



#Flutter Notes

Introduction

Directory structure

- android –
 - where Android-related files are stored.
 - If you've done any sort of cross-platform mobile app development before, this, along with the ios folder should be pretty familiar.
- ios –
 - where iOS-related files are stored.
- test –
 - this is where you put the unit testing code for the app.



- **lib** –

- this is where you'll be working on most of the time.
- By default, it contains a main.dart file, this is the entry point file of the Flutter app.

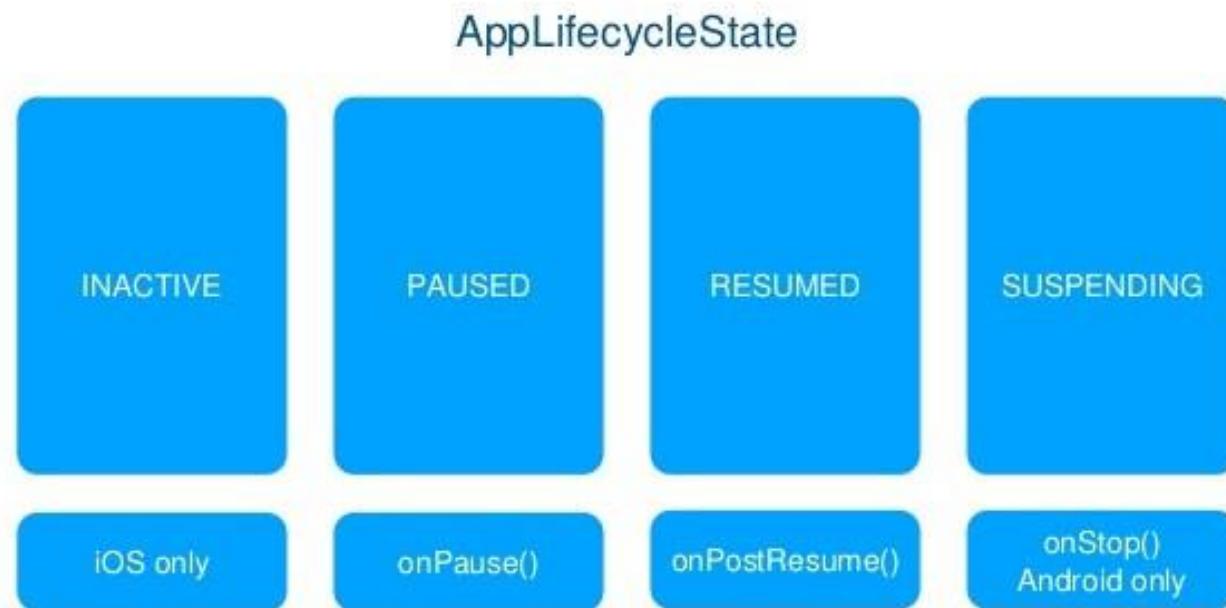
- **pubspec.yaml** –

- this file defines the version and build number of your app.
- It's also where you define your dependencies.
- If you're coming from a web development background, this file has the same job description as the package.json file so you can define the external packages (from the Dart packages website) you want to use in here.



Flutter App Lifecycle

Life cycle



<https://api.flutter.dev/flutter/dart-ui/AppLifecycleState-class.html>



■ **inactive**

- The application is in an inactive state and is not receiving user input. This event only works on iOS, as there is no equivalent event to map to on Android

■ **paused**

- The application is not currently visible to the user, not responding to user input, and running in the background. This is equivalent to onPause() in Android



■ **resumed**

- The application is visible and responding to user input. This is equivalent to `onPostResume()` in Android

■ **suspending**

- The application is suspended momentarily. This is equivalent to `onStop` in Android; it is not triggered on iOS as there is no equivalent event to map to on iOS



Example

```
@override  
void initState() {  
    WidgetsBinding.instance.addObserver(this);  
    super.initState();  
}
```

```
@override  
void dispose() {  
    WidgetsBinding.instance.removeObserver(this)  
;  
    super.dispose();  
}
```



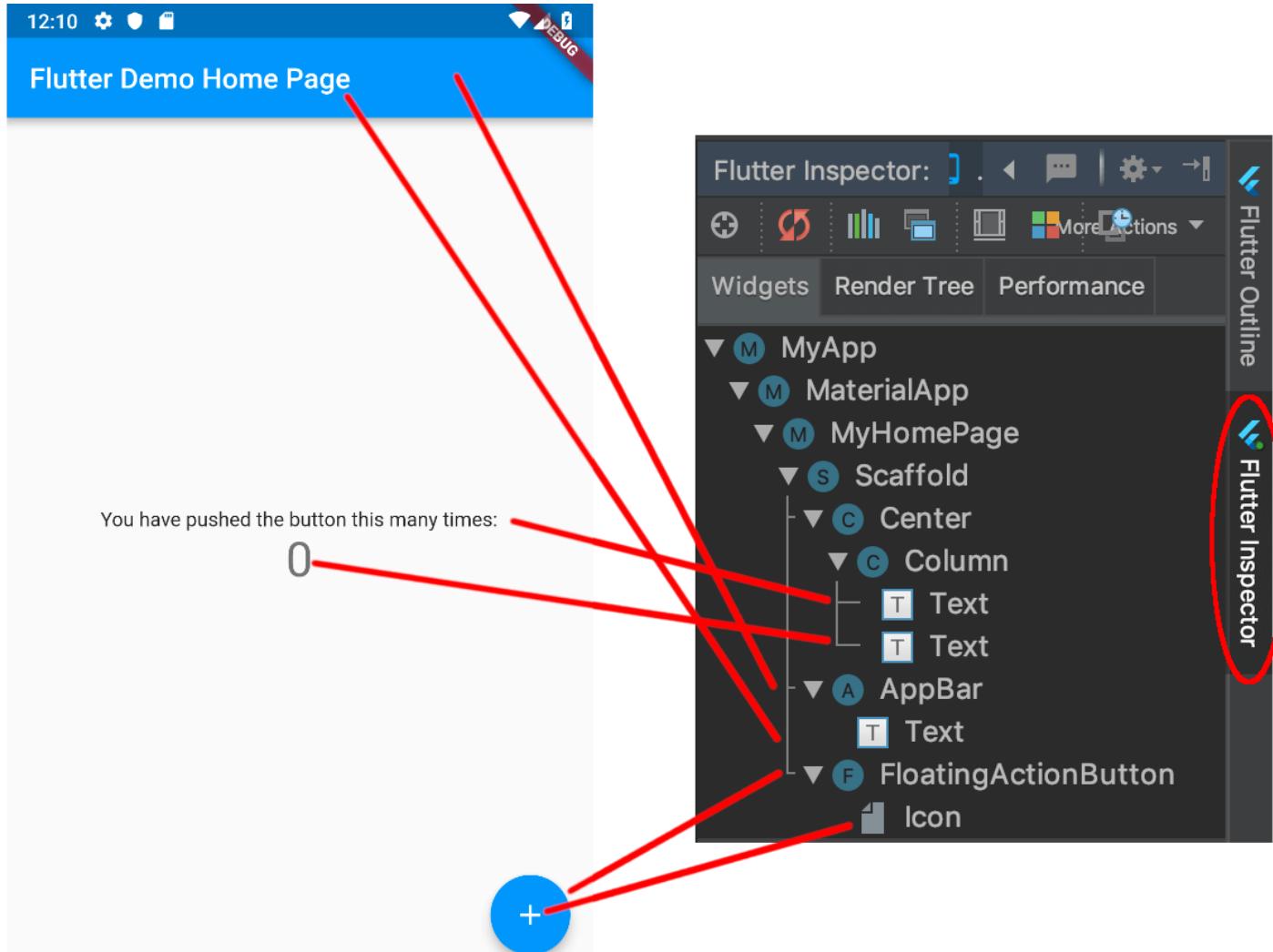
Widget

Widget

- Widget is a basic building block on flutter code. For example, button is one of widget component within apps. So widget can create "structural element","stylistic element(font / color)", and "layout" etc.



Widget trees





‘Everything is a widget’



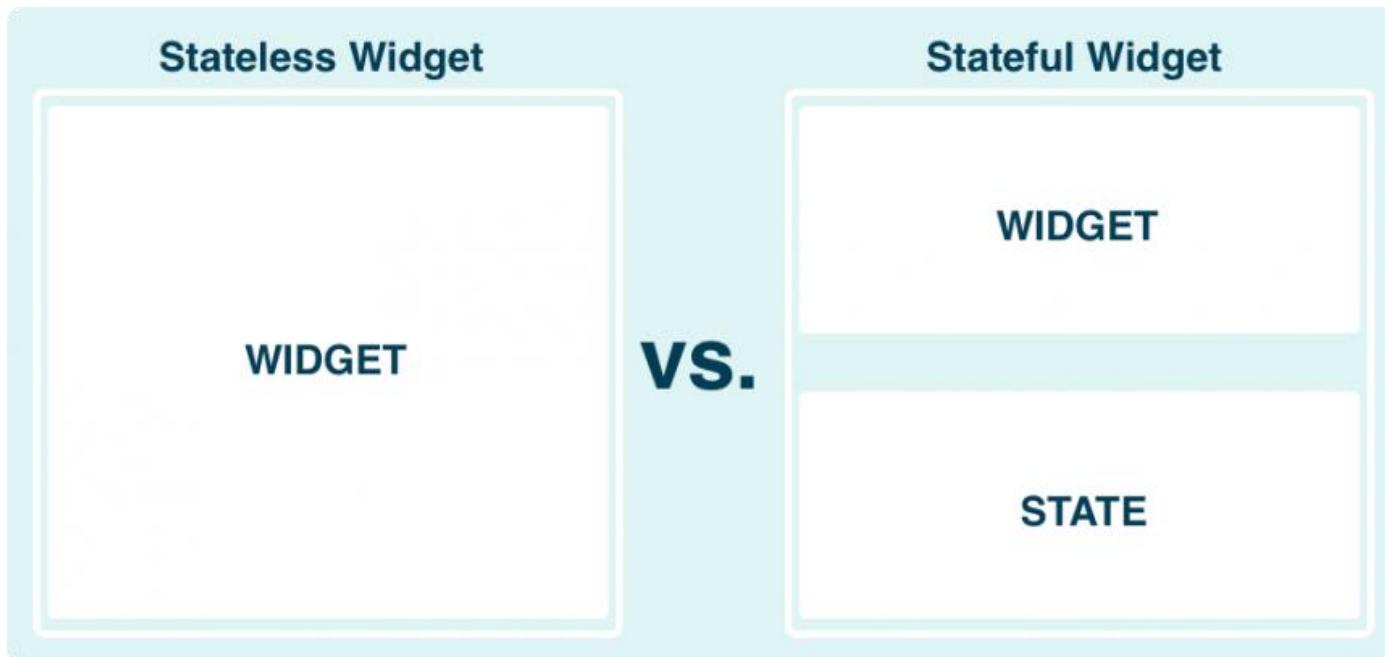
Akash Technolabs

www.akashsir.com



Types of Widget

- StatelessWidget StatelessWidget is immutable once it is built.
- StatefulWidget StatefulWidget is changed dynamically and can be redrawn within life cycle



