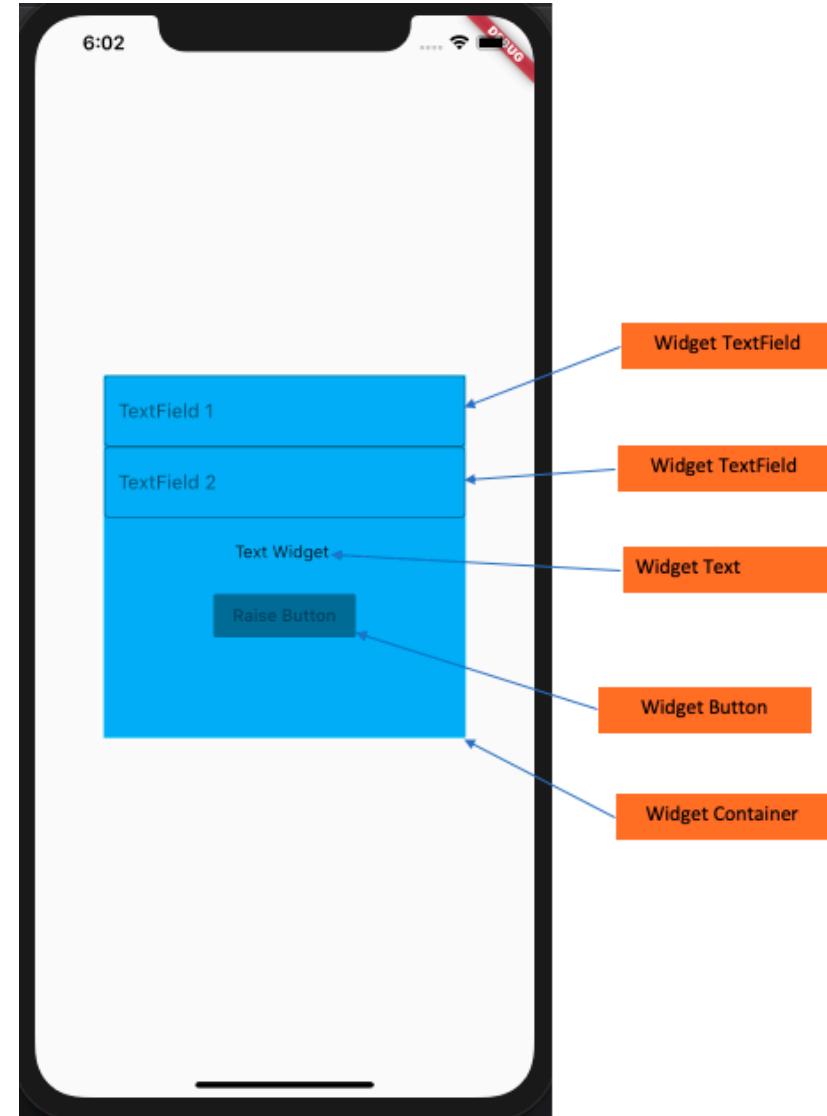


Basic Widgets

What is widgets in Flutter?

- Each **element** on a **screen** of the Flutter app is a **widget**.
- The view of the **screen** completely depends upon the **choice** and **sequence** of the **widgets** used to build the app.



Text

- The Text widget displays a string.
- url: <https://api.flutter.dev/flutter/widgets/Text-class.html>

```
import 'package:flutter/material.dart';
Run | Debug | Profile
void main() => runApp(DemoApp());
class DemoApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Demo App',
      home: Text('Hello Flutter!'),
    ); // MaterialApp
  }
}
```

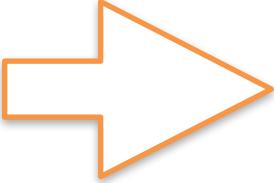
Text

```
class DemoApp extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            debugShowCheckedModeBanner: false,  
            title: 'Demo App',  
            home: SizedBox(  
                width: double.infinity,  
                child: Text(  
                    'Hello Flutter',  
                    textAlign: TextAlign.center,  
                    overflow: TextOverflow.ellipsis,  
                    style: TextStyle(  
                        fontSize: 20,  
                        fontWeight: FontWeight.bold,  
                        color: Colors.blue), // TextStyle  
                ), // Text  
            ), // SizedBox  
        ); // MaterialApp  
    }  
}
```



Text

```
home: Scaffold(  
    appBar: AppBar(  
        title: Text('My First App'),  
    ), // AppBar  
    body: Text.rich(  
        TextSpan(  
            text: 'Hello', // default text style  
            children: [  
                TextSpan(  
                    text: ' beautiful ',  
                    style: TextStyle(fontStyle: FontStyle.italic)), // TextSpan  
                TextSpan(  
                    text: 'world',  
                    style: TextStyle(fontWeight: FontWeight.bold)), // TextSpan  
            ],  
        ), // TextSpan  
    )), // Text.rich // Scaffold
```

