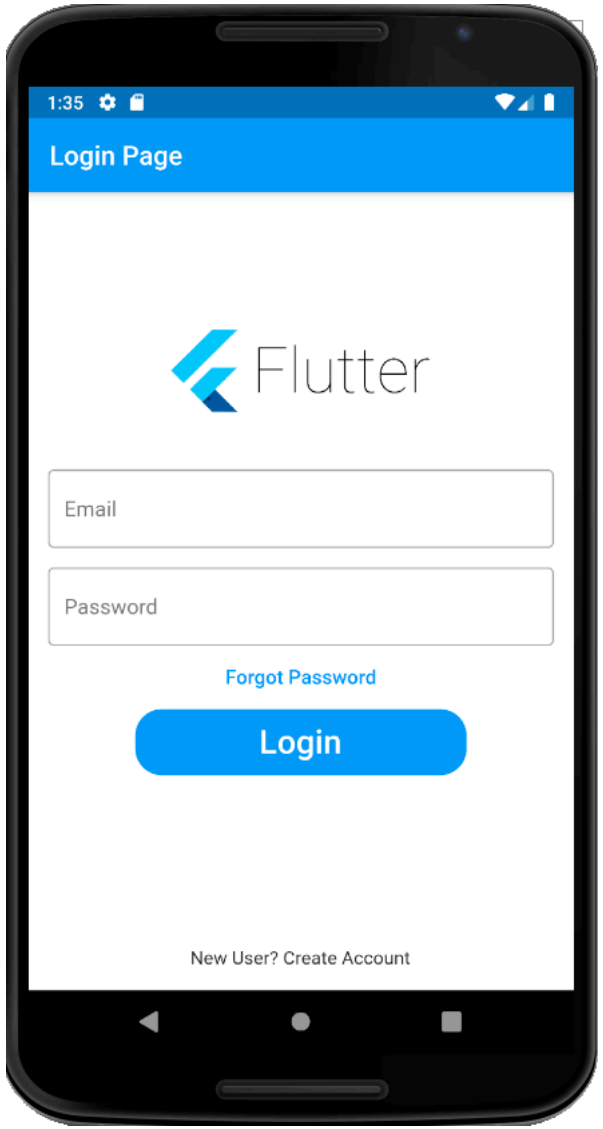
R2S
Resource Software Solution

Basic Widgets

Contents

1. Introduction
2. Widgets
3. Material Design

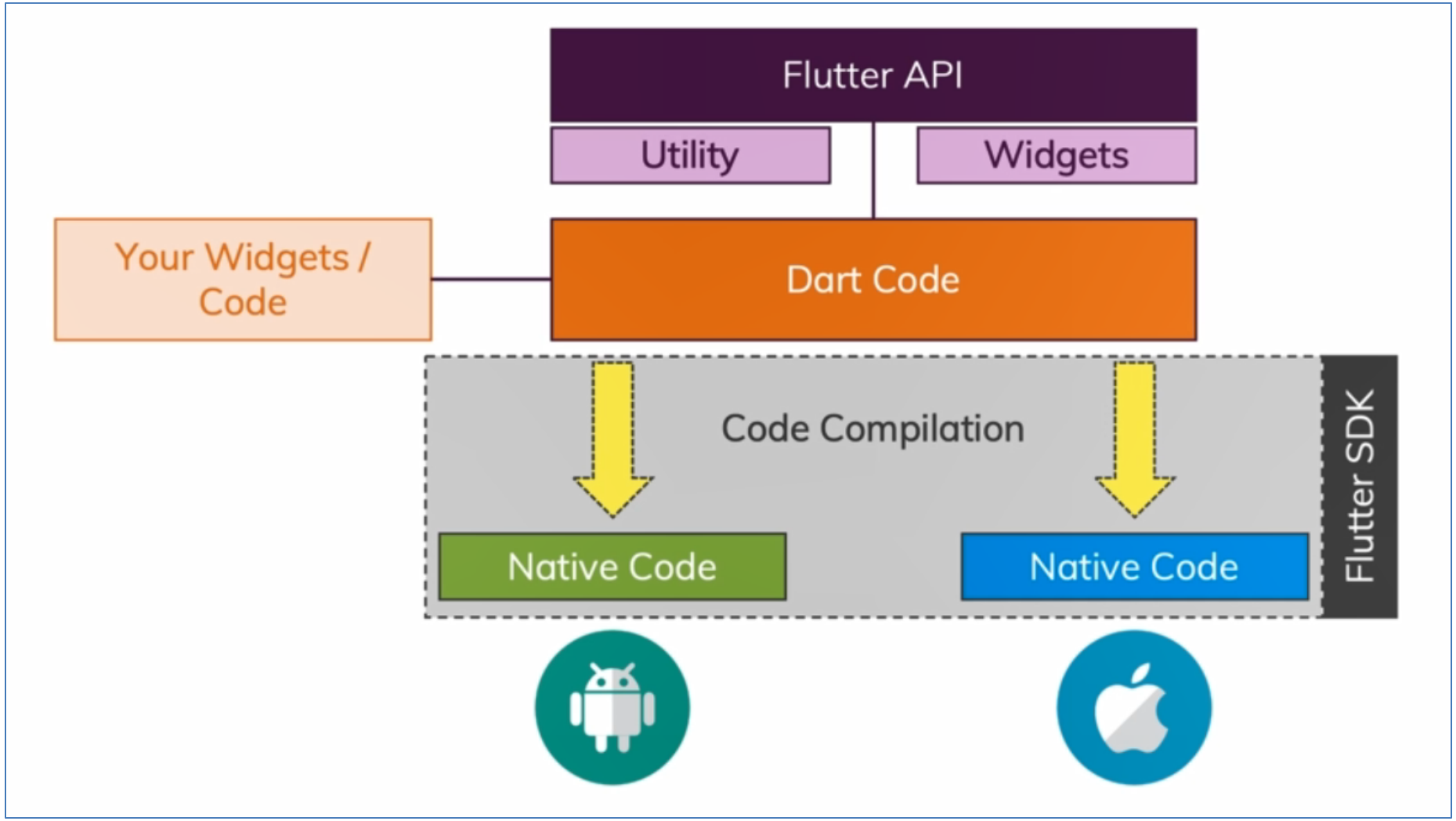


Introduction (1)

- Fluter is a “tool” that allows you to build Hybrid (iOS, Android) apps with one programming language. Working in **one project**, **one source code**, but get **two different apps** as a result.
- Normally, for native apps, you have to build two projects with different languages.
  - Native **iOS app**: **Swift** or Objective C
  - Native **Android app**: Kotlin or **Java**