Stateless Widget

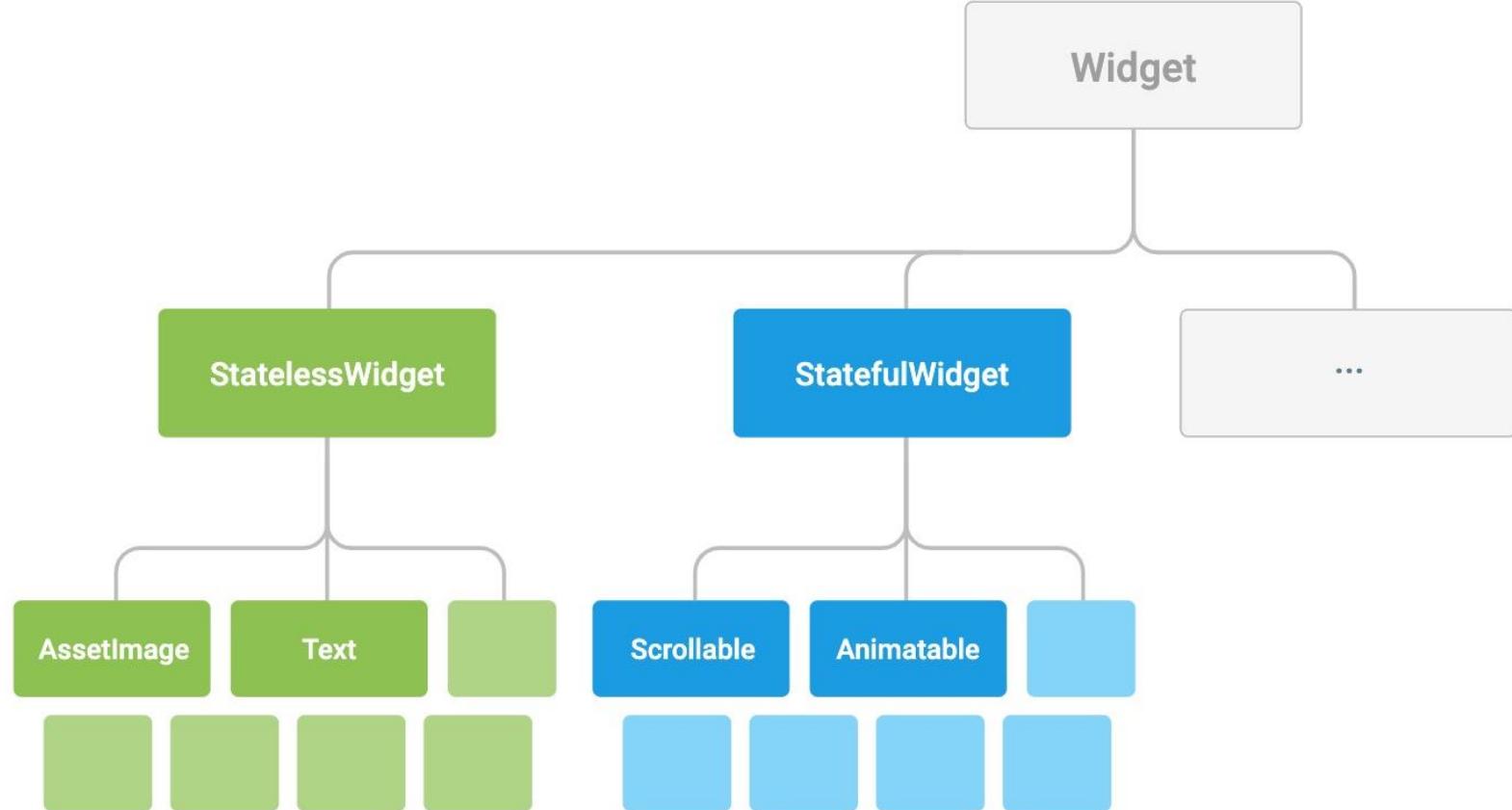
- Stateless widgets are widgets that don't store any state. That is, they don't store values that might change.
- For example, an Icon is stateless; you set the icon image when you create it and then it doesn't change any more.



Statefull Widget

- That means it can keep track of changes and update the UI based on those changes.
- A stateful widget is useful for something like a checkbox. When a user clicks it, the check state is updated





Element of Widget

- Flutter provide basic element of widget.
 - Text
 - Row, Column
 - Stack
 - Container



Element of Widget

- **Text** is widget that create styled text element in the screen.
- **Row, Column** is widget that generate flexible layout for Row(horizontal) and Column(vertical) element in the screen.
- **Stack** is widget that stack widget on the top of each.
- **Container** is widget that create rectangular visual element. Container can be decorated with "BoxDecoration" such as a background, a border, or a shadow.

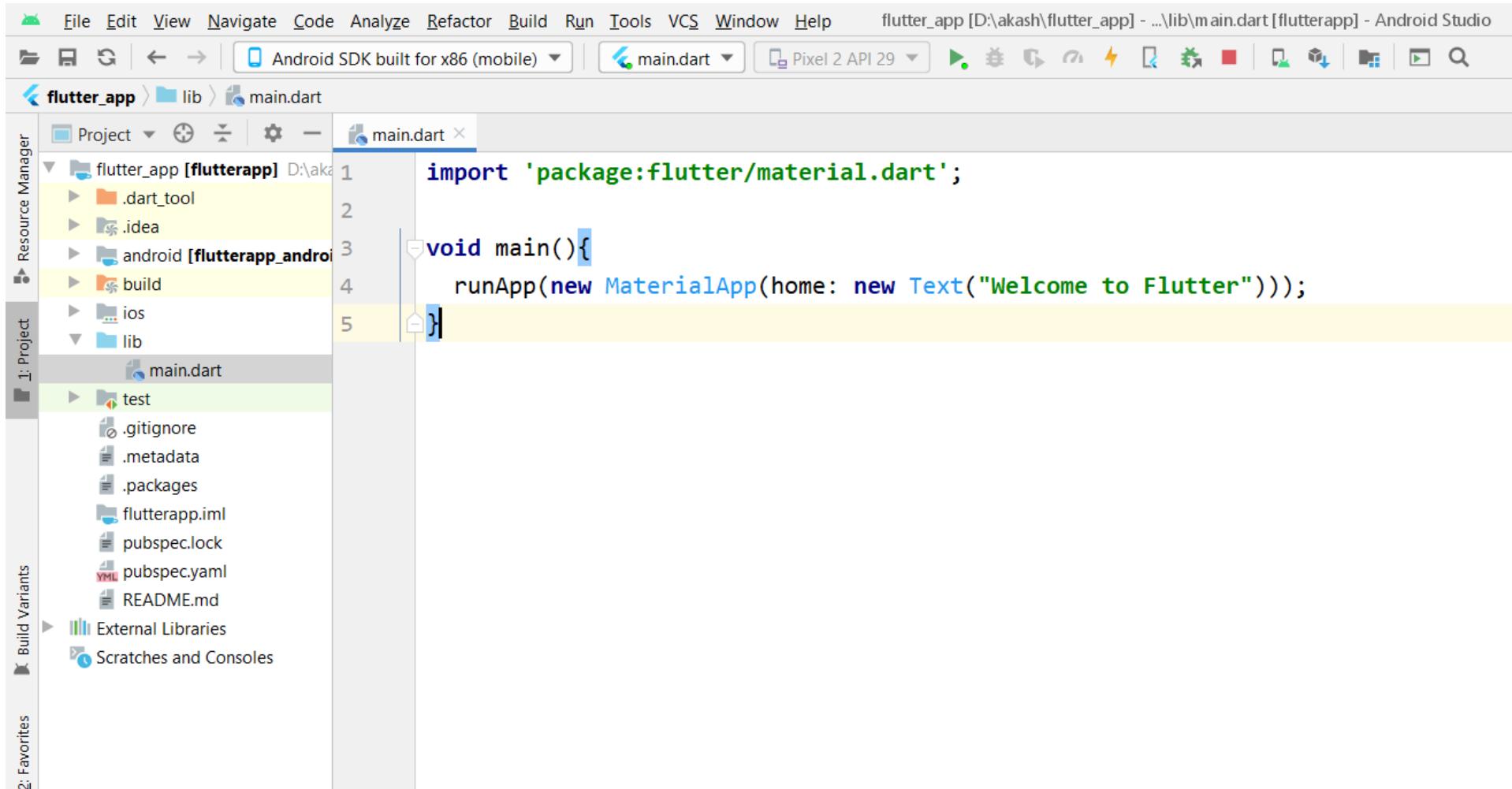


Let's Write Code

- Open the main.dart file. It's in the lib folder in your project outline.
- Delete all the text in this file and replace it with
 - void main() {}
- If you hot reload your app now it should be a blank screen.
- The main() function is the starting point for every Flutter app.



Simple Material App



The screenshot shows the Android Studio interface with the following details:

- Toolbar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** flutter_app [D:\akash\flutter_app] - ...\\lib\\main.dart [flutterapp] - Android Studio.
- File Bar:** main.dart, Pixel 2 API 29.
- Tool Buttons:** Back, Forward, Refresh, Run, Stop, Build, Clean, Sync, Find, Replace, Open, Save, Import, Export, Settings, Help.
- Project Tree (Resource Manager):**
 - Project: flutter_app [flutterapp] D:\akash\flutter_app
 - Build Variants: 2: Favorites
 - Build Variants: 2: Favorites
 - Flutter: .dart_tool, .idea
 - Android: android [flutterapp_android], build, ios
 - Lib: lib (selected), main.dart, test, .gitignore, .metadata, .packages, flutterapp.iml, pubspec.lock, pubspec.yaml, README.md, External Libraries, Scratches and Consoles
- Code Editor (main.dart):**

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

void main(){
    runApp(new MaterialApp(home: new Text("Welcome to Flutter")));
}
```



The screenshot shows the Android Studio interface with a Flutter project named "flutter_app". The "lib/main.dart" file is open, containing the following code:

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

void main(){
  runApp(new MaterialApp(home: new Text("Welcome to Flutter")));
}
```

The "Run" tab is selected, and the "Console" tab shows the output of a hot reload:

```
Performing hot reload...
Syncing files to device Android SDK built for x86...
Reloaded 0 of 478 libraries in 118ms.
```

The right side of the screen displays a virtual iPhone X simulator with the text "welcome" (in red) and "to Flutter" (in green) visible on its screen.



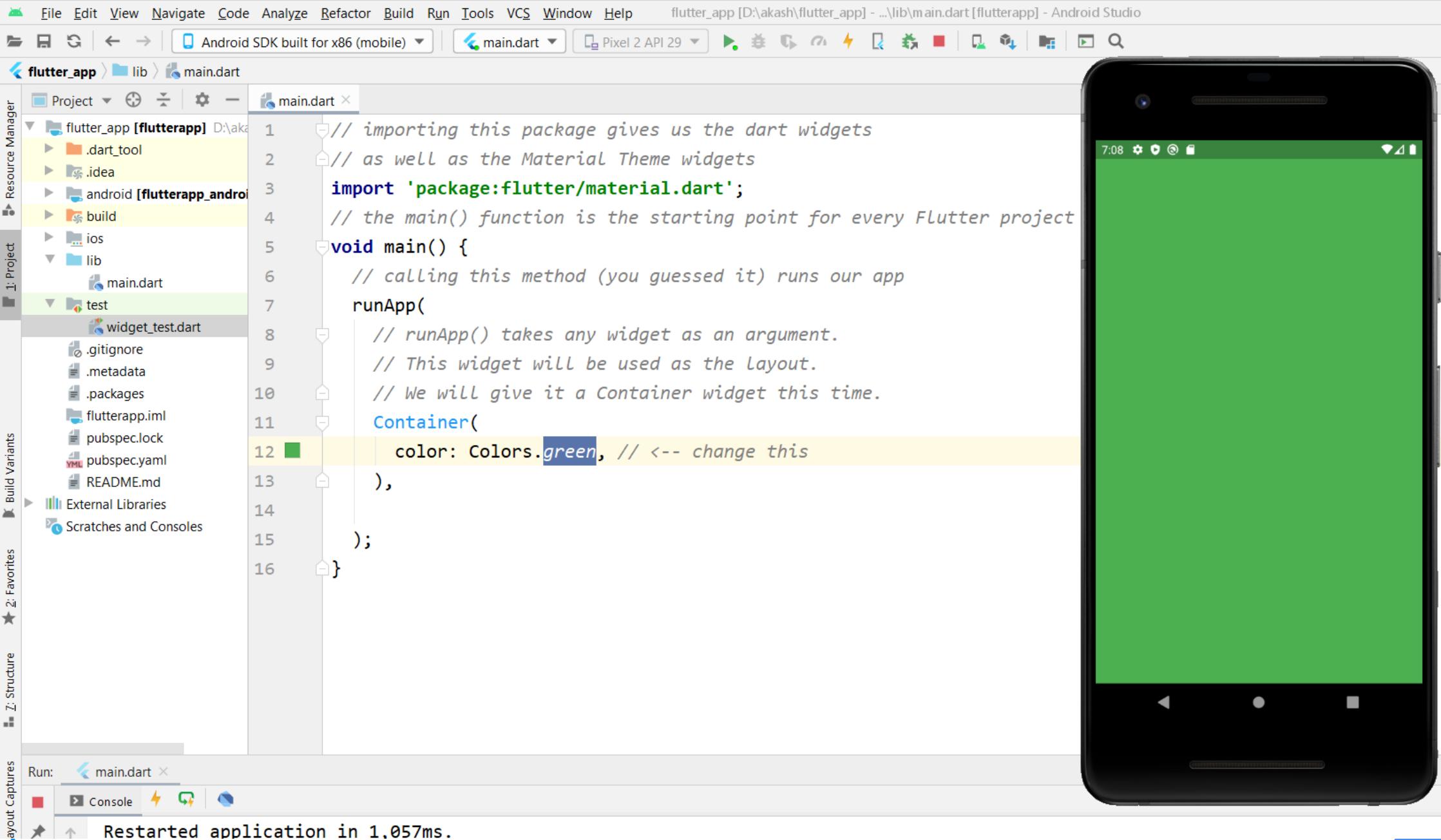
Container widget

```
// importing this package gives us the dart widgets
// as well as the Material Theme widgets
import 'package:flutter/material.dart';

// the main() function is the starting point for every Flutter project
void main() {

    // calling this method (you guessed it) runs our app
    runApp(
        // runApp() takes any widget as an argument.
        // This widget will be used as the layout.
        // We will give it a Container widget this time.
        Container(
            color: Colors.green, // <-- change this
        ),
    );
}
```





File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help flutter_app [D:\akash\flutter_app] - ...\\lib\\main.dart [flutterapp] - Android Studio

Android SDK built for x86 (mobile) main.dart Pixel 2 API 29

flutter_app lib main.dart

Project Manager Resource Manager 1: Project Build Variants Favorites Z: Structure Run Captures

Flutter App

main.dart

```
// importing this package gives us the dart widgets
// as well as the Material Theme widgets
import 'package:flutter/material.dart';
// the main() function is the starting point for every Flutter project
void main() {
    // calling this method (you guessed it) runs our app
    runApp(
        // runApp() takes any widget as an argument.
        // This widget will be used as the layout.
        // We will give it a Container widget this time.
        Container(
            color: Colors.green, // <-- change this
        ),
    );
}
```

Run: main.dart

Console Restarted application in 1.057ms.



