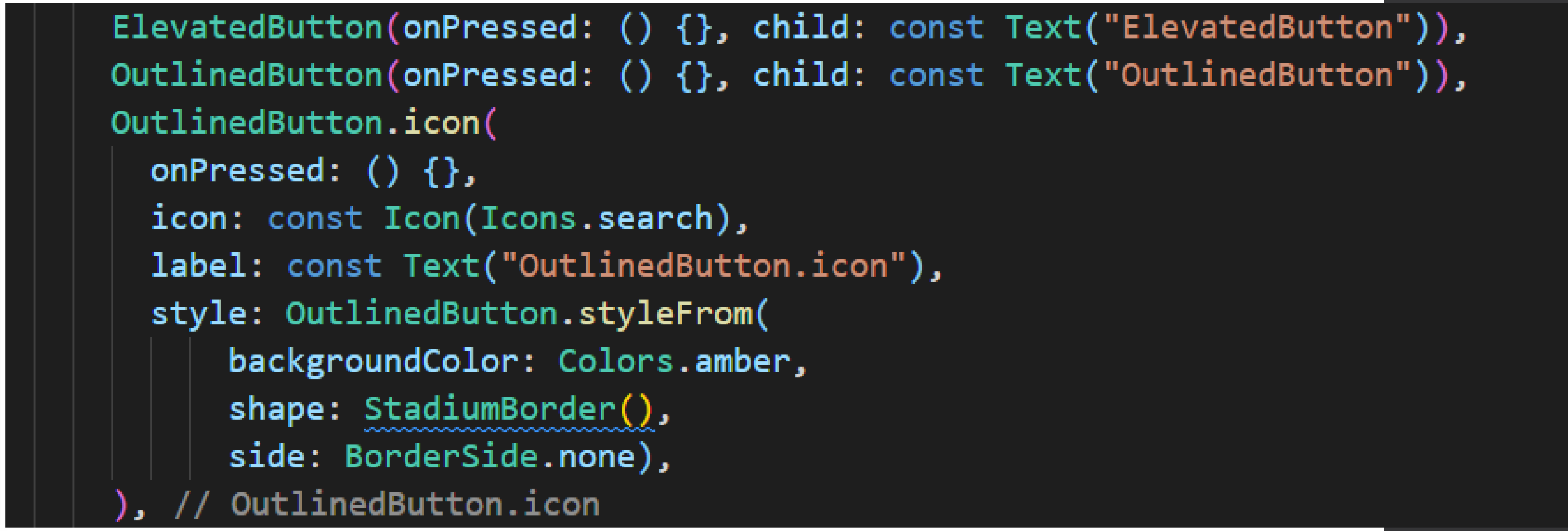
- ..



```
TextButton(
  onPressed: () {},
  child: const Text(
    "TextButton",
    style: TextStyle(fontSize: 24, color: Colors.orange),
  ), // Text
  style: ButtonStyle(
      backgroundColor: MaterialStateProperty.all(Colors.yellow)), // ButtonStyle
), // TextButton
```