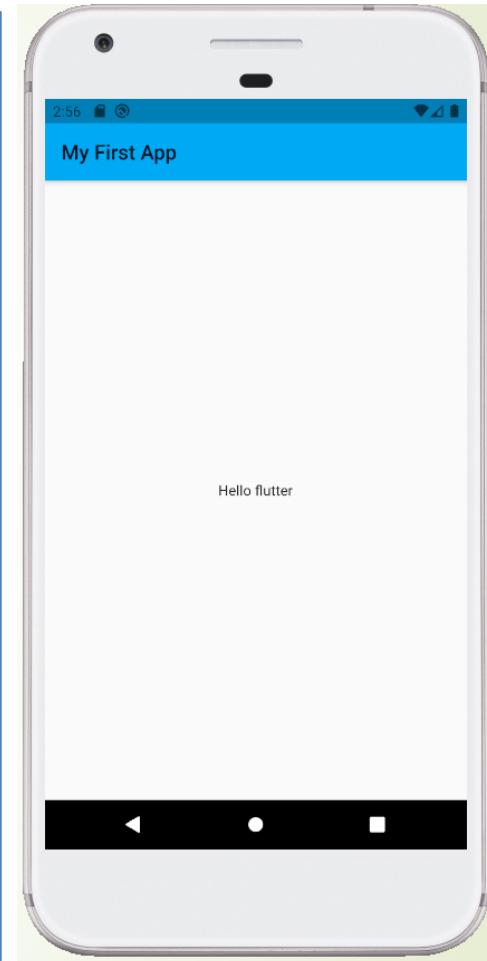


Hello *beautiful* world

Center

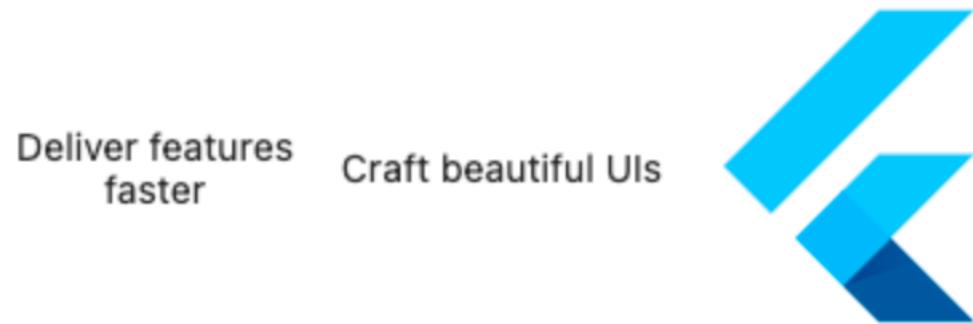
- A widget that centers its child within itself.
- Url: <https://api.flutter.dev/flutter/widgets/Center-class.html>

```
class DemoApp extends StatelessWidget {  
    // This widget is the root of your application.  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            debugShowCheckedModeBanner: false,  
            title: 'Demo App',  
            theme: ThemeData(  
                primarySwatch: Colors.lightBlue,  
            ), // ThemeData  
            home: Scaffold(  
                appBar: AppBar(  
                    title: Text('My First App'),  
                ), // AppBar  
                body: Center(child: Text('Hello flutter')),  
            ), // Scaffold  
        ); // MaterialApp  
    }  
}
```



Row and Column

- Row:
 - A widget that displays its children in a **horizontal** array.
 - <https://api.flutter.dev/flutter/widgets/Row-class.html>
- Column:
 - A widget that displays its children in a **vertical** array.
 - <https://api.flutter.dev/flutter/widgets/Column-class.html>



Deliver features
faster

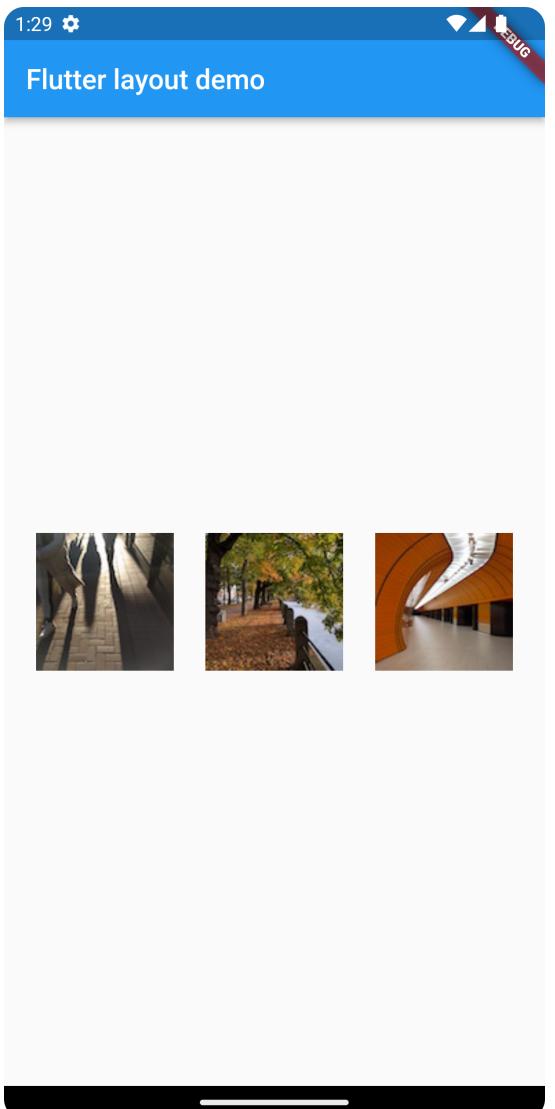
Craft beautiful UIs

Deliver features faster
Craft beautiful UIs



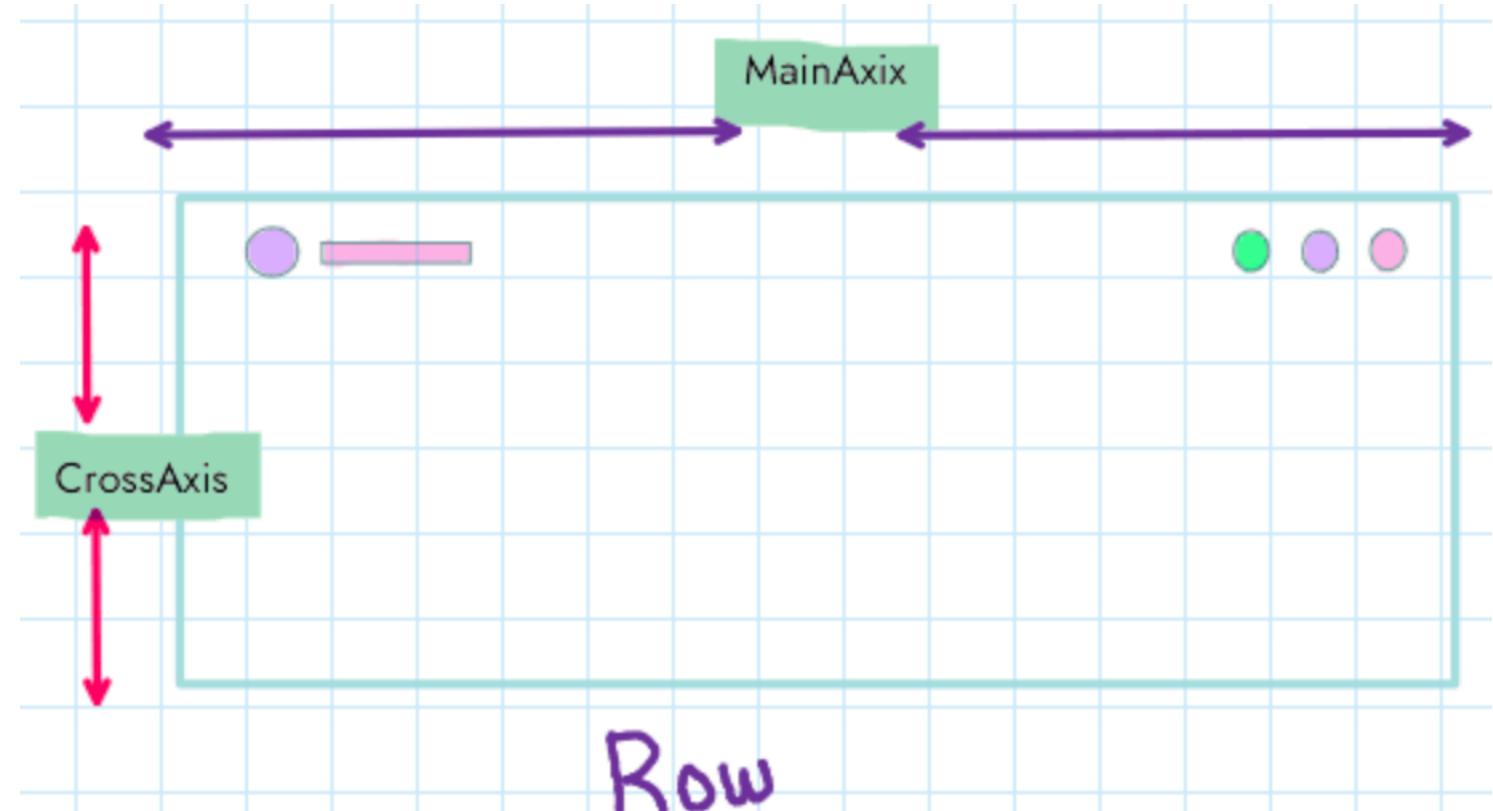
Using Row

```
return Row(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: [  
        Image.asset('images/pic1.jpeg'),  
        Image.asset('images/pic2.jpeg'),  
        Image.asset('images/pic3.jpeg'),  
    ],  
); // Row
```