Material Color

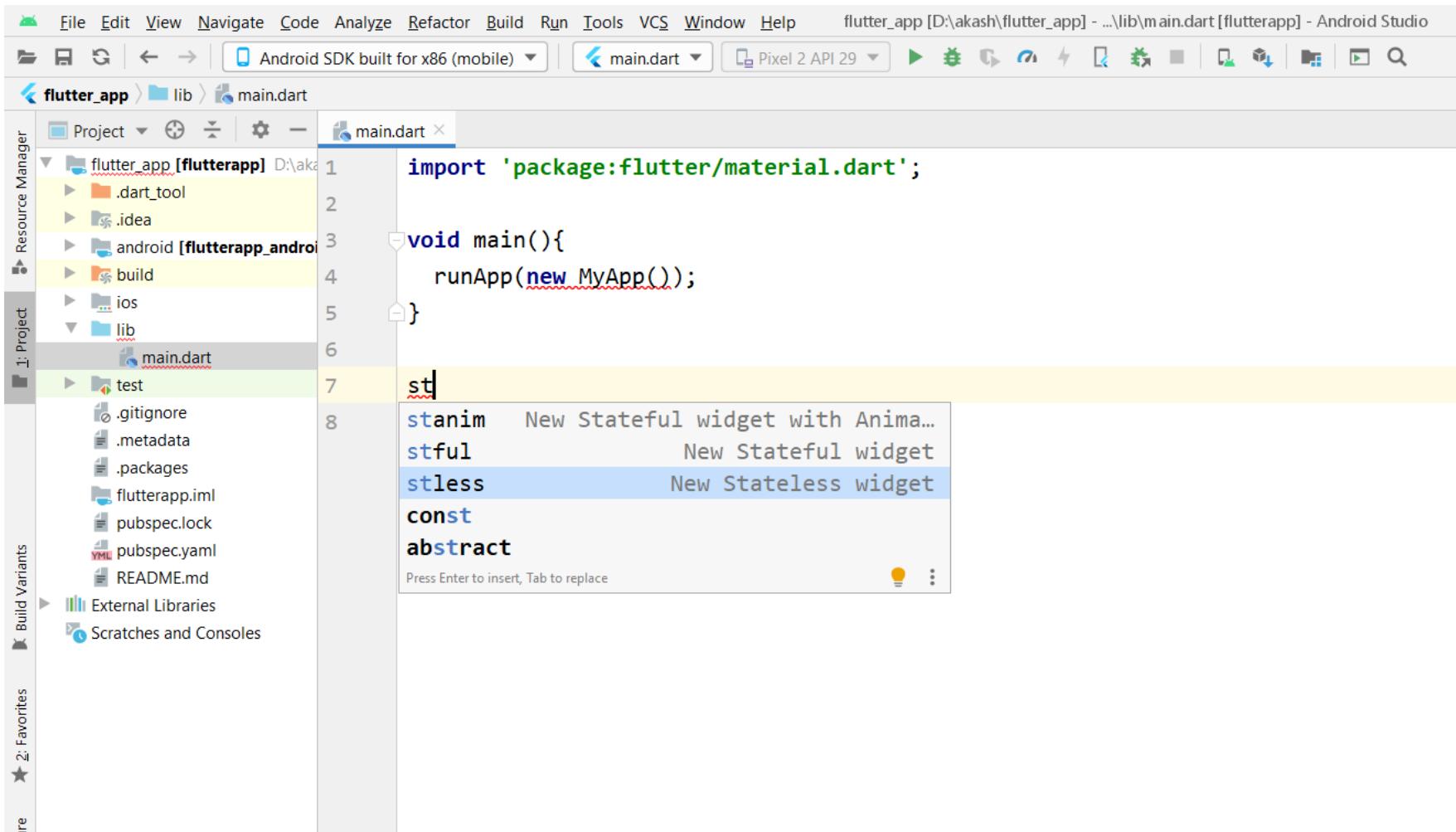
- Flutter already has many MaterialColor available
- Most swatches have colors from 100 to 900 in increments of one hundred, plus the color 50. The smaller the number, the lighter the color.
- The greater the number, the darker the color. The accent swatches (e.g. redAccent) only have the values 100, 200, 400, and 700.
- <https://material-ui.com/customization/color/>

Colors.yellow[50]	0xFFFFFDE7
Colors.yellow[100]	0xFFFFF9C4
Colors.yellow[200]	0xFFFFF59D
Colors.yellow[300]	0xFFFFF176
Colors.yellow[400]	0xFFFFEE58
Colors.yellow	0xFFFFEB3B
Colors.yellow[600]	0xFFFDD835
Colors.yellow[700]	0xFFFBC02D
Colors.yellow[800]	0xFFF9A825
Colors.yellow[900]	0xFFF57F17



StateLess Widget

StateLess



The screenshot shows the Android Studio interface with the project 'flutter_app' open. The main.dart file is selected in the editor. A code completion dropdown is open at line 7, showing suggestions for 'st'. The suggestion 'stless' is highlighted, indicating it's the correct spelling for a Stateless widget.

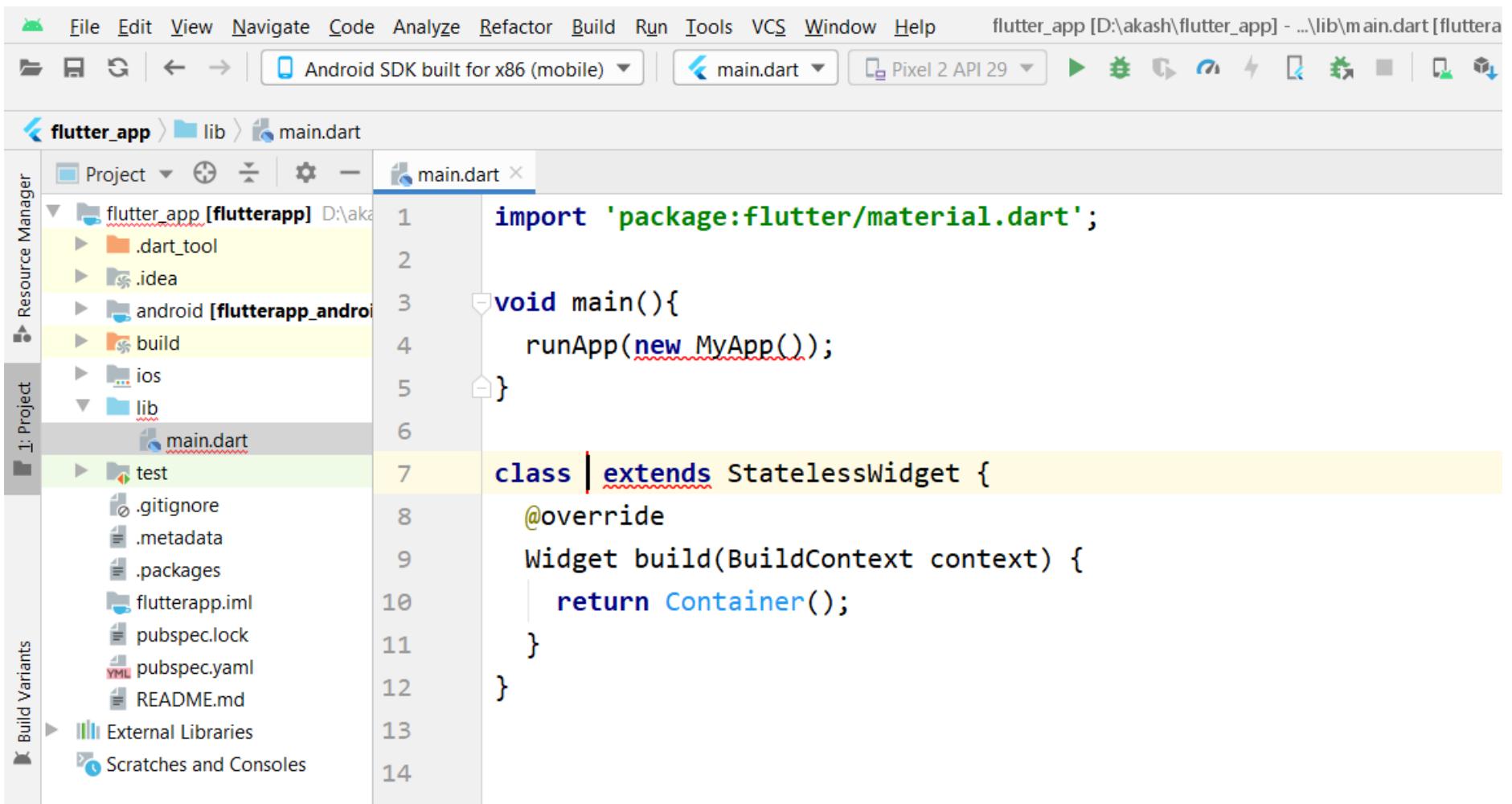
```
import 'package:flutter/material.dart';

void main(){
    runApp(new MyApp());
}

st
stanim  New StatefulWidget with Animation...
stful    New StatefulWidget
stless   New StatelessWidget
const
abstract
Press Enter to insert, Tab to replace
```



Specify WidgetName



The screenshot shows the Android Studio interface with the project 'flutter_app' open. The 'main.dart' file is selected in the editor. The code defines a simple StatelessWidget named 'MyApp'. The class definition starts with 'class | extends StatelessWidget {'. The cursor is positioned at the start of the class definition, indicated by a vertical line and a red underline.