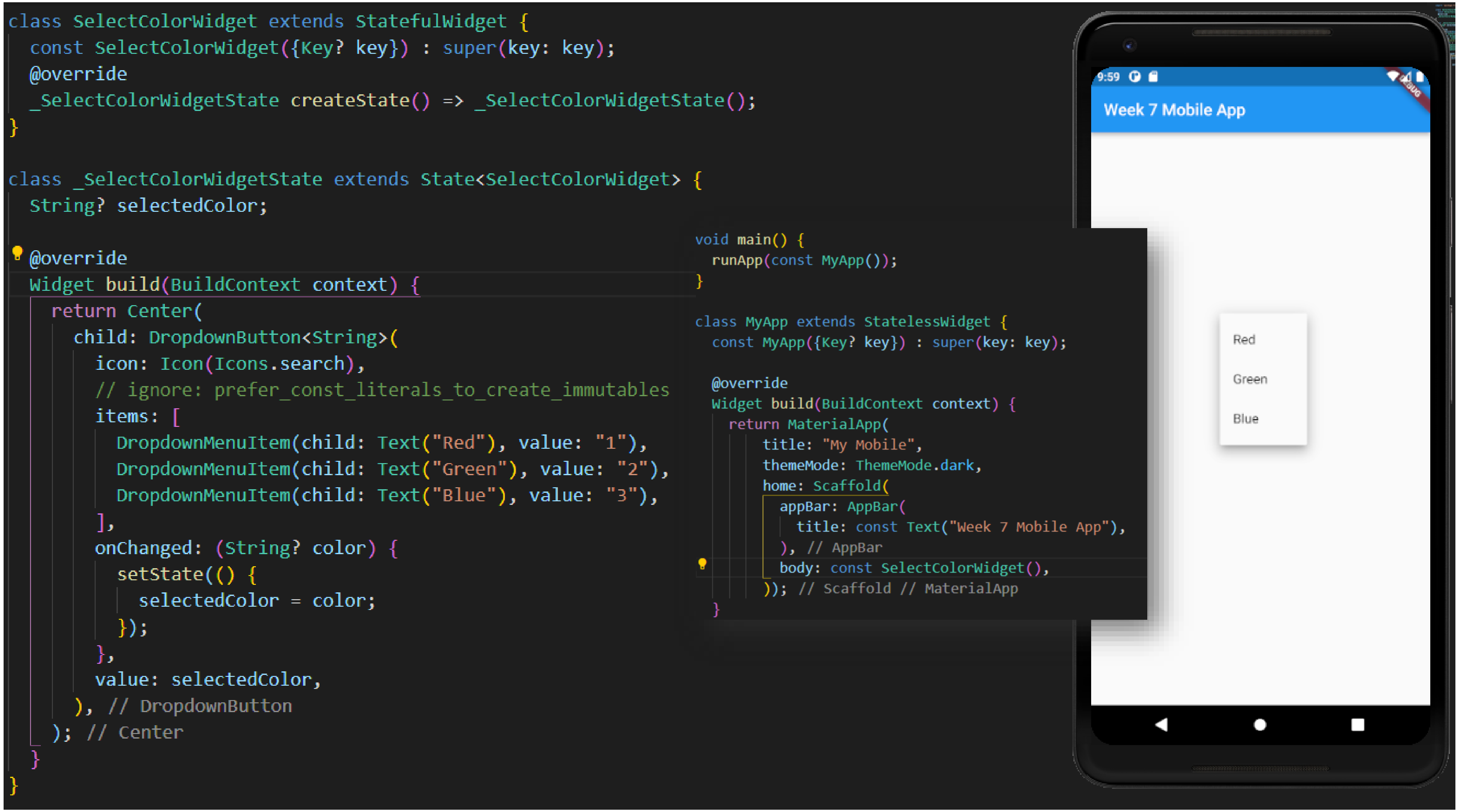


```
TextButton.icon(
  onPressed: () {},
  icon: Icon(Icons.search),
  label: Text("TextButton.icon"),
  style: ButtonStyle(
    backgroundColor: MaterialStateProperty.resolveWith((states) {
      if (states.contains(MaterialState.pressed)) {
        return Colors.amber;
      }
      if (states.contains(MaterialState.hovered)) {
        //for web
        return Colors.lightBlue;
      }
      return null;
    }),
  ), // ButtonStyle
), // TextButton.icon
```



```
ElevatedButton(onPressed: () {}, child: const Text("ElevatedButton")),
OutlinedButton(onPressed: () {}, child: const Text("OutlinedButton")),
OutlinedButton.icon(
  onPressed: () {},
  icon: const Icon(Icons.search),
  label: const Text("OutlinedButton.icon"),
  style: OutlinedButton.styleFrom(
      backgroundColor: Colors.amber,
      shape: StadiumBorder(),
      side: BorderSide.none),
), // OutlinedButton.icon
```