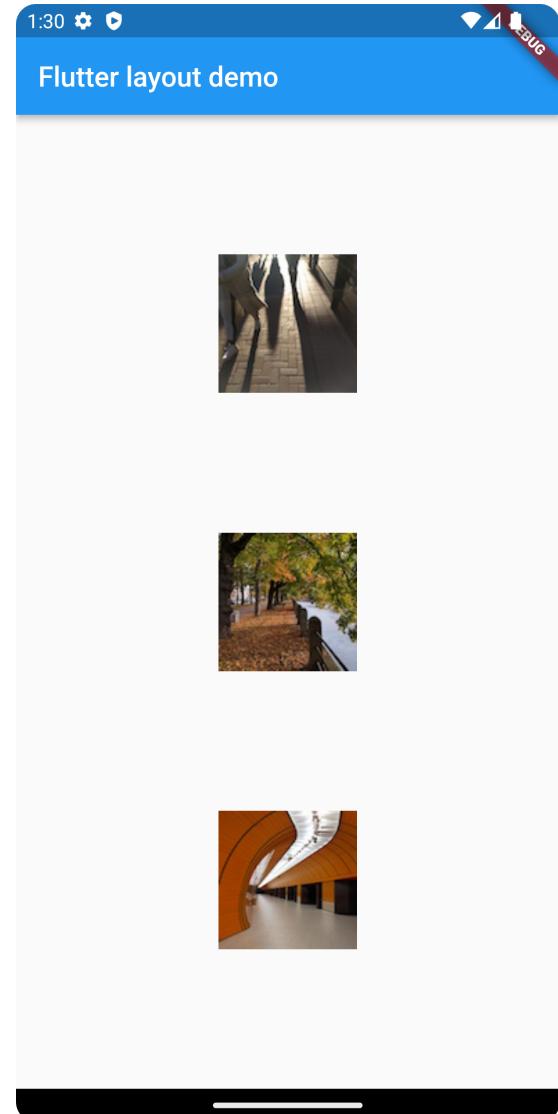
Using Row

- In a row,
 - to center(or align) horizontally, `mainAxisAlignment` is used.
 - to center(or align) vertically, `crossAxisAlignment` is used.



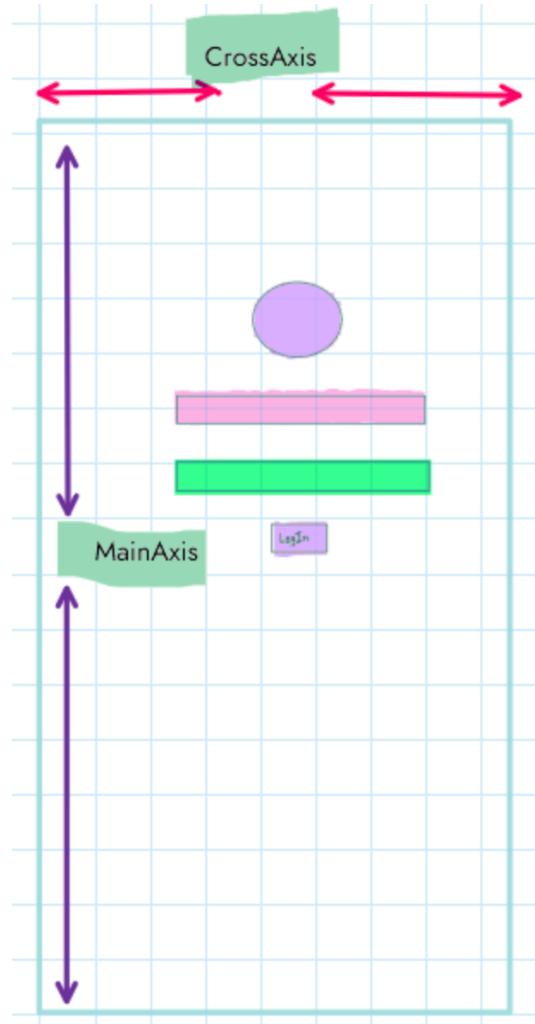
Using Column

```
return Column(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: [  
        Image.asset('images/pic1.jpeg'),  
        Image.asset('images/pic2.jpeg'),  
        Image.asset('images/pic3.jpeg'),  
    ],  
); // Column
```

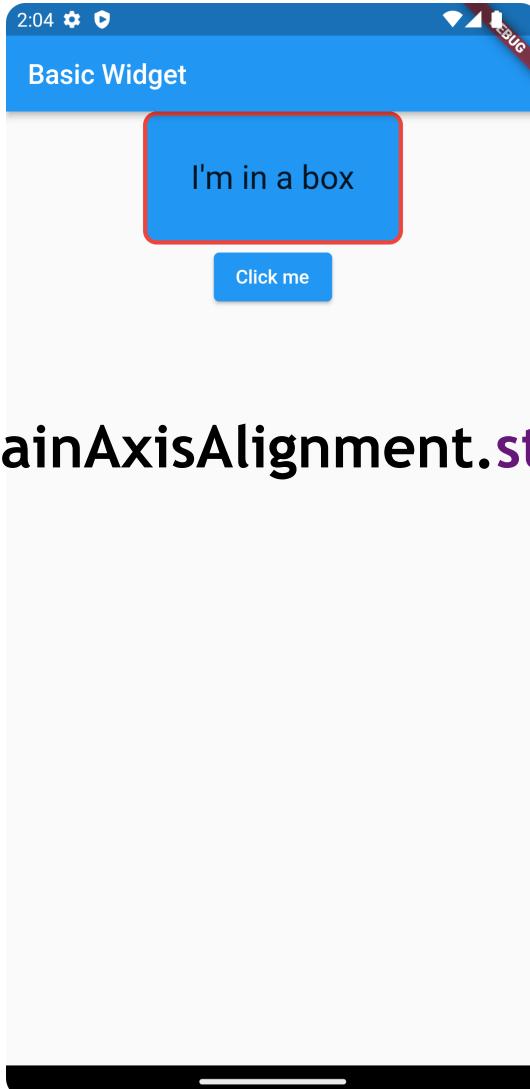


Using Column

- In a column,
 - to center(or align) vertically, **mainAxisAlignment** is used.
 - to center(or align) horizontally, **crossAxisAlignment** is used.