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

void main(){
    runApp(new MyApp());
}

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



MyApp

The screenshot shows the Android Studio IDE interface with the following details:

- Toolbar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** flutter_app [flutterapp] D:\akash\flutter_app - ...\\lib\\main.d
- Toolbars:** Android SDK built for x86 (mobile), main.dart, Pixel 2 API 29.
- Code Editor:** The main.dart file is open, showing the following Dart code:

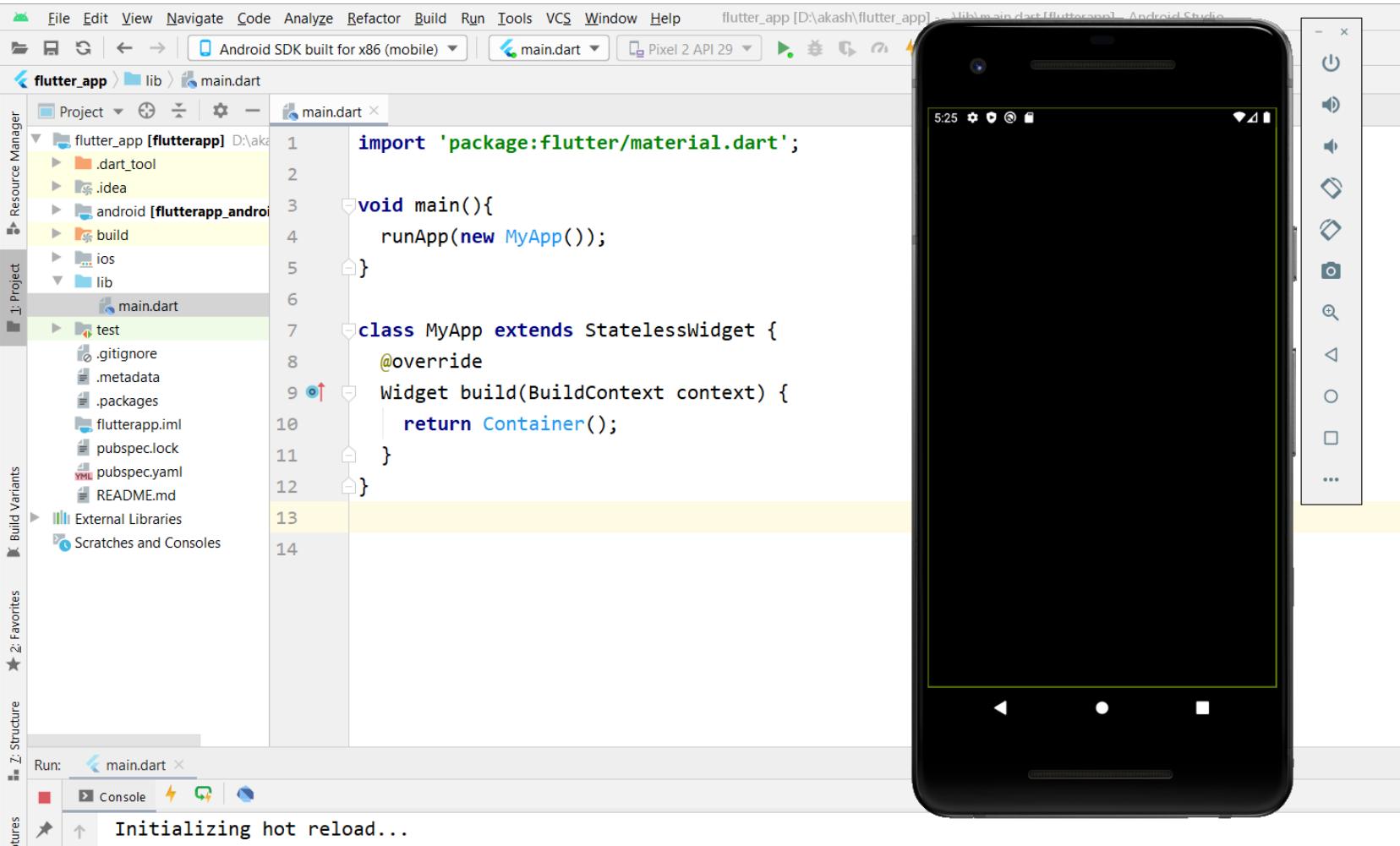
```
import 'package:flutter/material.dart';

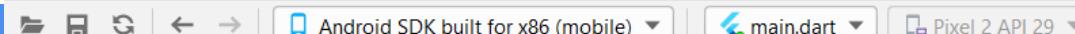
void main(){
    runApp(new MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Container();
    }
}
```
- Project Structure:** The Project Manager shows the following structure:
 - flutter_app [flutterapp]
 - .dart_tool
 - .idea
 - android [flutterapp_android]
 - build
 - ios
 - lib
 - main.dart
 - test
 - .gitignore
 - .metadata
 - .packages
 - flutterapp.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Build Variants:** Build Variants tab is visible.
- Bottom Navigation:** Ites tab is selected.



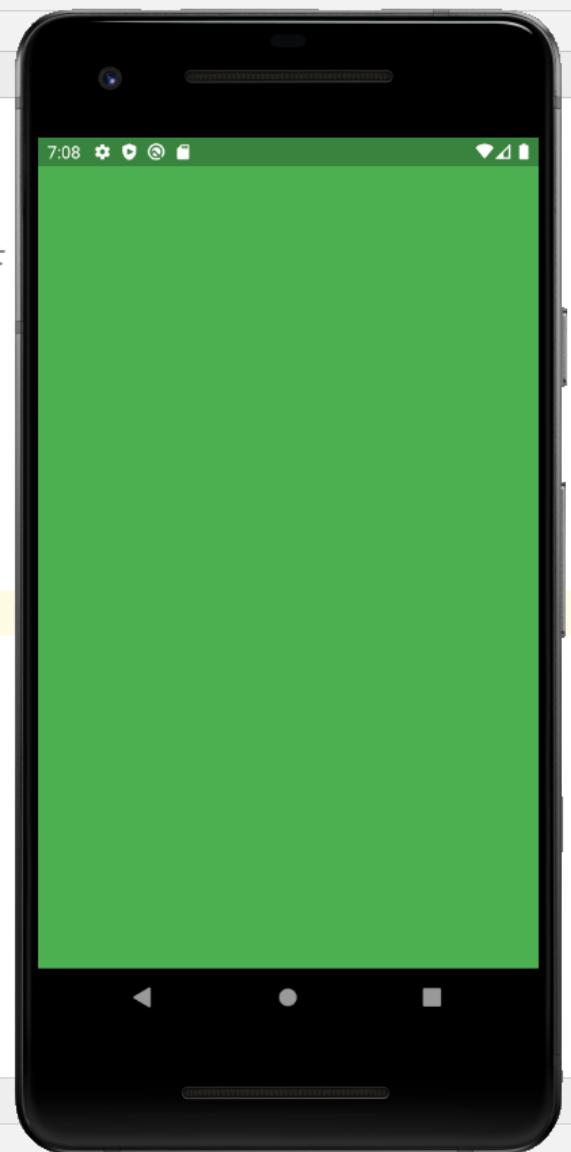
Container Demo



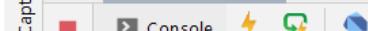


flutter_app > lib > main.dart

```
main.dart x
1 // importing this package gives us the dart widgets
2 // as well as the Material Theme widgets
3 import 'package:flutter/material.dart';
4 // the main() function is the starting point for every Flutter project
5 void main() {
6     // calling this method (you guessed it) runs our app
7     runApp(
8         // runApp() takes any widget as an argument.
9         // This widget will be used as the Layout.
10        // We will give it a Container widget this time.
11        Container(
12            color: Colors.green, // <-- change this
13        ),
14    );
15 }
16 }
```



Run: main.dart x



Restarted application in 1.057ms.



Akash Technolabs

www.akashsir.com

What is Scaffold?

- A Scaffold Widget provides a framework which implements the basic material design visual layout structure of the flutter app.
- It provides APIs for showing drawers, snack bars and bottom sheets.



appBar

- An *appBar* is to display at the top of the scaffold it's one of the primary content of Scaffold without which a scaffold widget is incomplete.
- It defines what has to be displayed at the top of the screen. appBar uses the AppBar widget properties through Scaffold.appBar.
- This AppBar widget itself has various properties like
 - title,
 - elevation,
 - brightness etc to name a few.



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help flutter_app [D:\akash\flutter_app]

Android SDK built for x86 (mobile) main.dart Pixel 2 API 29

flutter_app lib main.dart

Resource Manager Project Manager

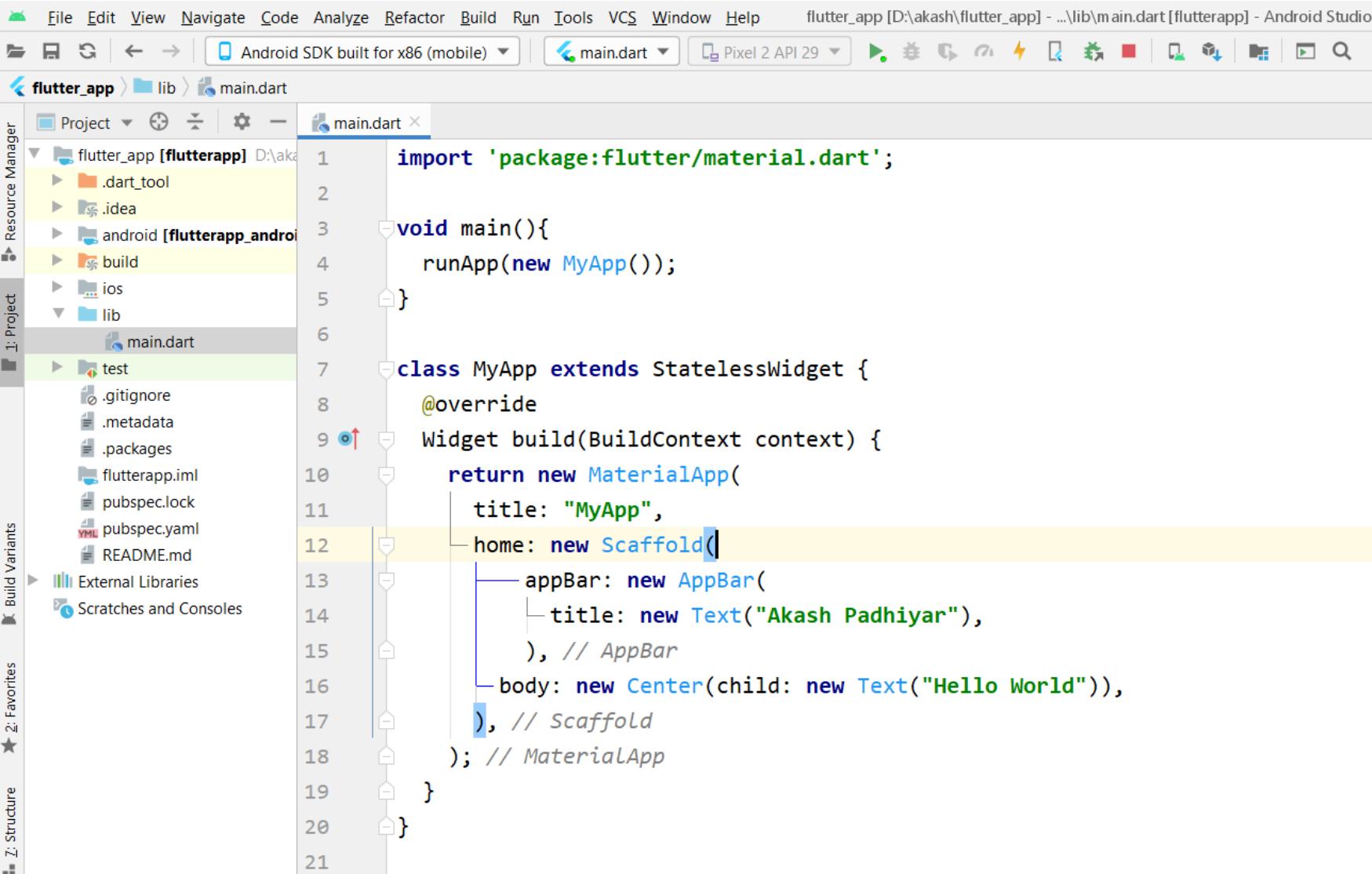
Project flutter_app [flutterapp] D:\akash\flutter_app .dart_tool .idea android [flutterapp_android] build ios lib main.dart test widget_test.dart .gitignore .metadata .packages flutterapp.iml pubspec.lock pubspec.yaml README.md External Libraries Scratches and Consoles

1 import 'package:flutter/material.dart';
2 void main(){
3 runApp(new MyApp());
4 }
5 class MyApp extends StatelessWidget {
6 @override
7 Widget build(BuildContext context) {
8 return new MaterialApp(
9 title: "MyApp",
10 home: new Scaffold(
11 appBar: new AppBar(
12 title: new Text("Akash Padhiyar"),
13 centerTitle: true,
14), // AppBar
15 body: Text("Welcome to Flutter"),
16), // Scaffold
17); // MaterialApp
18 }
19 }
20 }

7:38 Akash Padhiyar
Welcome to Flutter



Scaffold Widget



The screenshot shows the Android Studio interface with the project 'flutter_app' open. The 'lib/main.dart' file is the active editor. The code demonstrates the use of the Scaffold widget within a StatelessWidget.