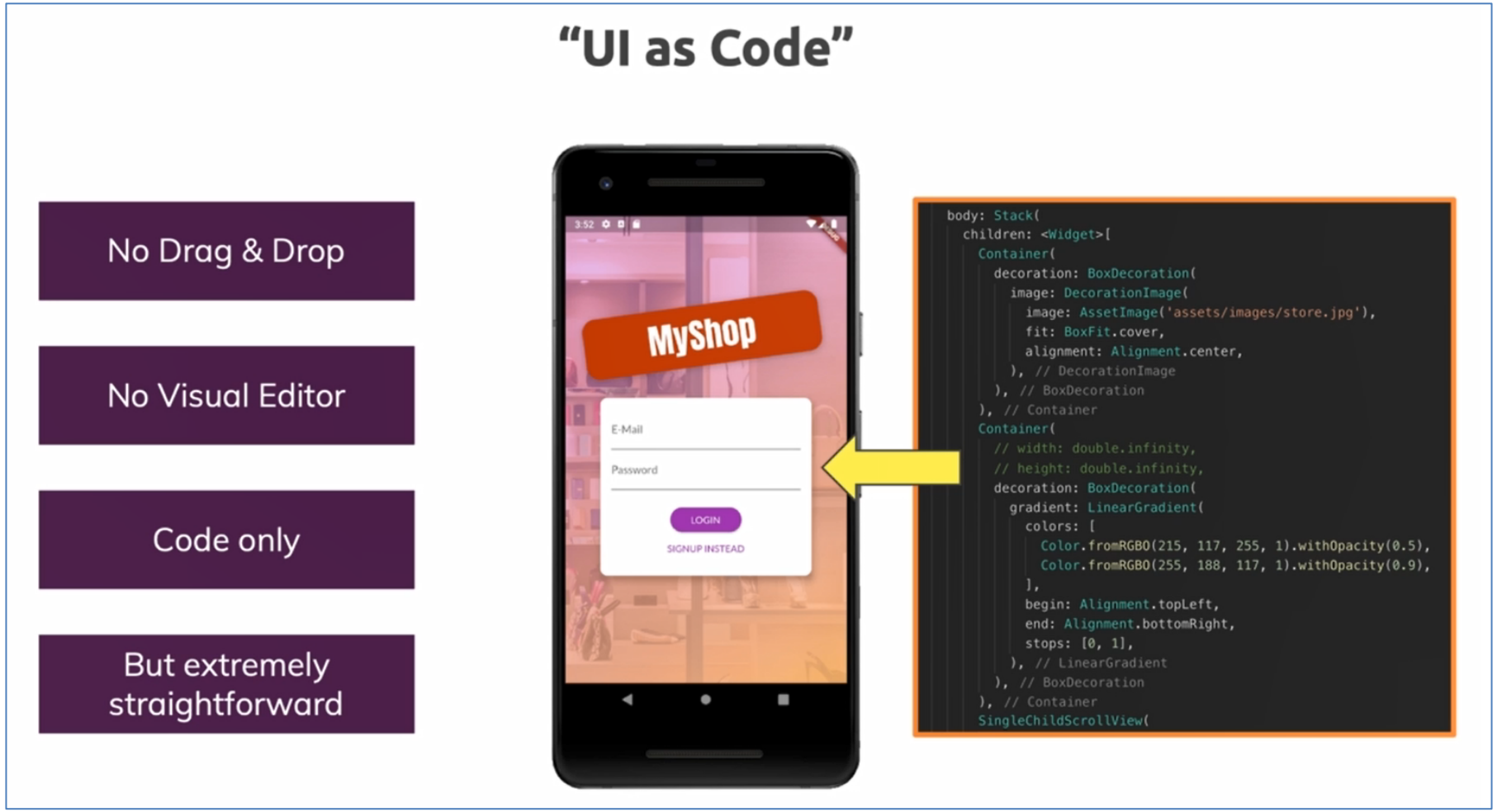
Introduction (2)

- Flutter “transformed” to