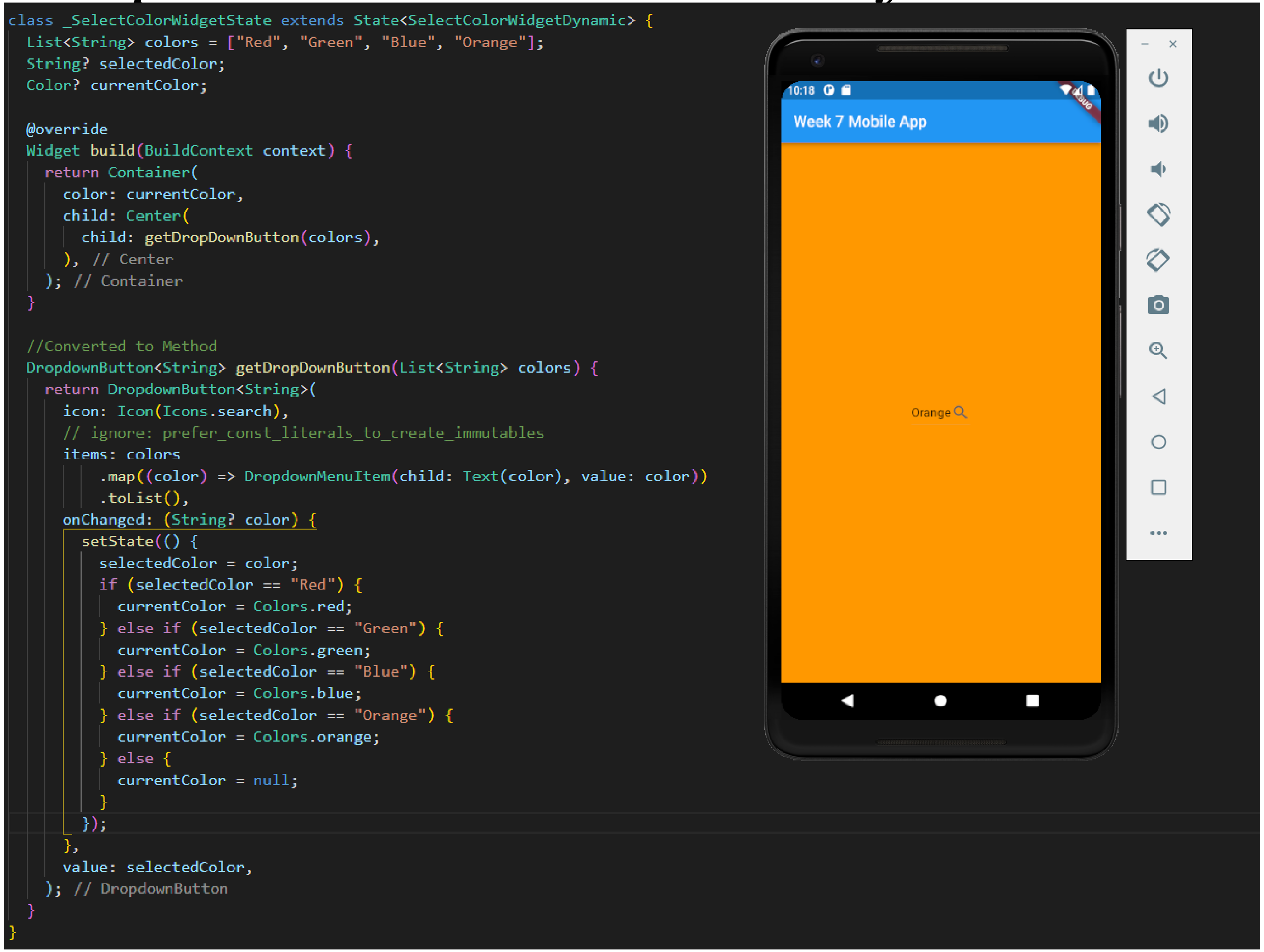
DropDownButton with Static Data



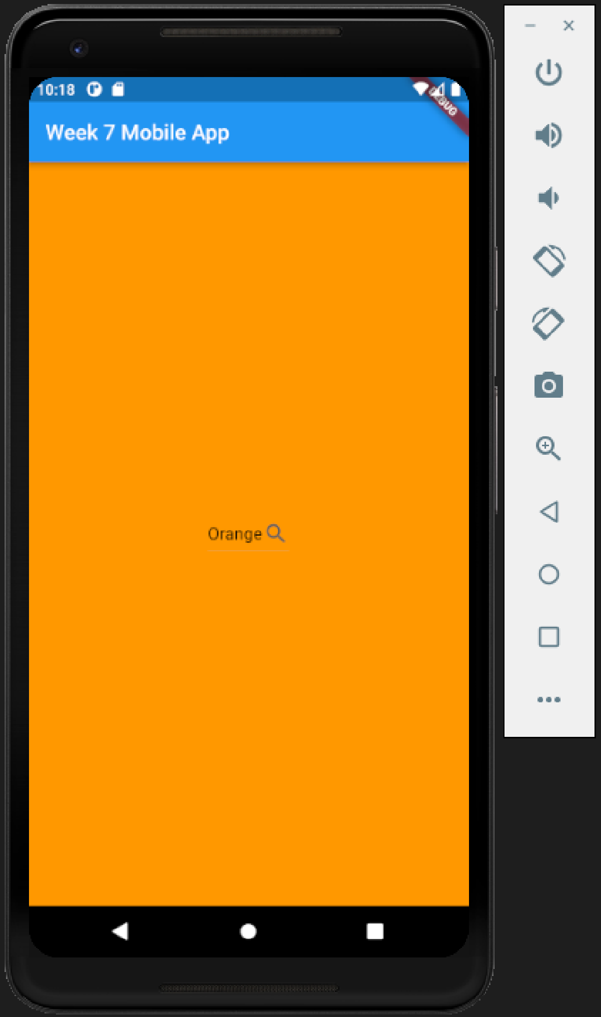
```
class SelectColorWidget extends StatefulWidget {
  const SelectColorWidget({Key? key}) : super(key: key);
  @override
  _SelectColorWidgetState createState() => _SelectColorWidgetState();
}

class _SelectColorWidgetState extends State<SelectColorWidget> {
  String? selectedColor;

  @override
  Widget build(BuildContext context) {
    return Center(
      child: DropdownButton<String>(
        icon: Icon(Icons.search),
        // ignore: prefer_const_literals_to_create_immutables
        items: [
          DropdownMenuItem(child: Text("Red"), value: "1"),
          DropdownMenuItem(child: Text("Green"), value: "2"),
          DropdownMenuItem(child: Text("Blue"), value: "3"),
        ],
        onChanged: (String? color) {
          setState(() {
            selectedColor = color;
          });
        },
        value: selectedColor,
      ), // DropdownButton
    ); // Center
  }
}
```

```
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "My Mobile",
      themeMode: ThemeMode.dark,
      home: Scaffold(
        appBar: AppBar(
          title: const Text("Week 7 Mobile App"),
        ), // AppBar
        body: const SelectColorWidget(),
      )); // Scaffold // MaterialApp
}
```