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

void main(){
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "MyApp",
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text("Akash Padhiyar"),
        ), // AppBar
        body: new Center(child: new Text("Hello World")),
      ), // Scaffold
    ); // MaterialApp
}
}
```



Stateless Widget Output

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "flutter_app" and contains "lib/main.dart" which is currently selected.
- Code Editor:** The code in "main.dart" is as follows:

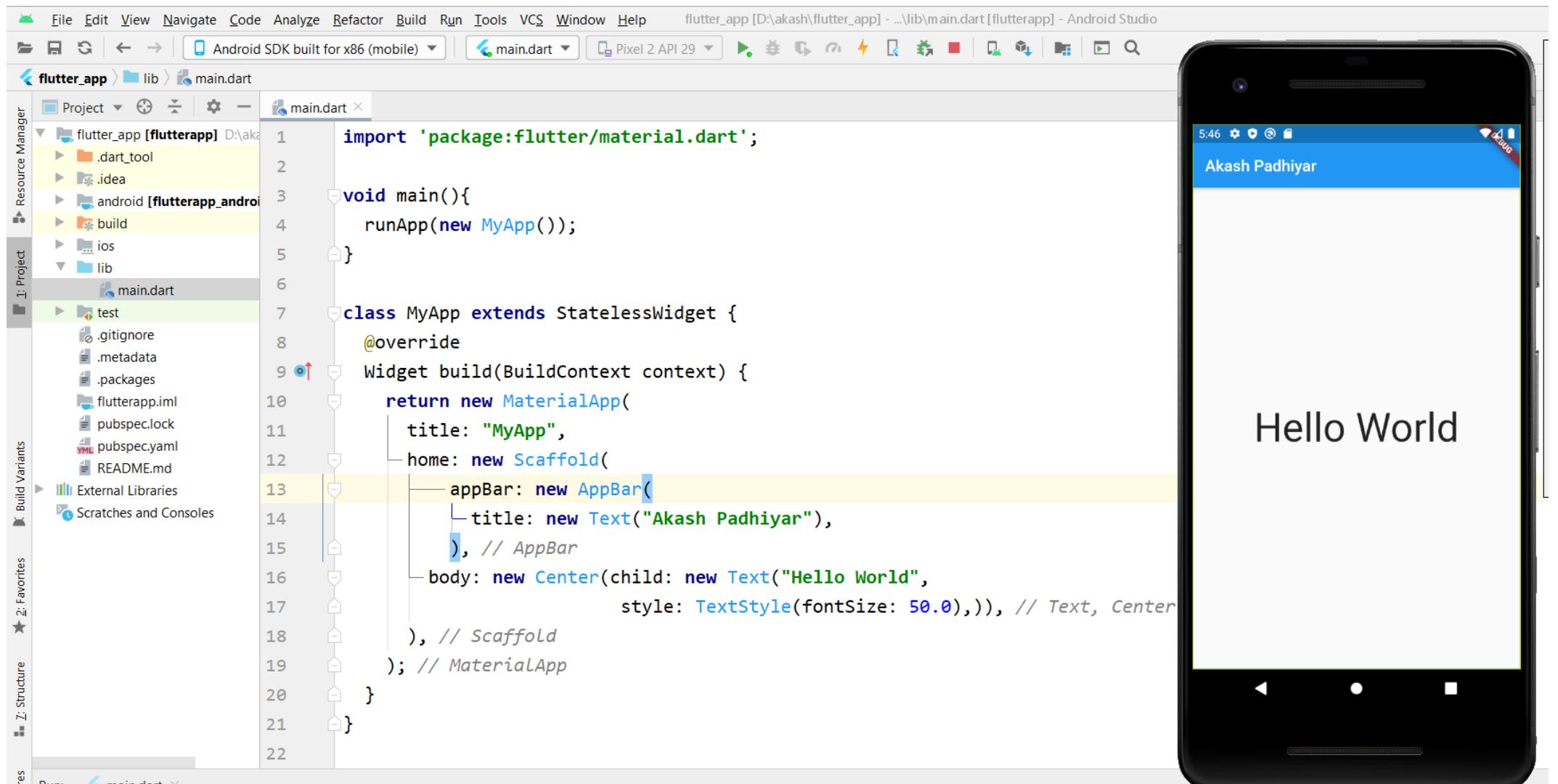
```
import 'package:flutter/material.dart';

void main(){
    runApp(new MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return new MaterialApp(
            title: "MyApp",
            home: new Scaffold(
                appBar: new AppBar(
                    title: new Text("Akash Padhiyar"),
                ),
                body: new Center(child: new Text("Hello World")),
            ),
        );
    }
}
```

- Output Preview:** On the right, a smartphone screen displays the app's UI. The title bar says "Akash Padhiyar" and the main screen shows "Hello World".
- Bottom Bar:** The "Run" tab is active, showing "main.dart". Other tabs include "Console" and "Captures".

TextStyle



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "flutter_app". The "lib/main.dart" file is open in the code editor.
- Code Editor:** The code for the "main.dart" file is displayed. It imports the necessary package and defines the main function which runs an instance of the MyApp class. The MyApp class extends StatelessWidget and overrides the build method to return a MaterialApp. This widget has a title of "MyApp", a home page (Scaffold) with an AppBar containing the text "Akash Padhiyar", and a body containing a large "Hello World" text.
- Run Configuration:** The "main.dart" file is selected in the run dropdown. The device is set to "Pixel 2 API 29".
- Preview:** On the right, a preview of the app running on the device shows the title bar with "Akash Padhiyar" and the main screen with the text "Hello World".

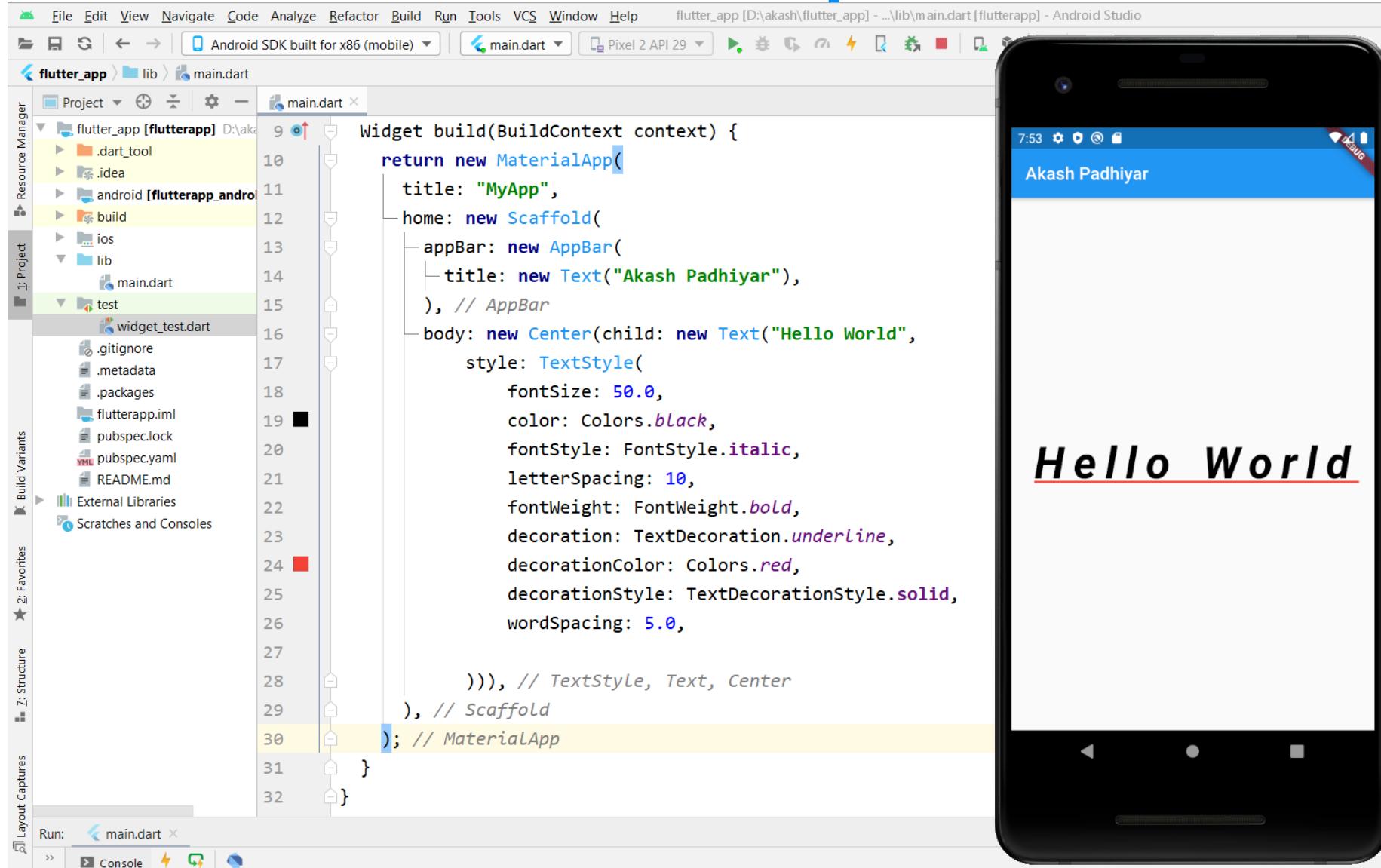
```
import 'package:flutter/material.dart';

void main(){
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "MyApp",
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text("Akash Padhiyar"),
        ), // AppBar
        body: new Center(child: new Text("Hello World",
          style: TextStyle(fontSize: 50.0),)), // Text, Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```



Font Properties



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The Project Manager shows the project structure with files like `main.dart`, `test/widget_test.dart`, and various build-related files.
- Code Editor:** The main editor window displays the `main.dart` code. The highlighted part of the code is:

```
Widget build(BuildContext context) {
  return new MaterialApp(
    title: "MyApp",
    home: new Scaffold(
      appBar: new AppBar(
        title: new Text("Akash Padhiyar"),
      ), // AppBar
      body: new Center(child: new Text("Hello World",
        style: TextStyle(
          fontSize: 50.0,
          color: Colors.black,
          fontStyle: FontStyle.italic,
          letterSpacing: 10,
          fontWeight: FontWeight.bold,
          decoration: TextDecoration.underline,
          decorationColor: Colors.red,
          decorationStyle: TextDecorationStyle.solid,
          wordSpacing: 5.0,
        )));
    ), // Scaffold
  ); // MaterialApp
}
```
- Emulator:** A Pixel 2 API 29 emulator is running, displaying the application's UI. The title bar says "Akash Padhiyar" and the main screen shows the text "Hello World" in a large, bold, italicized black font with a red underline.



TextStyle

```
style: TextStyle(  
    fontSize: 50.0,  
    color: Colors.black,  
    fontStyle: FontStyle.italic,  
    letterSpacing: 10,  
    fontWeight: FontWeight.bold,  
    decoration: TextDecoration.underline,  
    decorationColor: Colors.red,  
    decorationStyle: TextDecorationStyle.solid,  
    wordSpacing: 5.0,  
)  
),
```



Change Theme

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "flutter_app" and contains a "lib" directory which holds the "main.dart" file.
- Code Editor:** The "main.dart" file is open, displaying the Dart code for a Flutter application. The code defines a "MyApp" class that extends "StatelessWidget". It sets the app title to "MyApp" and the home screen to a "Scaffold" widget. The scaffold has an "AppBar" with the title "Akash Padhiyar" and a "body" containing a large "Hello World" text centered on the screen.
- Run Tab:** The "Run" tab is selected, showing the path "flutter_app [flutterapp] D:\akash\flutter_app" and the file "main.dart".
- Virtual Device:** A virtual device for a Pixel 2 API 29 is running, displaying the "Hello World" screen with the title "Akash Padhiyar" at the top and the text "Hello World" in the center.

Dark Theme

The screenshot shows an Android Studio environment with a Flutter project named "flutter_app". The main.dart file is open in the code editor, displaying the following Dart code:

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

void main(){
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "MyApp",
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text("Akash Padhiyar"),
        ), // AppBar
        body: new Center(child: new Text("Hello World",
          style: TextStyle(fontSize: 50.0))), // Text, Center
      ), // Scaffold
      theme: new ThemeData(
        primarySwatch: Colors.deepOrange,
        brightness: Brightness.dark //dark Light
      ), // ThemeData
    ); // MaterialApp
}
```

The code defines a simple Flutter application with a dark theme. The title of the app is "MyApp" and the main screen displays the text "Hello World" in a large, bold white font.



TextField

The screenshot shows an Android Studio environment with a Flutter project named "flutter_app". The main.dart file is open in the code editor, displaying the following Dart code:

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

void main(){
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "MyApp",
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text("Akash Padhiyar"),
        ), // AppBar
        body: new Center(child: new TextField()),
      ), // Scaffold
    ); // MaterialApp
  }
}
```

The right side of the interface shows a simulated iPhone X displaying the Flutter application. The app's title bar says "Akash Padhiyar". Below it is a text input field containing the same text. A virtual keyboard is visible at the bottom, showing the letters "Padhiyar" and other standard keyboard controls.