Using Column



MainAxisAlignment.start



MainAxisAlignment.center

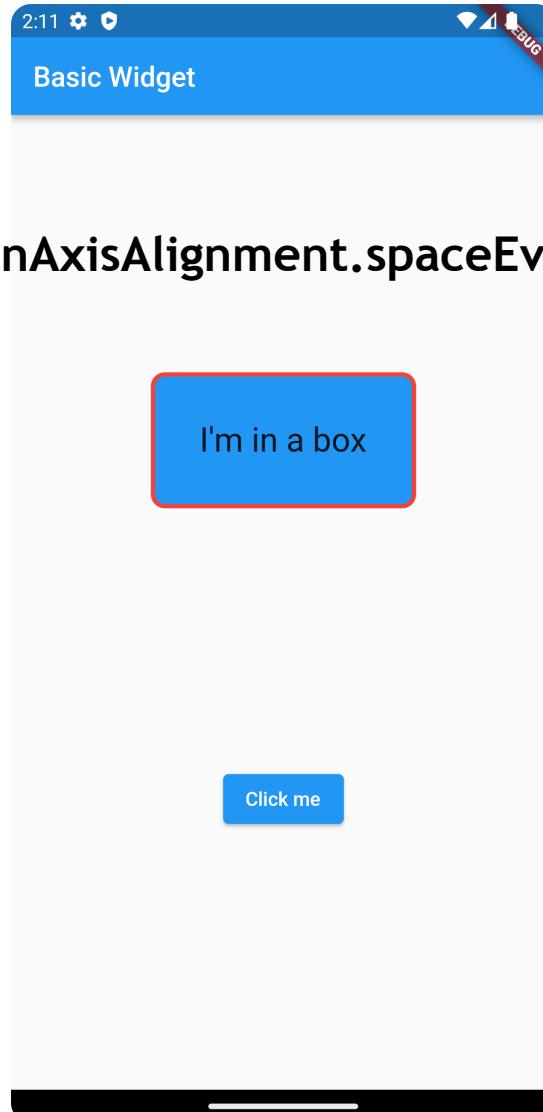


MainAxisAlignment.end

Using Column



MainAxisAlignment.spaceBetween



MainAxisAlignment.spaceEvenly



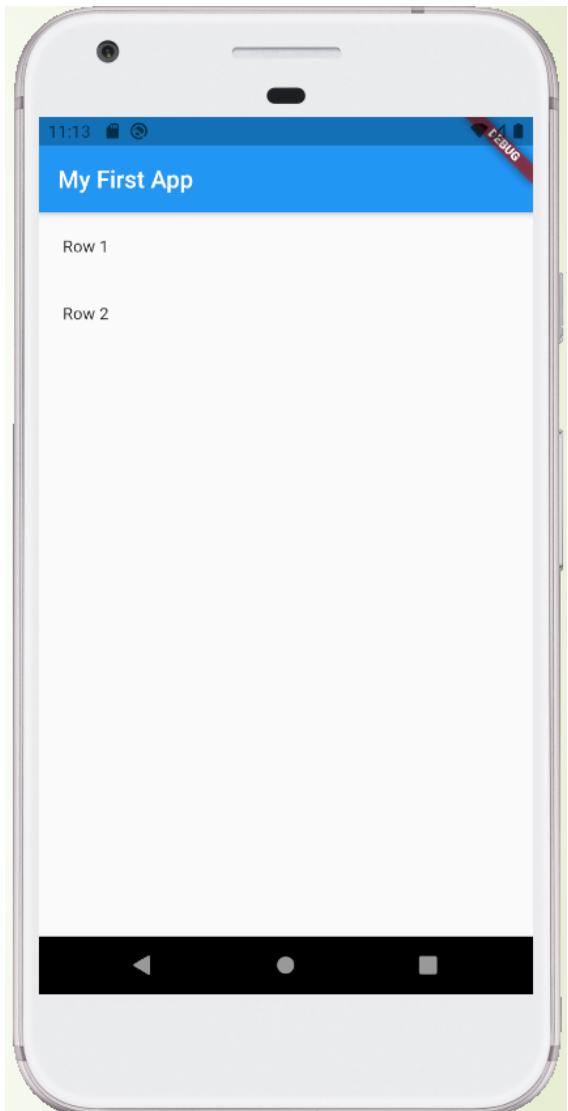
MainAxisAlignment.spaceAround

(Similar to **spaceEvenly**, but reduces half of the space before the first child and after the last child to half of the width between the children.)

Padding

- The size of the child widget inside padding is constrained by how much space is remaining after adding empty space around.
- Url: <https://api.flutter.dev/flutter/widgets/Padding-class.html>

```
return MaterialApp(  
    title: 'Flutter Demo',  
    home: Scaffold(  
        appBar: AppBar(  
            title: Text('My First App'),  
        ), // AppBar  
        body: Column(  
            children: [  
                Padding(  
                    padding: EdgeInsets.only(top: 20, bottom: 20),  
                    child: Text('Row 1'),  
                ), // Padding  
                Padding(  
                    padding: EdgeInsets.all(20),  
                    child: Text('Row 2'),  
                ), // Padding  
            ],  
        ), // Column  
    )); // Scaffold // MaterialApp
```



EdgeInsets

- Constructors:
 - `EdgeInsets.all(double value)` : Creates insets where all the offsets are value
 - `EdgeInsets.only({double left = 0.0, double top = 0.0, double right = 0.0, double bottom = 0.0})`: Creates insets with only the given values non-zero.
 - `EdgeInsets.symmetric({double vertical = 0.0, double horizontal = 0.0})`: Creates insets with symmetrical vertical and horizontal offsets.
- Url: <https://api.flutter.dev/flutter/painting/EdgeInsets-class.html>

EdgeInsets

- Example

Here are some examples of how to create EdgeInsets instances:

Typical eight-pixel margin on all sides:

```
const EdgeInsets.all(8.0)
```

Eight pixel margin above and below, no horizontal margins:

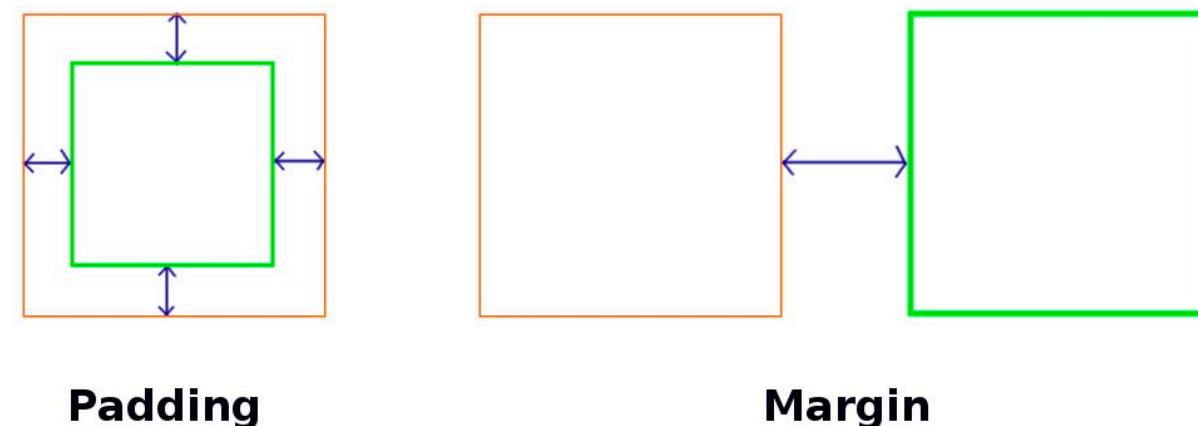
```
const EdgeInsets.symmetric(vertical: 8.0)
```