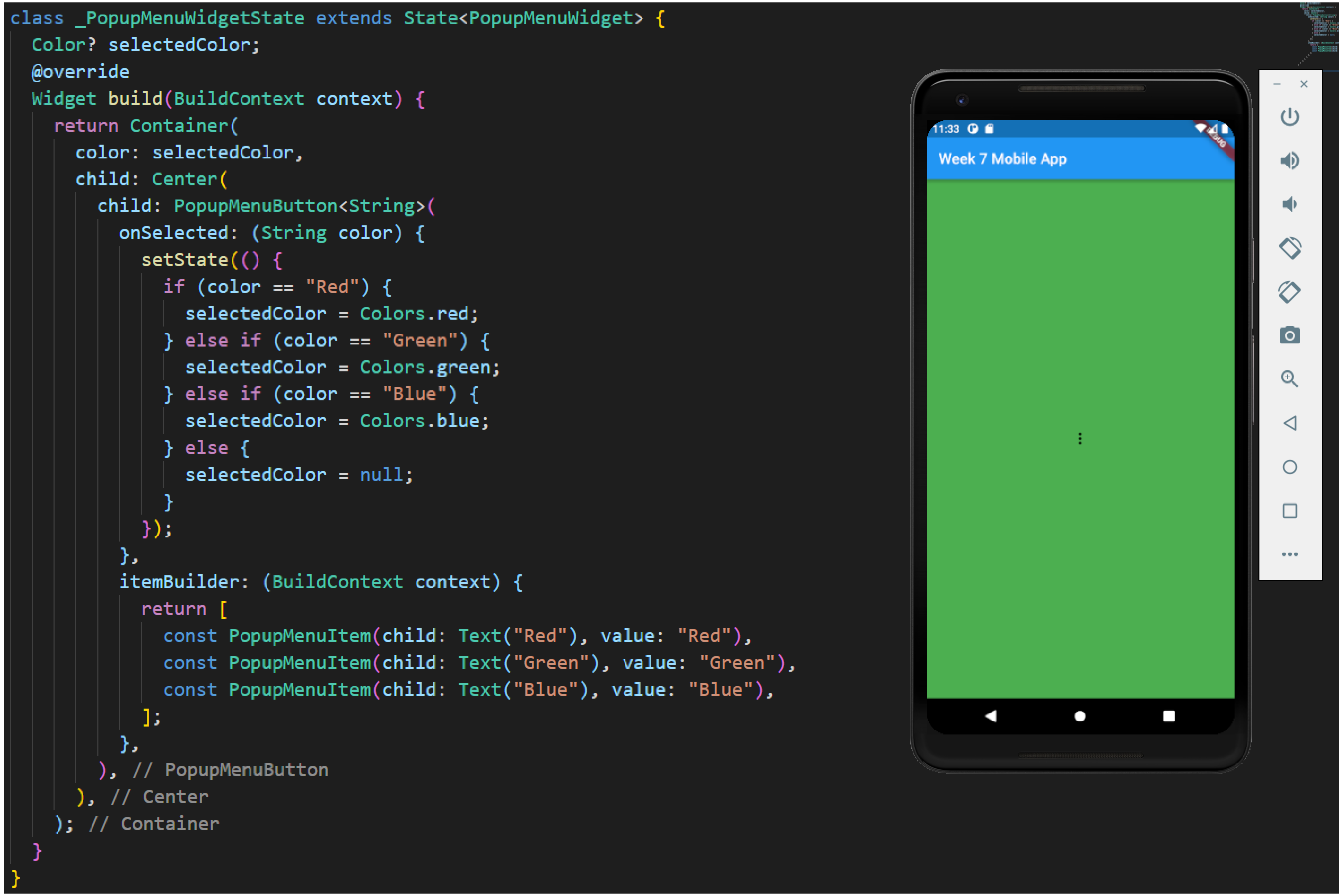
DropDownButton with Dynamic Data



```
class _SelectColorWidgetState extends State<SelectColorWidgetDynamic> {
  List<String> colors = ["Red", "Green", "Blue", "Orange"];
  String? selectedColor;
  Color? currentColor;

  @override
  Widget build(BuildContext context) {
    return Container(
      color: currentColor,
      child: Center(
        child: getDropDownButton(colors),
      ), // Center
    ); // Container
  }

  //Converted to Method
  DropdownButton<String> getDropDownButton(List<String> colors) {
    return DropdownButton<String>(
      icon: Icon(Icons.search),
      // ignore: prefer_const_literals_to_create_immutables
      items: colors
          .map((color) => DropdownMenuItem(child: Text(color), value: color))
          .toList(),
      onChanged: (String? color) {
        setState(() {
          selectedColor = color;
          if (selectedColor == "Red") {
            currentColor = Colors.red;
          } else if (selectedColor == "Green") {
            currentColor = Colors.green;
          } else if (selectedColor == "Blue") {
            currentColor = Colors.blue;
          } else if (selectedColor == "Orange") {
            currentColor = Colors.orange;
          } else {
            currentColor = null;
          }
        });
      },
      value: selectedColor,
    ); // DropdownButton
  }
}
```