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help flutter_app [D:\akash\flutter_app] - ...lib\main.dart [flutterapp] - Android Studio

flutter_app lib main.dart

Project Manager

Resource Manager

1: Project

2: Favorites

Build Variants

External Libraries

Scratches and Consoles

Layout Captures

Run: main.dart

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

void main(){
    runApp(new MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return new MaterialApp(
            title: "MyApp",
            home: new Scaffold(
                appBar: new AppBar(
                    title: new Text("Akash Padhiyar"),
                ), // AppBar
                body: new Center(child: new TextField()),
            ), // Scaffold
        ); // MaterialApp
    }
}
```

autocorrect: ,
autofocus: ,
buildCounter: , Widget Function(Build...)
controller: , TextEditingController
cursorColor: , Color
cursorRadius: , Radius
cursorWidth: , double
decoration: , InputDecoration
dragStartBehavior: , DragStartBehavior
enabled: , bool
enableInteractiveSelection: , bool
onChanged: , Function(String)



The screenshot shows the Android Studio interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Includes icons for project navigation, code search, and build/run.
- Project Structure:** Shows the project tree with `flutter_app [flutterapp]` at the root, containing `.dart_tool`, `.idea`, `android [flutterapp_android]`, `build`, `ios`, and `lib` (which contains `main.dart`).
- Code Editor:** The `main.dart` file is open, displaying the following Dart code:

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return new MaterialApp(  
      title: "MyApp",  
      home: new Scaffold(  
        appBar: new AppBar(  
          title: new Text("Akash Padhiyar"),  
        ), // AppBar  
        body: new Center(child: new TextField(  
          keyboardType: TextInputType.emailAddress,  
          autocorrect: false,  
          textCapitalization: TextCapitalization.words,  
          textAlign: TextAlign.center,  
          maxLength: 160,  
          maxLines: 3,  
  
          decoration: InputDecoration(  
            border: OutlineInputBorder()  
          ) // InputDecoration  
        )), // TextField, Center  
      ), // Scaffold  
    ); // MaterialApp  
  }  
}
```
- Run Tab:** Set to `main.dart`.
- Bottom Navigation:** Logcat, Terminal, Dart Analysis, Run, TODO.
- Preview:** A virtual Android device displays the app's UI with the title "Akash Padhiyar" and a centered text field containing "Akash Padhiyar".



KeyBoard Type

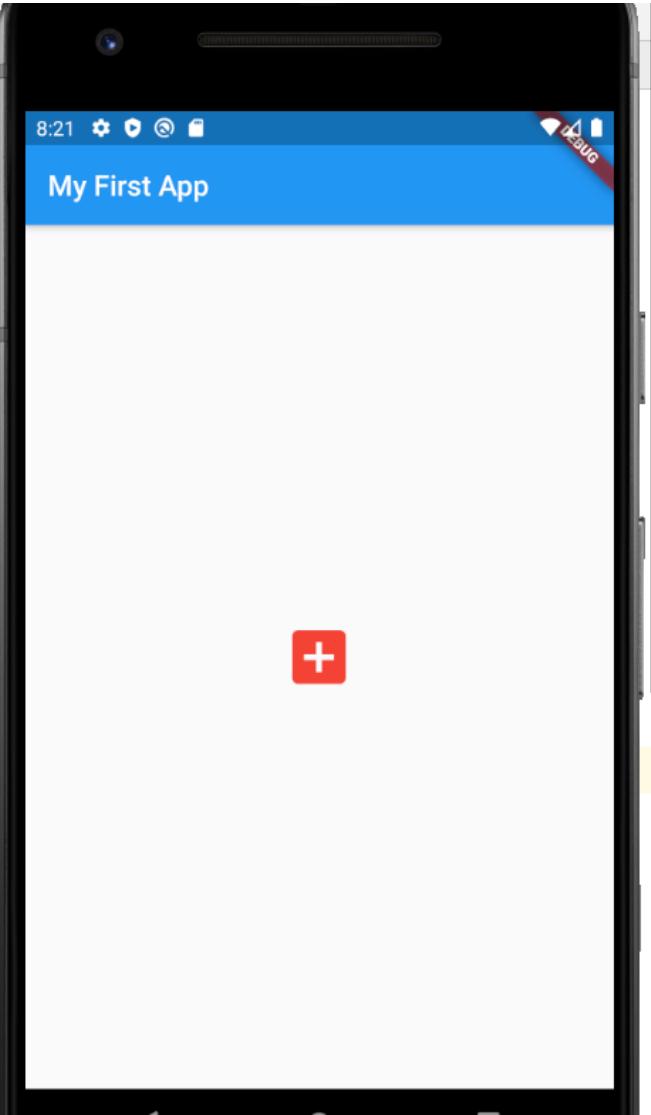
- **TextInputType.text** (Normal complete keyboard)
- **TextInputType.number** (A numerical keyboard)
- **TextInputType.emailAddress** (Normal keyboard with an “@”)
- **TextInputType.datetime** (Numerical keyboard with a “/” and “：“)
- **TextInputType.numberWithOptions** (Numerical keyboard with options to enable signed and decimal mode)
- **TextInputType.multiline** (Optimises for multi-line information)



Basic Layout

Icon

```
main.dart x
19     _MyScreenState createState() => _MyScreenState();
20 }
21
22 class _MyScreenState extends State<MyScreen> {
23     @override
24     Widget build(BuildContext context) {
25         return new Scaffold(
26             appBar: new AppBar(
27                 title: new Text("My First App"),
28             ), // AppBar
29             body: Center(
30                 child: Icon(
31                     Icons.add_box,
32                     color: Colors.red,
33                     size: 50.0,
34                 ), // Icon
35             ), // Center
36         ); // Scaffold
37     }
38 }
```



```
child: Icon(
    Icons.add_box,
    color: Colors.red,
    size: 50.0,
),
```



Basic layout widgets (multiple children)

- The widgets above only took one child. When creating a layout, though, it is often necessary to arrange multiple widgets together. We will see how to do that using rows, columns, and stacks.
-



Rows and columns

- Rows are easy. Just pass in a list of widgets to Row's children parameter.



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "flutter_app [flutterapp]". The "lib" folder contains "main.dart" and "test". "main.dart" is currently selected.
- Code Editor:** The main editor window shows the Dart code for "main.dart". The code defines a MaterialApp with a title "MyApp" and a Scaffold containing an AppBar with the title "Akash Padhiyar" and a body consisting of a Row of five home icons.
- Run Preview:** On the right, a virtual Android device is shown displaying the app's UI. The title bar says "Akash Padhiyar" and the screen shows a Row of five house icons.
- Bottom Navigation:** The bottom navigation bar includes tabs for Logcat, Terminal, Dart Analysis, Run, and TODO.

```
Widget build(BuildContext context) {  
  return new MaterialApp(  
    title: "MyApp",  
    home: new Scaffold(  
      appBar: new AppBar(  
        title: new Text("Akash Padhiyar"),  
      ), // AppBar  
      body:myLayoutWidget()  
    ), // Scaffold  
  ); // MaterialApp  
}  
}  
Widget myLayoutWidget() {  
  return Row(  
    children: [  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
      Icon(Icons.home),  
    ],  
  ); // Row  
}
```



The screenshot shows the Android Studio interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Android SDK built for x86 (mobile), main.dart, Pixel 2 API 29.
- Project Structure:** flutter_app [flutterapp] D:\akash\flutter_app, .dart_tool, .idea, android [flutterapp_android], build, ios, lib (selected), main.dart, test (selected), widget_test.dart, .gitignore, .metadata, .packages, flutterapp.iml, pubspec.lock, pubspec.yaml, README.md.
- Code Editor (main.dart):**

```
Widget build(BuildContext context) {
  return new MaterialApp(
    title: "MyApp",
    home: new Scaffold(
      appBar: new AppBar(
        title: new Text("Akash Padhiyar"),
      ), // AppBar
      body:myLayoutWidget()
    ), // Scaffold
  ); // MaterialApp
}

Widget myLayoutWidget() {
  return Column(
    children: [
      Icon(Icons.home),
      Icon(Icons.home),
      Icon(Icons.home),
      Icon(Icons.home),
    ],
  ); // Column
}
```
- Run Tab:** Run: main.dart.
- Preview:** A virtual Pixel 2 smartphone displays the app's UI with the title "Akash Padhiyar" and four home icons in a column.

The screenshot shows the Android Studio interface with a Flutter project named "flutter_app". The main.dart file is open in the editor, displaying the following code:

```
Widget build(BuildContext context) {
    return new MaterialApp(
        title: "MyApp",
        home: new Scaffold(
            appBar: new AppBar(
                title: new Text("Akash Padhiyar"),
            ), // AppBar
            body:myLayoutWidget()
        ), // Scaffold
    ); // MaterialApp
}
Widget myLayoutWidget() {
    return Row(
        children: [
            Expanded(child: Icon(Icons.home)),
            Expanded(child: Icon(Icons.home)),
            Expanded(child: Icon(Icons.home)),
            Expanded(child: Icon(Icons.home)),
        ],
    ); // Row
}
```

The virtual device on the right displays the app's UI with the title "Akash Padhiyar" and four identical home icons in a row.

The screenshot shows the Android Studio interface with a Flutter project named "flutter_app". The main.dart file is the active code editor, displaying the following Dart code:

```
13     ), // AppBar
14     body:myLayoutWidget()
15   ), // Scaffold
16 ); // MaterialApp
17 }
18 }Widget myLayoutWidget() {
19   return Row(
20     children: [
21       Expanded(
22         flex: 7,
23         child: Container(
24           color: Colors.green,
25         ), // Container
26       ), // Expanded
27       Expanded(
28         flex: 3,
29         child: Container(
30           color: Colors.yellow,
31         ), // Container
32       ), // Expanded
33     ],
34   ); // Row
35 }
```

The virtual device running the app shows a screen with a blue header bar containing the text "Akash Padhiyar". Below the header is a horizontal row divided into two sections: a green section on the left and a yellow section on the right.

Alignment

Alignment.topLeft

Alignment.topCenter

Alignment.topRight

Alignment.centerLeft

Alignment.center

Alignment.centerRight

Alignment.bottomLeft

Alignment.bottomCenter

Alignment.bottomRight

Alignment(-1.0, -1.0)

Alignment(0.0, -1.0)

Alignment(1.0, -1.0)

Alignment(-1.0, 0.0)

Alignment(0.0, 0.0)

Alignment(1.0, 0.0)

Alignment(-1.0, 1.0)

Alignment(0.0, 1.0)

Alignment(1.0, 1.0)



Stateful Widget

Stateful Widget



Akash Technolabs

www.akashsir.com



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

void main(){
    runApp(new MyApp());
}

st
  stless      New Stateless widget
  stanim     New Stateful widget with Animation...
  stful       New Stateful widget
  const
  abstract
```



The screenshot shows the Android Studio interface with the following details:

- Top Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, flutter_app [D:\akash\flutter_app] - ...lib\main.dart
- Toolbar:** Android SDK built for x86 (mobile), main.dart, Pixel 2 API 29, various run and build icons.
- Project Structure:** flutter_app [flutterapp] D:\akash, .dart_tool, .idea, android [flutterapp_android], build, ios, lib, main.dart (selected), test, .gitignore, .metadata, .packages, flutterapp.iml, pubspec.lock, pubspec.yaml, README.md.
- Build Variants:** External Libraries, Scratches and Consoles.
- Favorites:** 2: Favorites.
- Code Editor:** The main.dart file is open, showing the following Dart code:

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

void main(){
    runApp(new MyApp());
}

class MyApp extends StatefulWidget {
    @override
    _MyAppState createState() => _MyAppState();
}

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



The screenshot shows the Android Studio interface with the following details:

- File Bar:** File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
- Toolbar:** Includes icons for project navigation, build, run, and device selection.
- Project Structure:** Shows the project tree:
 - flutter_app [flutterapp] D:\akash\flutter_app
 - .dart_tool
 - .idea
 - android [flutterapp_android]
 - build
 - ios
 - lib
 - main.dart
 - test
 - .gitignore
 - .metadata
 - .packages
 - flutterapp.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Code Editor:** The main.dart file is open, showing the following code:

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

void main(){
    runApp(new MyApp());
}

//Stateless
class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Container();
    }
}

//Stateful
class MyScreen extends StatefulWidget {
    @override
    _MyScreenState createState() => _MyScreenState();
}

class _MyScreenState extends State<MyScreen> {
    @override
    Widget build(BuildContext context) {
        return Container();
    }
}
```
- Sidemenu:** Includes sections for Project, Build Variants, Favorites, and Captures.