Left margin indent of 40 pixels:

```
const EdgeInsets.only(left: 40.0)
```

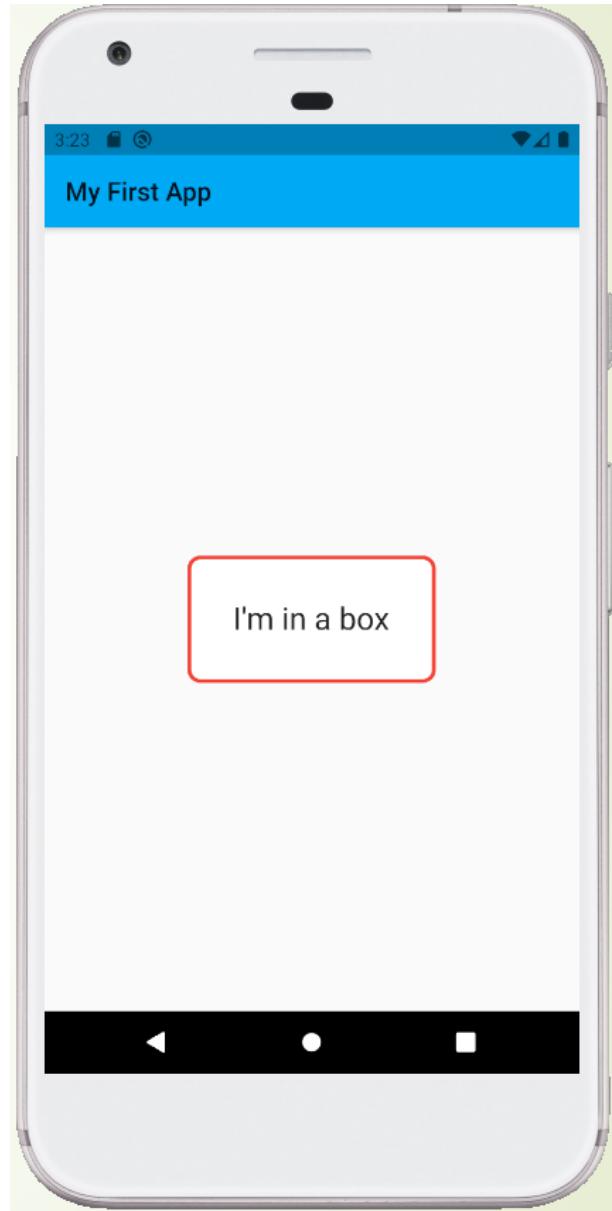
Container

- The Container widget lets you create a rectangular visual element. A container can be decorated with a [BoxDecoration](#), such as a **background**, a **border**, or a **shadow**.
- A Container can also have **margins**, **padding**, and constraints applied to its size.
- Url: <https://api.flutter.dev/flutter/widgets/Container-class.html>



Container

```
home: Scaffold(  
    appBar: AppBar(  
        title: Text('My First App'),  
    ), // AppBar  
    body: Center(  
        child: Container(  
            padding: EdgeInsets.all(32.0),  
            decoration: BoxDecoration(  
                color: Colors.white,  
                border: Border.all(color: Colors.red, width: 3),  
                borderRadius: BorderRadius.circular(10)  
            ), // BoxDecoration  
            child: Text(  
                'I\'m in a box',  
                style: TextStyle(fontSize: 24)  
            ), // Text  
        )), // Container // Center  
    ); // Scaffold // MaterialApp
```