Native App?



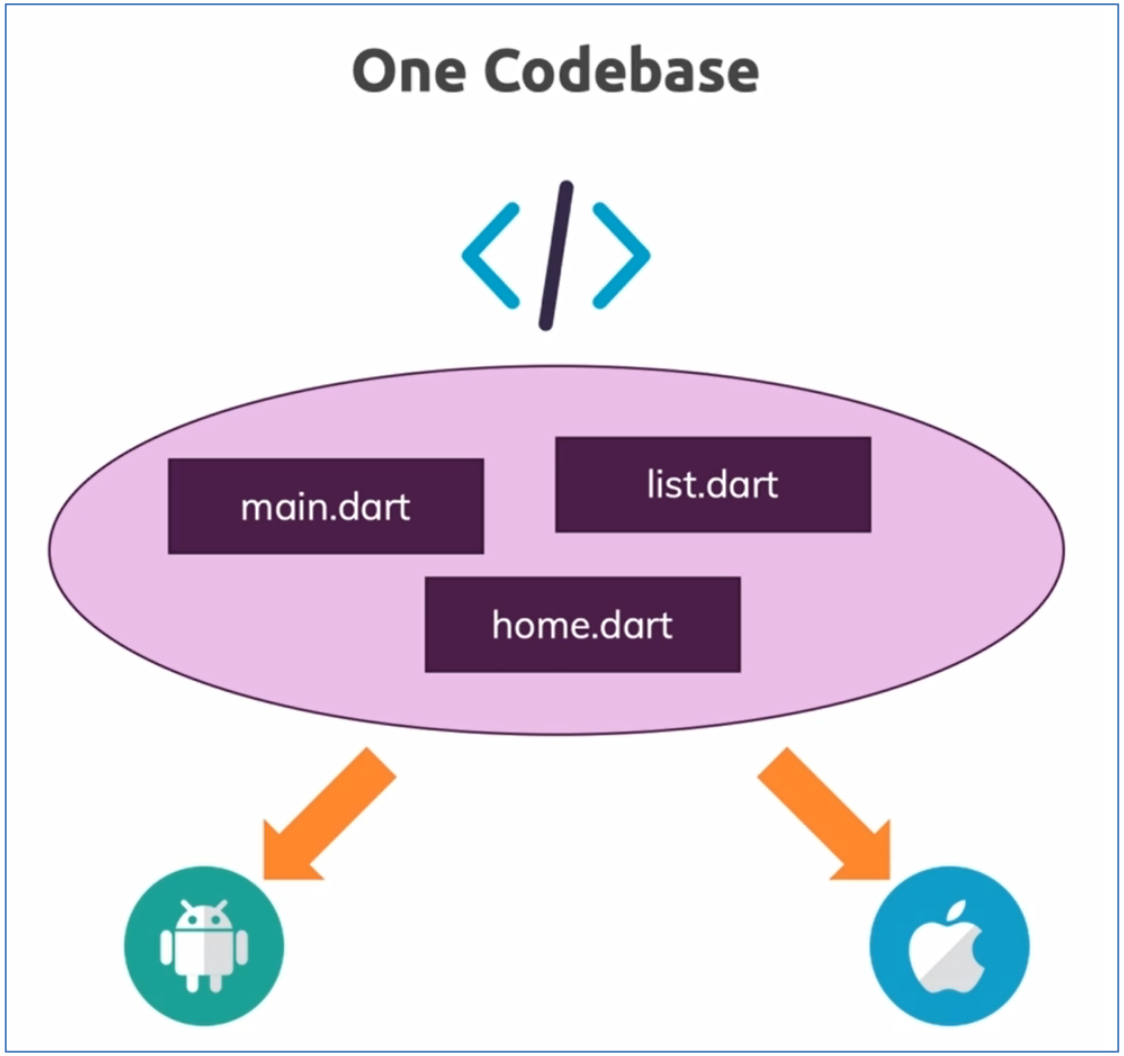
Introduction (3)