The screenshot shows the Android Studio interface with the following details:

- File Bar:** File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
- Project Bar:** flutter_app [D:\akash\flutter_app] - ...lib\main.dart [flutter]
- Toolbars:** Standard toolbar with icons for file operations.
- Main Area:**
 - Resource Manager:** Shows the project structure under "flutter_app [flutterapp]". The "lib/main.dart" file is currently selected.
 - Code Editor:** Displays the Dart code for the "main.dart" file. The code defines a simple Flutter application with a home screen named "MyScreen".

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

void main(){
    runApp(new MyApp());
}

//StateLess
class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return new MaterialApp(
            title: "My App",
            home: new MyScreen(),
        ); // MaterialApp
    }
}

//Stateful
class MyScreen extends StatefulWidget {
    @override
    _MyScreenState createState() => _MyScreenState();
}
```



The screenshot shows the Android Studio interface with the following details:

- File Bar:** File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
- Toolbar:** Includes icons for file operations (New, Open, Save, Find, etc.), build (Build, Clean, Run), and run configuration (Run, Stop, Emulator).
- Project Structure:** Shows the project tree:
 - flutter_app [flutterapp] D:\akash\flutter_app
 - .dart_tool
 - .idea
 - android [flutterapp_android]
 - build
 - ios
 - lib
 - main.dart
 - test
 - .gitignore
 - .metadata
 - .packages
 - flutterapp.iml
 - pubspec.lock
 - pubspec.yaml
 - README.md
- Code Editor:** The main editor window displays the `main.dart` file content. The code defines a simple Flutter application with a single screen.

```
return new MaterialApp(  
    title: "My App",  
    home: new MyScreen(),  
) // MaterialApp  
}  
}  
//Stateful  
class MyScreen extends StatefulWidget {  
@override  
_MyScreenState createState() => _MyScreenState();  
}  
  
class _MyScreenState extends State<MyScreen> {  
@override  
Widget build(BuildContext context) {  
    return new Scaffold(  
        appBar: new AppBar(  
            title: new Text("My First App"),  
, // AppBar  
        body: new Center(child: new Text("Welcome to App")),  
, // Scaffold  
    );  
}  
}
```



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

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

//StateLess
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "My App",
      home: new MyScreen(),
    );
  }
}

//StateFul
class MyScreen extends StatefulWidget {
  @override
  _MyScreenState createState() => _MyScreenState();
}

class _MyScreenState extends State<MyScreen> {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text("My First App"),
      ),
      body: new Center(child: new Text("Welcome to App"),),
    );
  }
}
```

StateLess and StateFull Widget



Raised Button

RaisedButton

- RaisedButton is one of the most widely used widget in the flutter material library. It is actually a simple button which can handle normal click event.

```
new RaisedButton(  
    onPressed: () => print("Button Pressed"),  
    child: new Text("Press here"),  
,
```



Button

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "flutter_app" and contains "lib", "test", and "lib/main.dart".
- Code Editor:** The main.dart file is open, showing the following Dart code:

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "MyApp",
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text("Akash Padhiyar"),
        ),
        body: new Center(child: new RaisedButton(
          onPressed: () => print("Button Pressed"),
          child: new Text("Press here"),
        )),
      ),
    );
  }
}
```
- Run Tab:** The "main.dart" tab is selected in the Run tab.
- Log Output:** The log shows the following output:

```
Performing hot reload...
Syncing files to device Android SDK built for x86...
Reloaded 1 of 478 libraries in 254ms.
I/flutter (14442): Button Pressed
I/flutter (14442): Button Pressed
```
- Virtual Device:** A Pixel 2 API 29 device is shown running the app, displaying the title "Akash Padhiyar" and a central button labeled "Press here".



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help flutter_app [D:\akash\flutter_app] - ...\\lib

Android SDK built for x86 (mobile) main.dart Pixel 2 API 29

flutter_app lib main.dart

Project Manager flutter_app [flutterapp] D:\akash\flutter_app .dart_tool .idea android [flutterapp_android] build ios lib main.dart test widget_test.dart .gitignore .metadata .packages flutterapp.iml pubspec.lock pubspec.yaml README.md External Libraries Scratches and Consoles

1: Project 2: Favorites 3: Structure

main.dart

```
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return new MaterialApp(
12      title: "MyApp",
13      home: new Scaffold(
14        appBar: new AppBar(
15          title: new Text("Akash Padhiyar"),
16        ), // AppBar
17        body: new Center(child: new FlatButton(
18          onPressed: () => print("Button Pressed"),
19          child: new Text("Press here"),
20          color: Colors.red,
21          splashColor: Colors.yellow[200],
22          padding: EdgeInsets.all(20.0)
23        )), // FlatButton, Center
24      ), // Scaffold
25    ); // MaterialApp
26  }
27}
```

Run: main.dart

Console I/flutter (14442): Button Pressed
I/flutter (14442): Button Pressed

9:19 Akash Padhiyar

Press here



Navigation

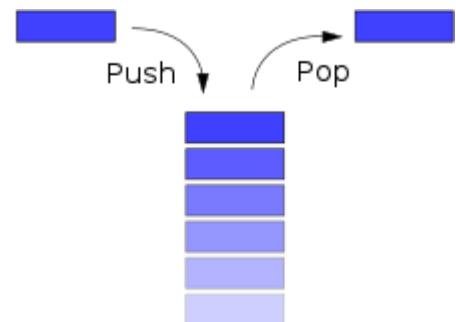
Push Pop

■ Push:

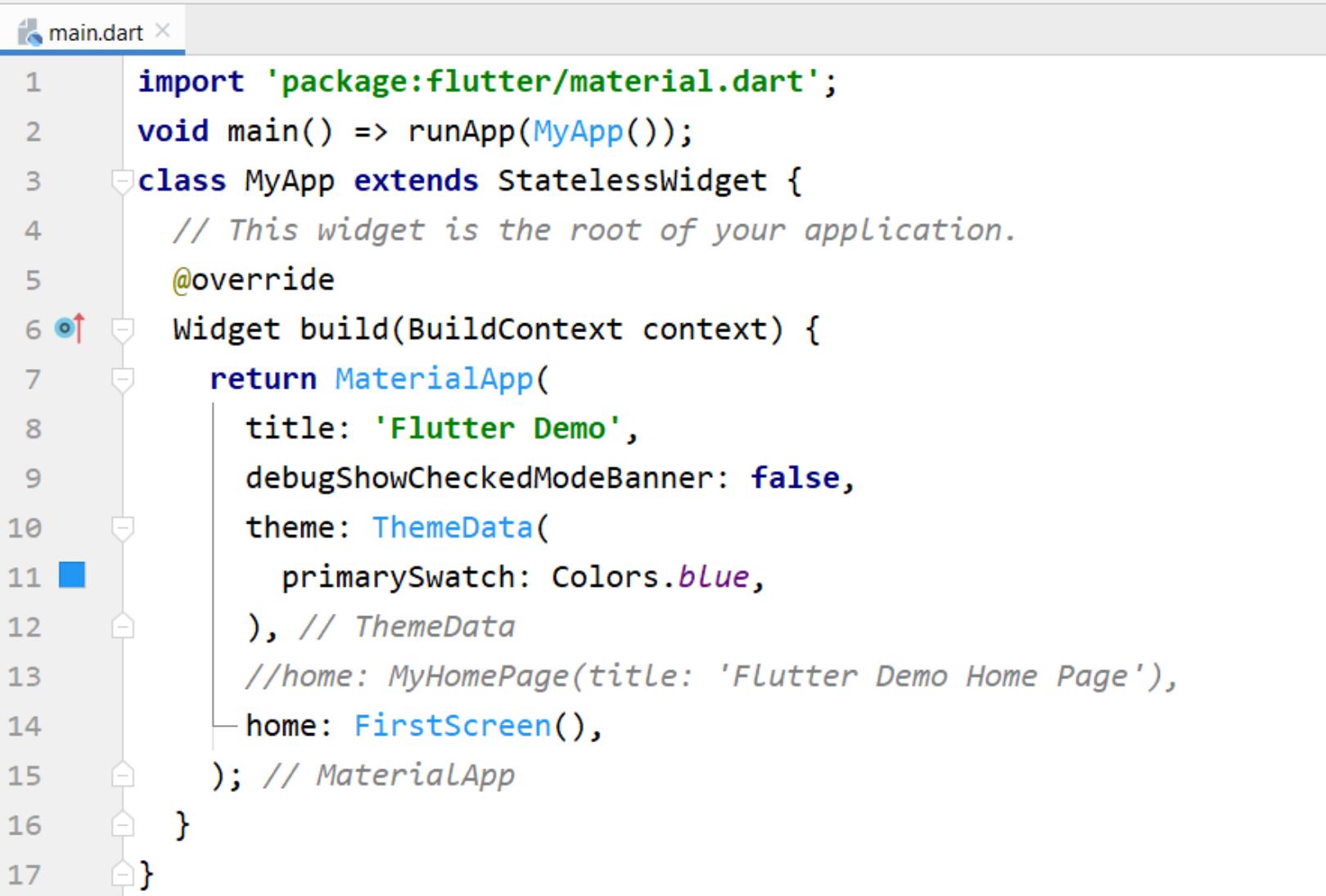
- Adding an element at the top of the existing stack is done by calling the push function.
- To add another stone to the stack, you can only think of adding it on the top of it.

■ Pop:

- An element is removed from the top of the stack by calling the pop function.
- You will have to go from the top and remove them one by one.



Main App



The screenshot shows a code editor window with the file 'main.dart' open. The code is a Dart script for a Flutter application. It defines a class 'MyApp' that extends 'StatelessWidget'. The build method returns a 'MaterialApp' widget with a title of 'Flutter Demo', a theme defined by 'ThemeData' with a primary color of 'Colors.blue', and a home screen of 'FirstScreen'. The code is numbered from 1 to 17.