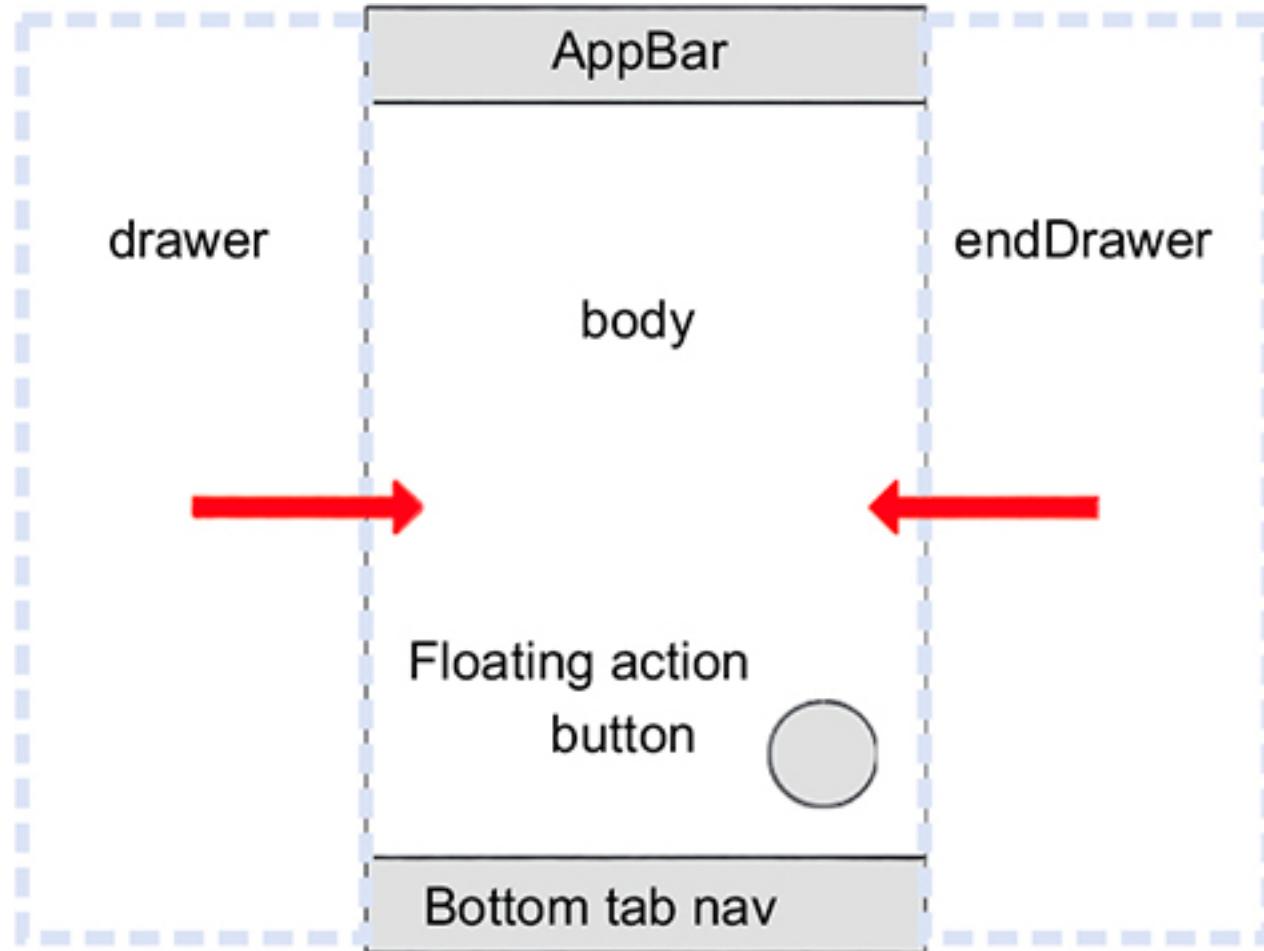
Using Material Components

- Flutter provides a number of widgets that help you build apps that follow Material Design.
- A Material app starts with the `MaterialApp` widget.

```
class BasicWidget extends StatelessWidget {  
  const BasicWidget({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    final fullNameController = TextEditingController();  
  
    return MaterialApp(  
      home: Scaffold(  
        appBar: AppBar(  
          title: const Text('Basic Widget'),  
        ), // AppBar  
        body: Center(  
          child: Container(...), // Container  
        ), // Center  
        floatingActionButton: const FloatingActionButton(  
          tooltip: 'Add',  
          onPressed: null,  
          child: Icon(Icons.add),  
        ), // FloatingActionButton  
      ), // Scaffold  
    ); // MaterialApp  
  }  
  
  void clickMe(String msg) {...}  
}
```

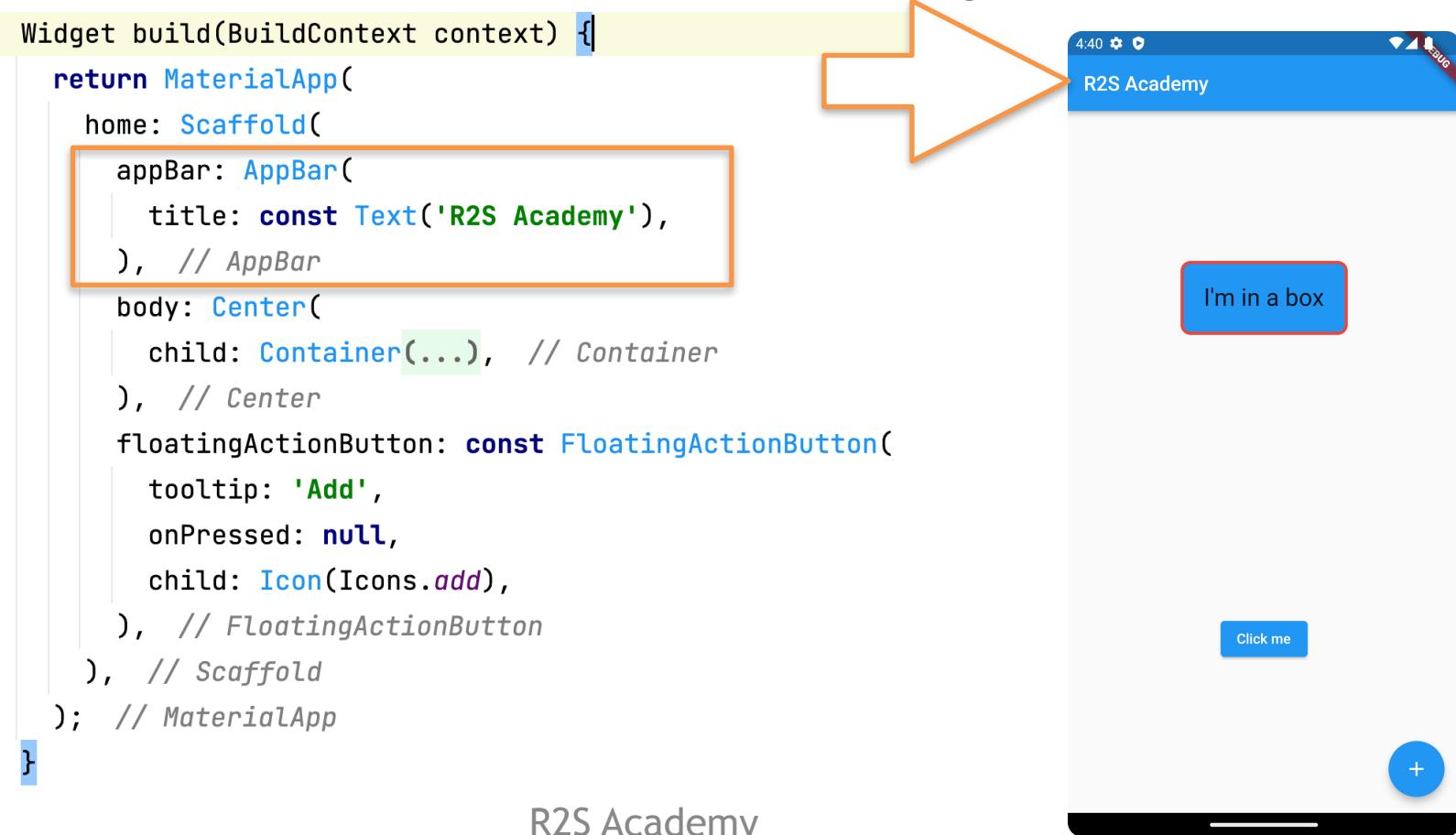
Scaffold

- Scaffold is a class which provides many widgets like **Drawer**, **SnackBar**, **BottomNavigationBar**, **FloatingActionButton**, **AppBar**, etc.
- Scaffold will expand or occupy the whole device screen.



Properties of Scaffold Class

- **appBar:** It displays a horizontal bar which mainly placed at the top of the *Scaffold*. *appBar* uses the widget *AppBar* which has its own properties like elevation, title, brightness, etc.



The screenshot shows an Android application interface. At the top, there is a blue navigation bar labeled "R2S Academy". Below it is a white content area. In the center of the content area is a red-bordered box containing the text "I'm in a box". In the bottom right corner of the content area is a blue floating action button with a white plus sign. In the bottom right corner of the entire screen is a black navigation bar with a white minus sign. An orange arrow points from the text "I'm in a box" in the code to the red-bordered box in the screenshot.