- Flutter app architecture



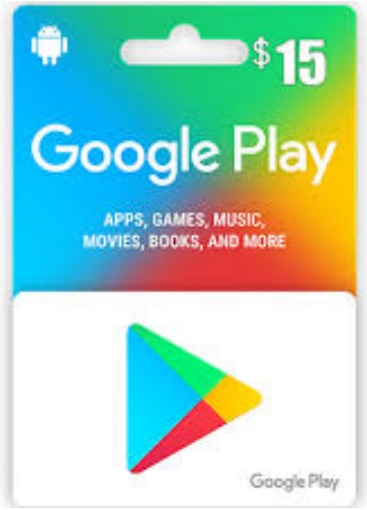
Introduction (4)

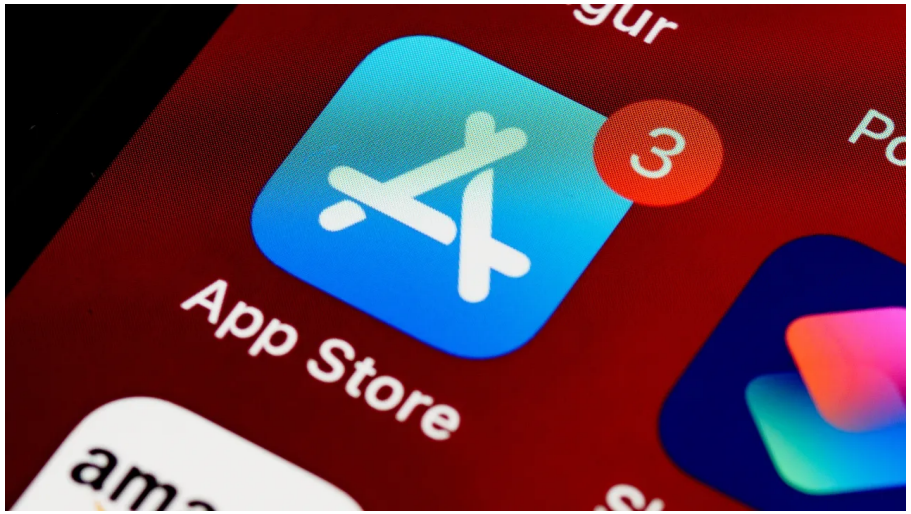
- Flutter app architecture



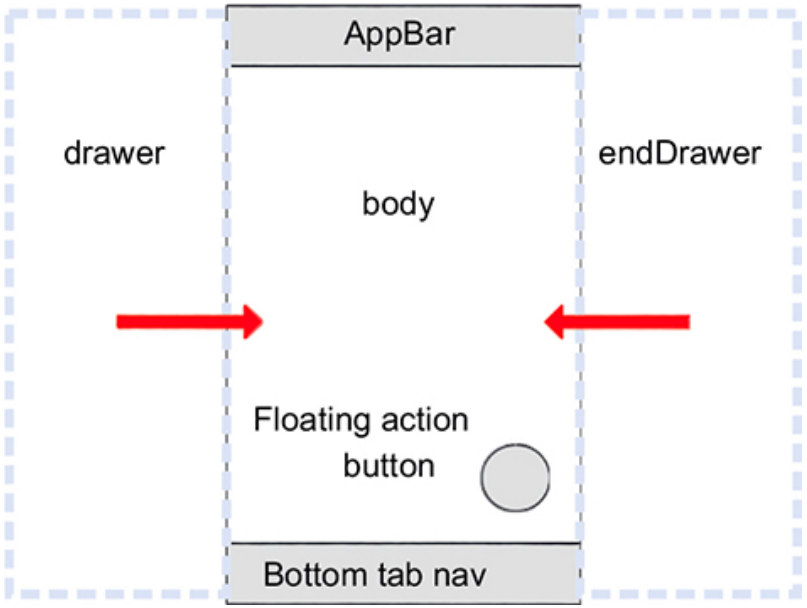
Flutter Setup

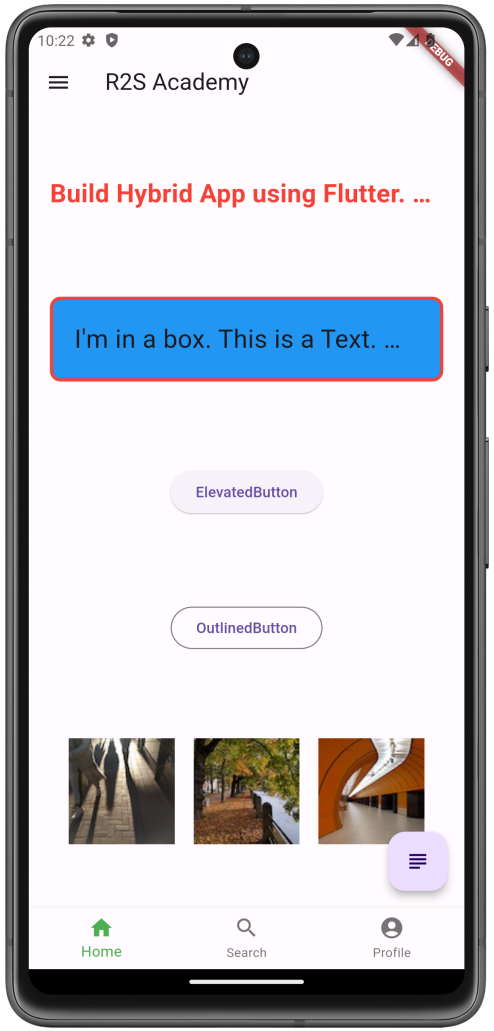
- Google Play (Android): Starting **August 31 2024**, new apps and app updates must target **Android 14 (API level 34)** to be submitted to Google Play
- Apple App Store (iOS): Please note that as of **April 2024** all iOS and iPadOS apps submitted to the App Store must be built with the **iOS 17 SDK**. Starting **April 2025**, all iOS and iPadOS apps uploaded to App Store Connect must be built with the **iOS 18** SDK.