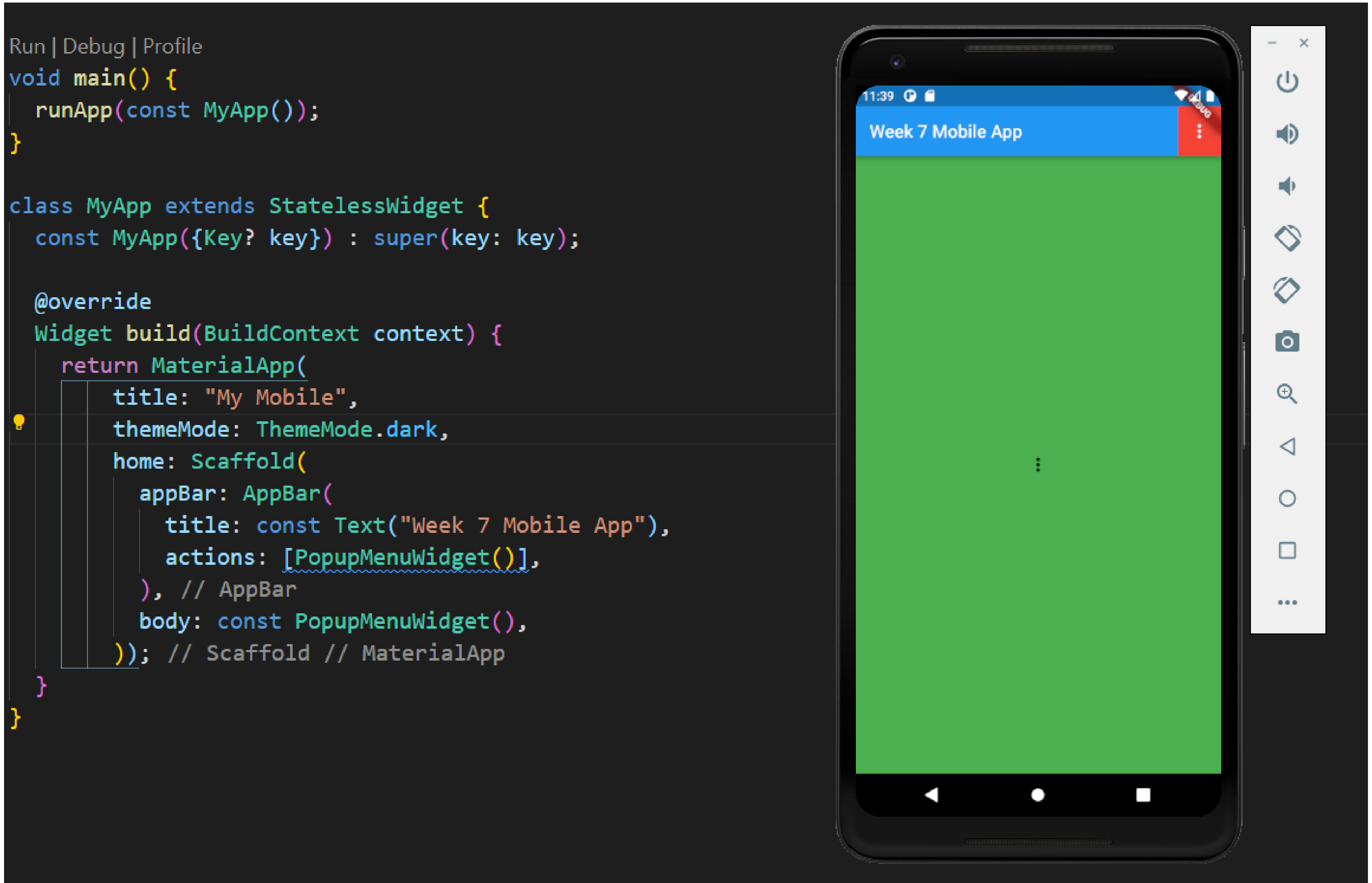
PopupMenuButton



```
class _PopupMenuWidgetState extends State<PopupMenuWidget> {
  Color? selectedColor;
  @override
  Widget build(BuildContext context) {
    return Container(
      color: selectedColor,
      child: Center(
        child: PopupMenuButton<String>(
          onSelected: (String color) {
            setState(() {
              if (color == "Red") {
                selectedColor = Colors.red;
              } else if (color == "Green") {
                selectedColor = Colors.green;
              } else if (color == "Blue") {
                selectedColor = Colors.blue;
              } else {
                selectedColor = null;
              }
            });
          },
          itemBuilder: (BuildContext context) {
            return [
              const PopupMenuItem(child: Text("Red"), value: "Red"),
              const PopupMenuItem(child: Text("Green"), value: "Green"),
              const PopupMenuItem(child: Text("Blue"), value: "Blue"),
            ];
          },
        ), // PopupMenuButton
      ), // Center
    ); // Container
  }
}
```

AppBar menu with PopupMenuButton

- The same widget can be used for AppBar menu



```
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        title: "My Mobile",
        themeMode: ThemeMode.dark,
        home: Scaffold(
          appBar: AppBar(
            title: const Text("Week 7 Mobile App"),
            actions: [PopupMenuWidget()],
          ), // AppBar
          body: const PopupMenuWidget(),
        )); // Scaffold // MaterialApp
  }
}
```

• • •

- ••

• • •

- ••

• • •

- ••

• • •

- ..

• • •

- ••

• • •

- ••

• • •

- • •

• • •

- ••

• • •

- ••

• • •

- ••

• • •

- ••

• • •

- • •