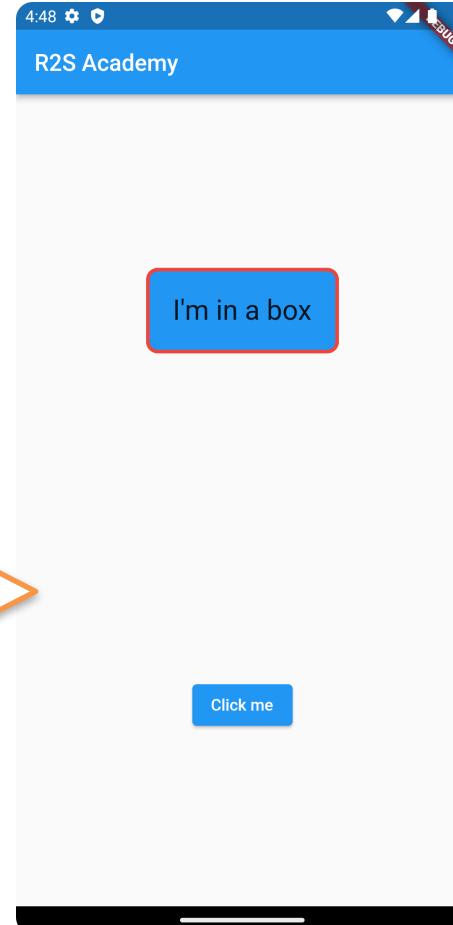
```
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(  
        title: const Text('R2S Academy'),  
      ), // AppBar  
      body: Center(  
        child: Container(...), // Container  
      ), // Center  
      floatingActionButton: const FloatingActionButton(  
        tooltip: 'Add',  
        onPressed: null,  
        child: Icon(Icons.add),  
      ), // FloatingActionButton  
    ), // Scaffold  
  ); // MaterialApp  
}
```

R2S Academy

Properties of Scaffold Class

- **body:** It will display the main or primary content in the Scaffold.
It is below the *appBar* and under the *floatingActionButton*.

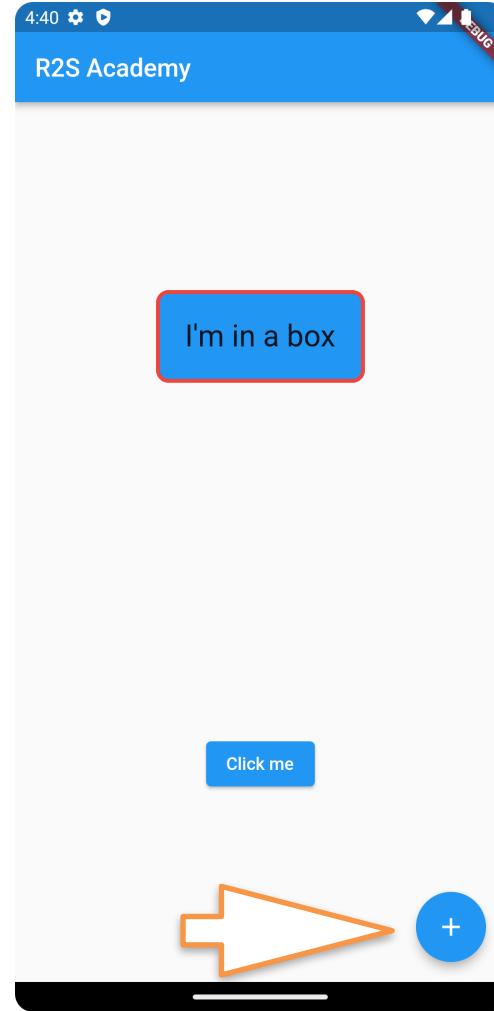
```
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(  
        title: const Text('R2S Academy'),  
      ), // AppBar  
      body: Center(...), // Center  
    ), // Scaffold  
  ); // MaterialApp  
}
```



Properties of Scaffold Class

- floatingActionButton:** It is a button that is placed at the right bottom corner by default.

```
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('R2S Academy'),
      ), // AppBar
      body: Center(
        child: Container(...), // Container
      ), // Center
      floatingActionButton: const FloatingActionButton(
        tooltip: 'Add',
        onPressed: null,
        child: Icon(Icons.add),
      ), // FloatingActionButton
    ), // Scaffold
  ); // MaterialApp
}
```



Properties of Scaffold Class

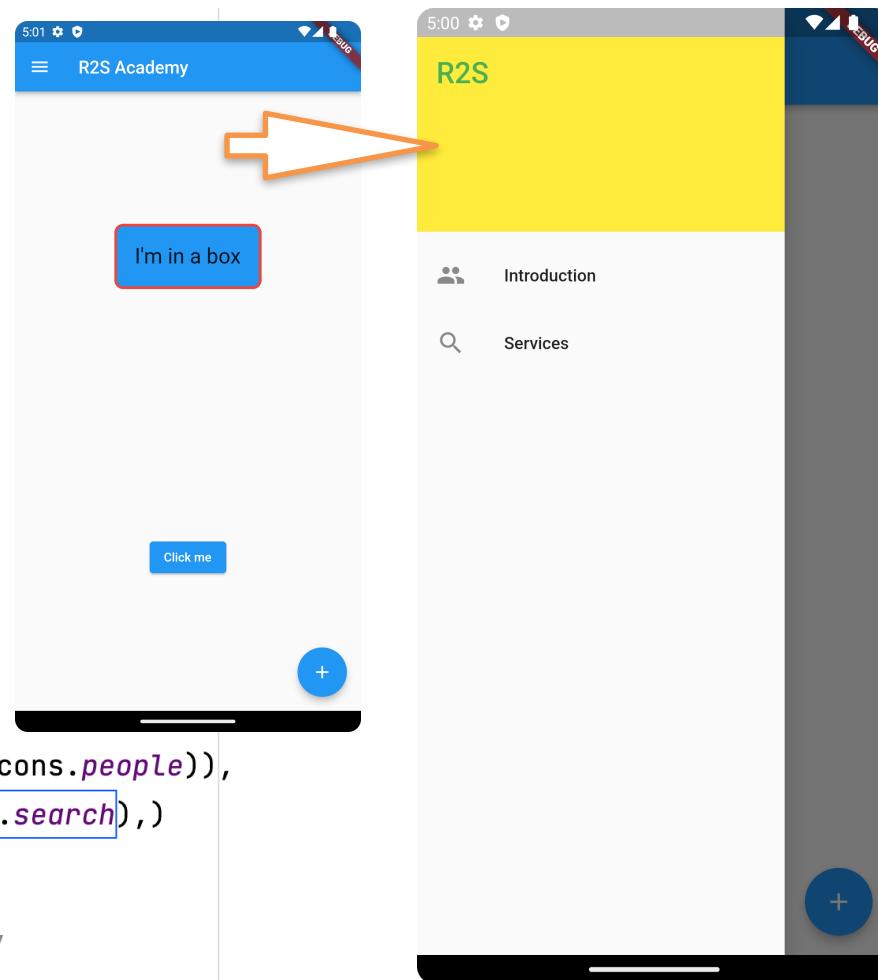
- drawer:** It is a slider menu or a panel which is displayed at the side of the Scaffold. The user has to swipe left to right or right to left according to the action defined to access the drawer menu.

```

drawer: Drawer(
  child: ListView(
    children: const [
      DrawerHeader(
        decoration: BoxDecoration(
          color: Colors.yellow
        ), // BoxDecoration
        child: Text(
          'R2S',
          style: TextStyle(
            color: Colors.green,
            fontSize: 24
          ), // TextStyle
          ), // Text
      ), // DrawerHeader
      ListTile(title: Text('Introduction'), leading: Icon(Icons.people)),
      ListTile(title: Text('Services'), leading: Icon(Icons.search),)
    ],
  ), // ListView
), // Drawer