```
1 import 'package:flutter/material.dart';
2 void main() => runApp(MyApp());
3 class MyApp extends StatelessWidget {
4     // This widget is the root of your application.
5     @override
6     Widget build(BuildContext context) {
7         return MaterialApp(
8             title: 'Flutter Demo',
9             debugShowCheckedModeBanner: false,
10            theme: ThemeData(
11                primarySwatch: Colors.blue,
12            ), // ThemeData
13            //home: MyHomePage(title: 'Flutter Demo Home Page'),
14            home: FirstScreen(),
15        ); // MaterialApp
16    }
17}
```



Screen 1

```
main.dart x
20   class FirstScreen extends StatelessWidget {
21     @override
22     Widget build(BuildContext context) {
23       return Scaffold(
24         appBar: AppBar(
25           title: Text('First Screen'),
26         ), // AppBar
27         body: Center(
28           child: RaisedButton(
29             color: Colors.blue,
30             child: Text('Go to Second Screen'),
31             onPressed: () {
32               //Use `Navigator` widget to push the second screen to out stack of screens
33               Navigator.of(context)
34                 .push(MaterialPageRoute<Null>(builder: (BuildContext context) {
35                   return new SecondScreen();
36                 })); // MaterialPageRoute
37               },
38             ),
39           ),
40         ), // Center
41       ); // Scaffold
42     }
43 }
```



Screen2



The image shows a screenshot of a code editor with a blue header bar. The tab bar at the top has a single tab labeled "main.dart x". The main content area displays the following Dart code:

```
44
45 class SecondScreen extends StatelessWidget {
46     @override
47     Widget build(BuildContext context) {
48         return Scaffold(
49             appBar: AppBar(
50                 title: Text('Second Screen'),
51             ), // AppBar
52             body: Center(
53                 child: RaisedButton(
54                     color: Colors.red,
55                     child: Text('Go back to First Screen'),
56                     onPressed: () {
57                         //Use `Navigator` widget to pop our go back to previous route / screen
58                         Navigator.pop(context);
59                     },
60                 ), // RaisedButton
61             ), // Center
62         ); // Scaffold
63     }
64 }
65
```

The code defines a class named "SecondScreen" that extends "StatelessWidget". It overrides the "build" method to return a "Scaffold" widget. The "appBar" property contains an "AppBar" with the title "Second Screen". The "body" property contains a "Center" widget, which in turn contains a "RaisedButton". The button has a red color and the text "Go back to First Screen". When pressed, it uses the "Navigator.pop" method to go back to the previous screen.



Navigation

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

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

class MyApp extends StatelessWidget {
    // This widget is the root of your application.

    @override

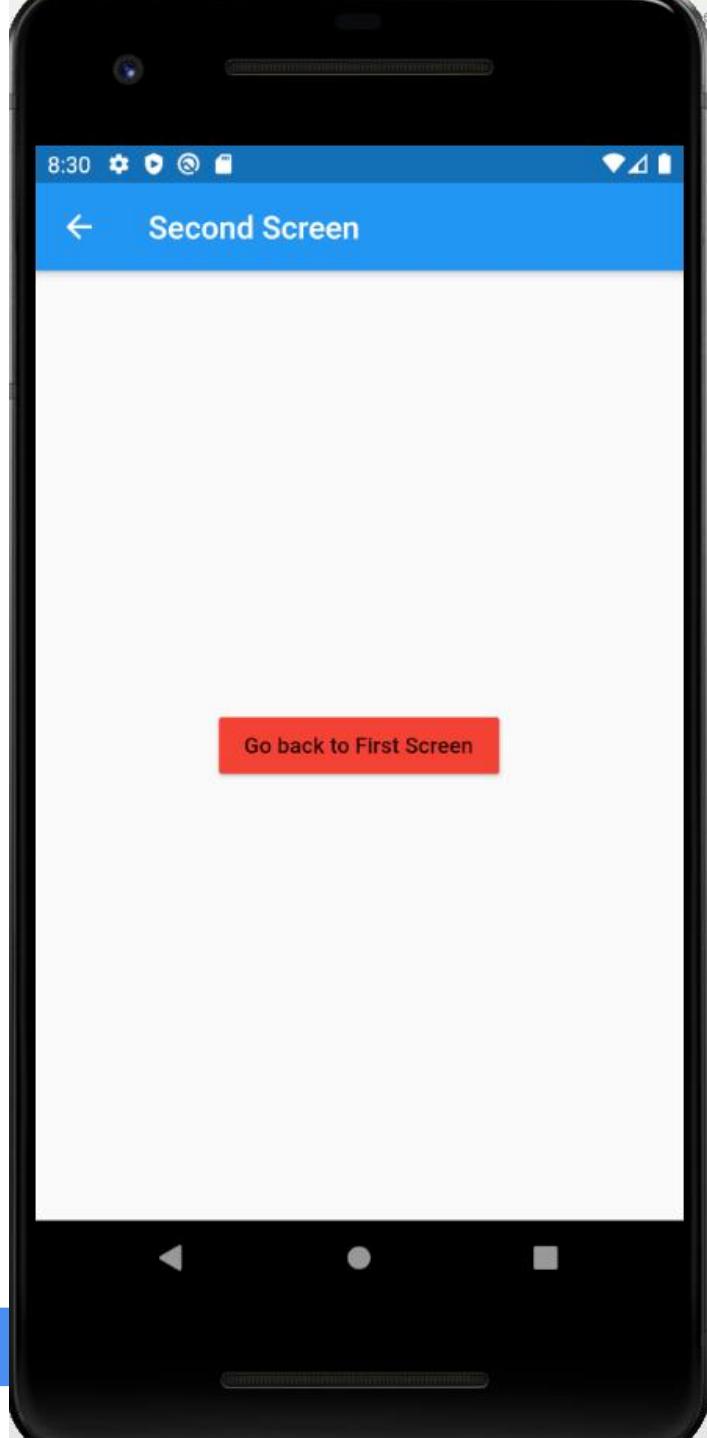
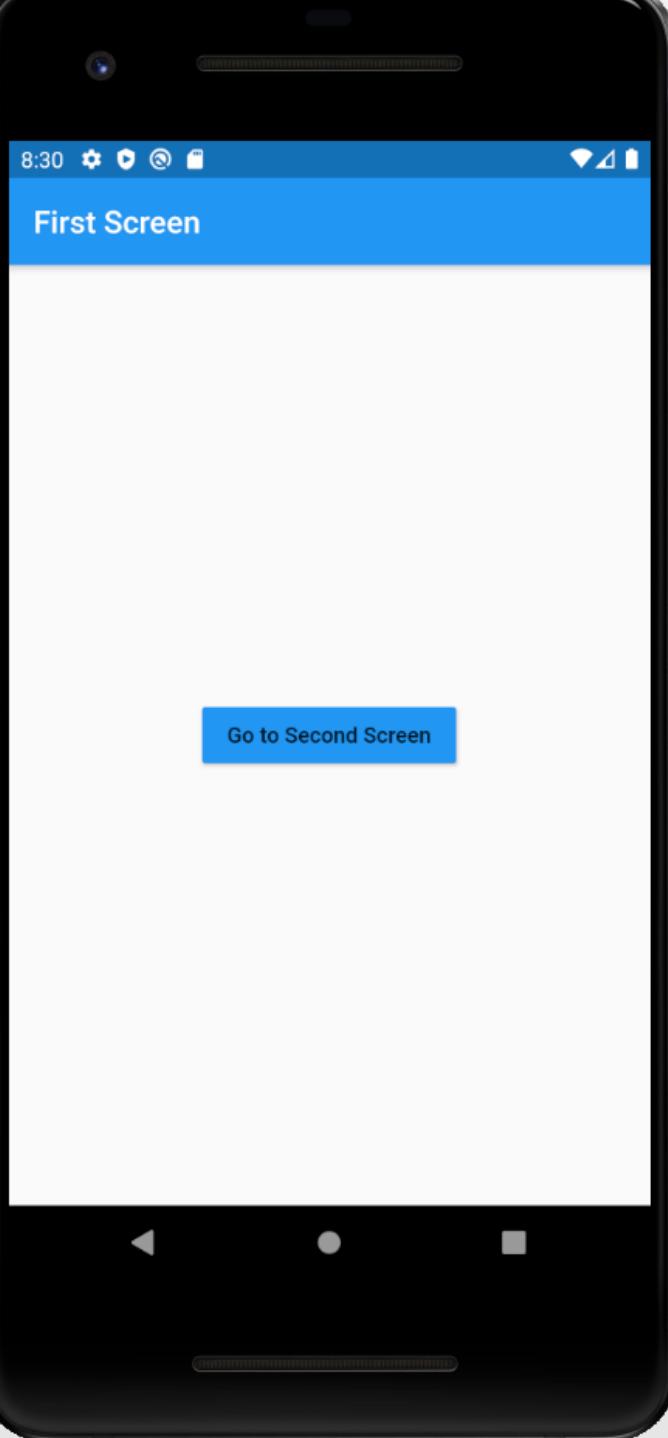
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Flutter Demo',
            debugShowCheckedModeBanner: false,
            theme: ThemeData(
                primarySwatch: Colors.blue,
            ),
            //home: MyHomePage(title: 'Flutter Demo Home Page'),
            home: FirstScreen(),
        );
    }
}
```



Screen 1 and Screen 2

```
class FirstScreen extends StatelessWidget {  
  
    @override  
  
    Widget build(BuildContext context) {  
  
        return Scaffold(  
  
            appBar: AppBar(  
  
                title: Text('First Screen'),  
  
,  
  
            body: Center(  
  
                child: RaisedButton(  
  
                    color: Colors.blue,  
  
                    child: Text('Go to Second Screen'),  
  
                    onPressed: () {  
  
                        //Use `Navigator` widget to push the second screen to out stack of screens  
  
                        Navigator.of(context)  
  
.push(MaterialPageRoute<Null>(builder: (BuildContext context) {  
  
                            return new SecondScreen();  
  
                        }));  
  
,  
  
,  
  
,  
  
                );  
  
            );  
  
        );  
  
    }  
  
}  
  
class SecondScreen extends StatelessWidget {  
  
    @override  
  
    Widget build(BuildContext context) {  
  
        return Scaffold(  
  
            appBar: AppBar(  
  
                title: Text('Second Screen'),  
  
,  
  
            body: Center(  
  
                child: RaisedButton(  
  
                    color: Colors.red,  
  
                    child: Text('Go back to First Screen'),  
  
                    onPressed: () {  
  
                        //Use `Navigator` widget to pop our go back to previous route / screen  
  
                        Navigator.pop(context);  
  
                    },  
  
                ),  
  
            );  
  
        );  
  
    }  
  
}
```





Aka

m



Topics

- Static ListView
- API
- HTTP
- AdMob
- Google Map
- Few Apps



Best Sites to Learn

- <https://flutterawesome.com/>

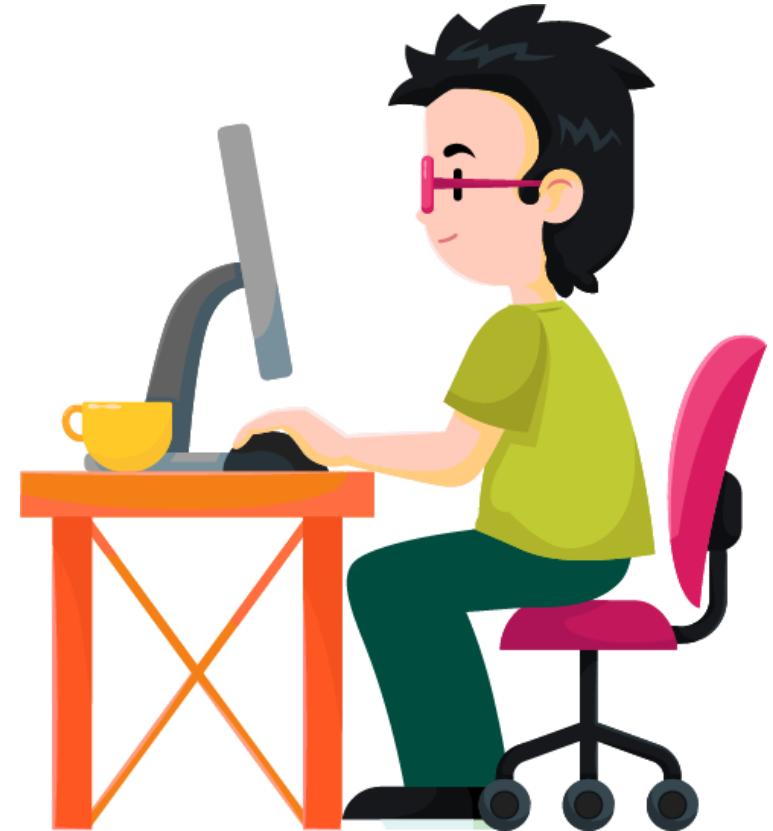


Akash Technolabs

www.akashsir.com



Get Exclusive Video Tutorials



www.aptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



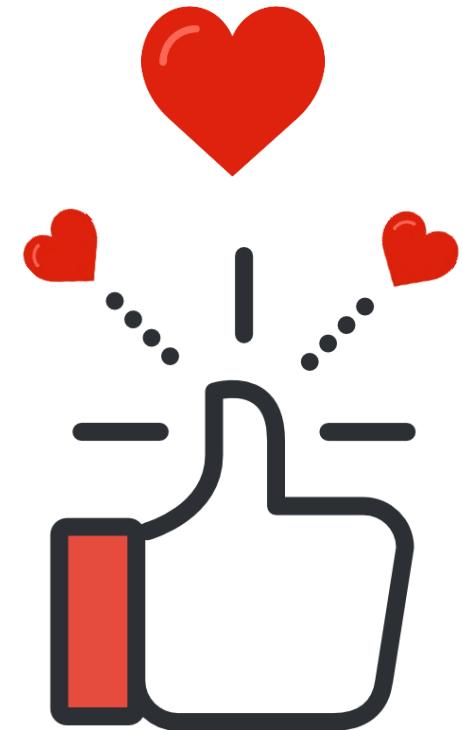
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechnolabs.com

www.akashpadhiyar.com

www.aptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar
#aptutorials