```

R2S Academy



Properties of Scaffold Class

- **bottomNavigationBar:** It is like a menu at the bottom of the Scaffold. We have seen this navigationbar in most of the applications. We can add multiple icons or texts or both in the bar as items.

```
bottomNavigationBar: BottomNavigationBar(
```

```
  currentIndex: 0,
  fixedColor: Colors.green,
  items: const [
    BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
    BottomNavigationBarItem(icon: Icon(Icons.search), label: 'Search'),
    BottomNavigationBarItem(
      icon: Icon(Icons.account_circle), label: 'Profile') // BottomNa
  ],
), // BottomNavigationBar
```

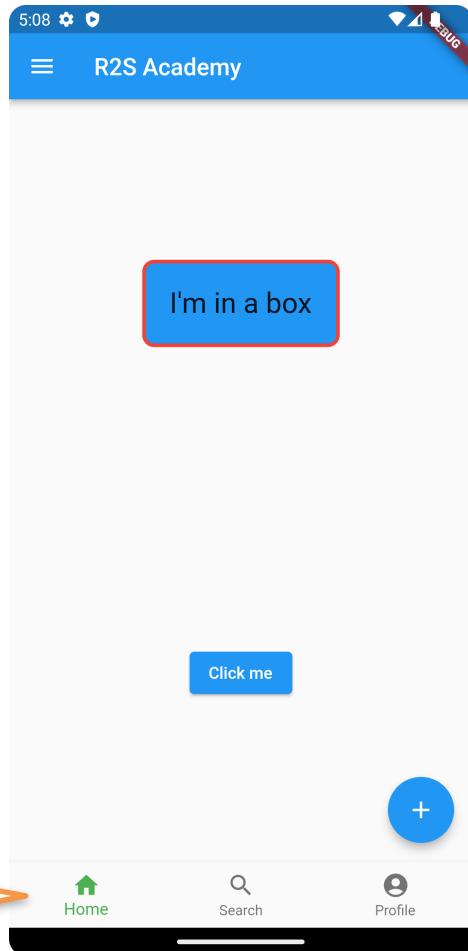
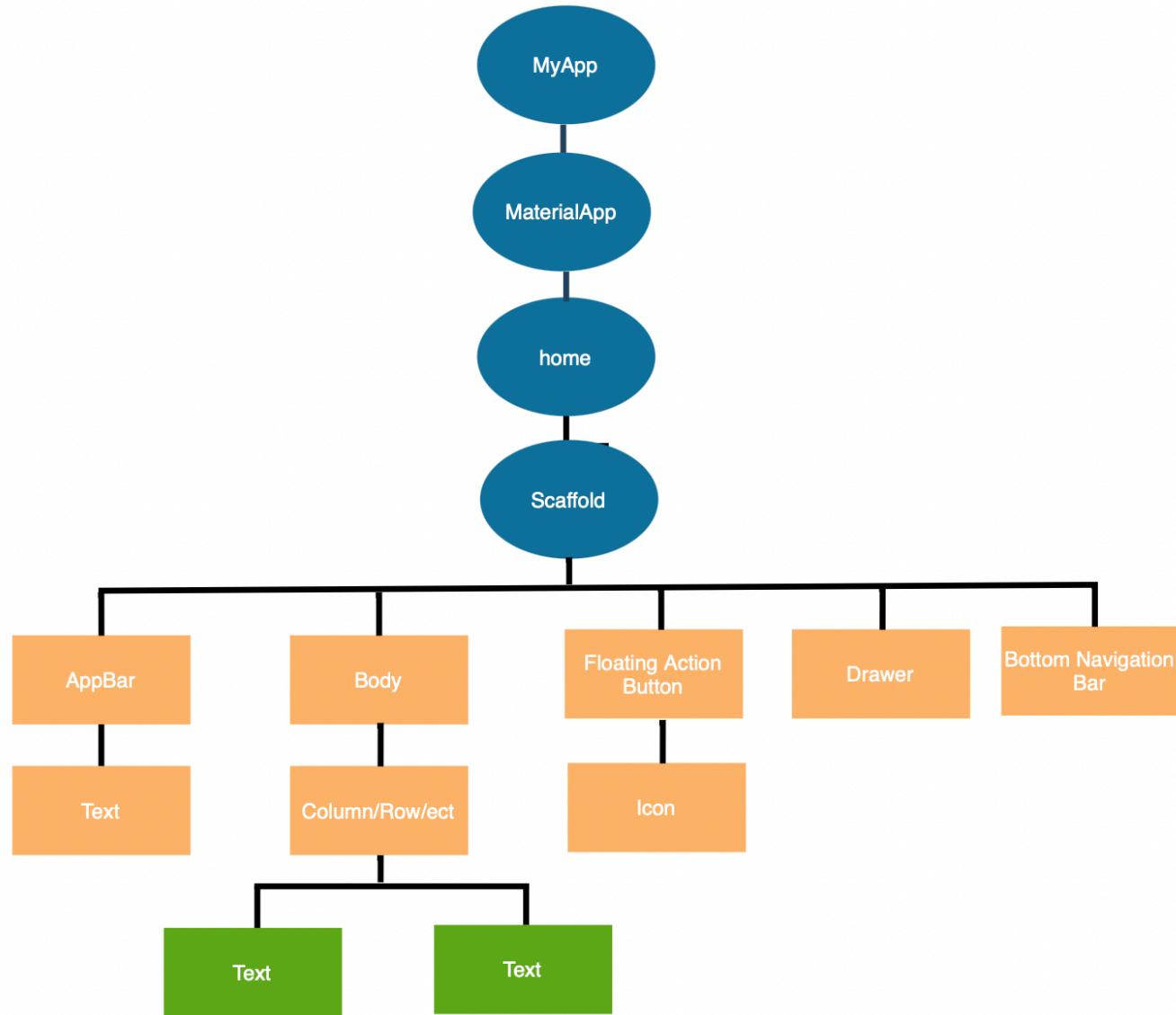


Diagram of the widget tree



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

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Flutter layout demo'),
        ), // AppBar
        body: Column(
          children: [
            imageSection,
            titleSection,
            buttonSection,
            textSection,
          ],
        ), // Column
        floatingActionButton: const FloatingActionButton(
          tooltip: 'Add',
          onPressed: null,
          child: Icon(Icons.add),
        ), // FloatingActionButton
      ), // Scaffold
    ); // MaterialApp
}
```