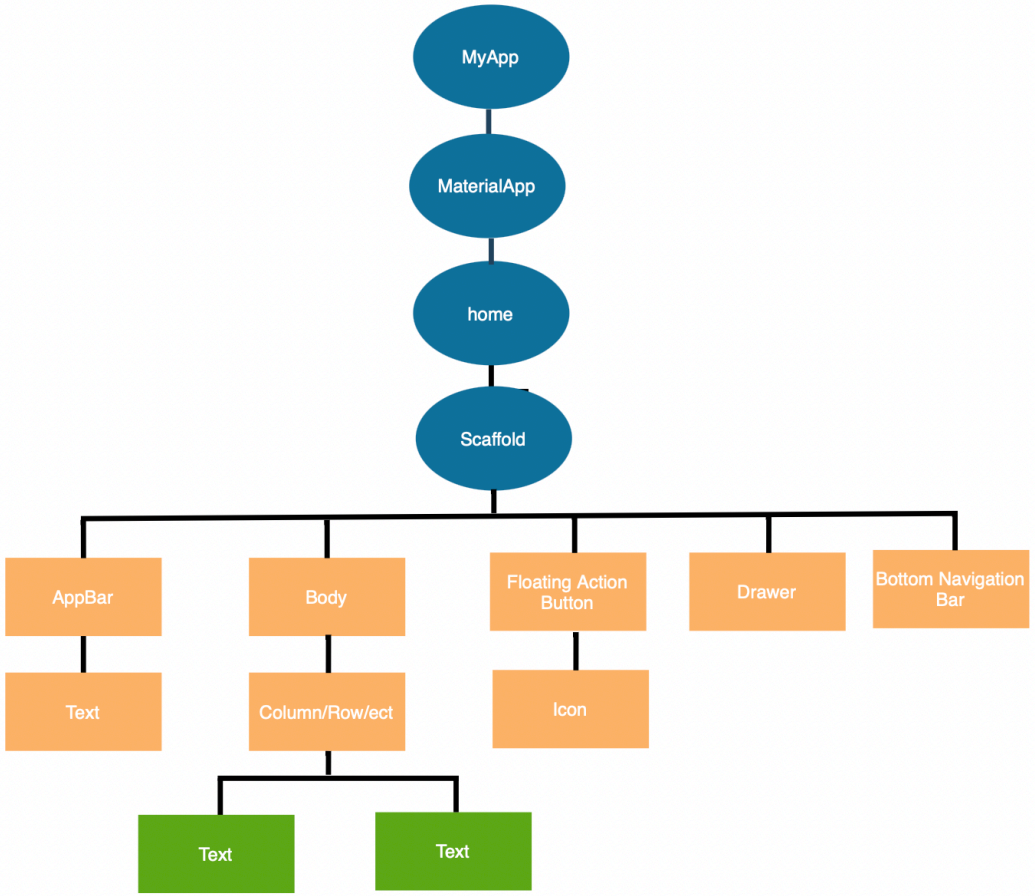




Widget Tree

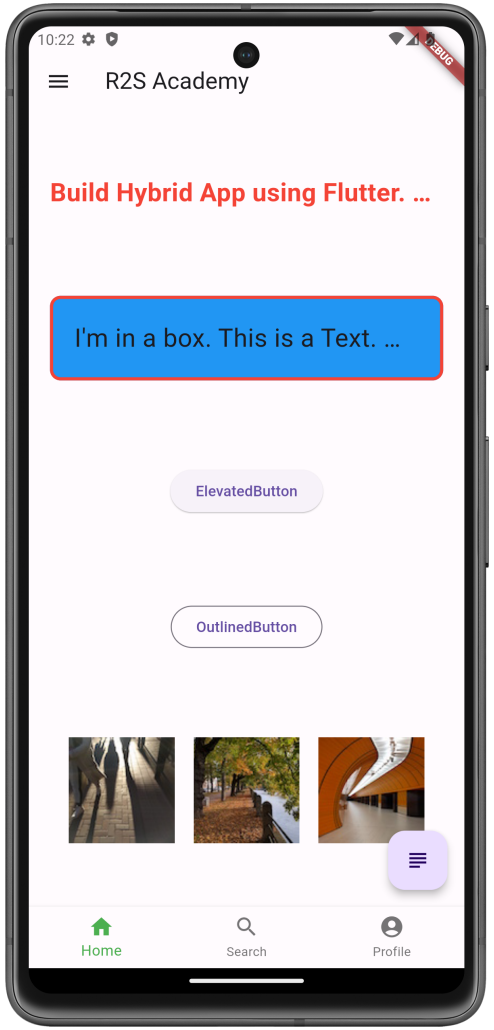

AppBar
drawer
body
endDrawer
Floating action button
Bottom tab nav


10:22
R2S Academy
Build Hybrid App using Flutter. ...
I'm in a box. This is a Text. ...
ElevatedButton
OutlinedButton
Home
Search
Profile


MyApp
MaterialApp
home
Scaffold
AppBar
Body
Floating Action Button
Drawer
Bottom Navigation Bar
Text
Column/Row/ect
Icon
Text
Text

Material Components

- A Material app starts with the **MaterialApp** widget.

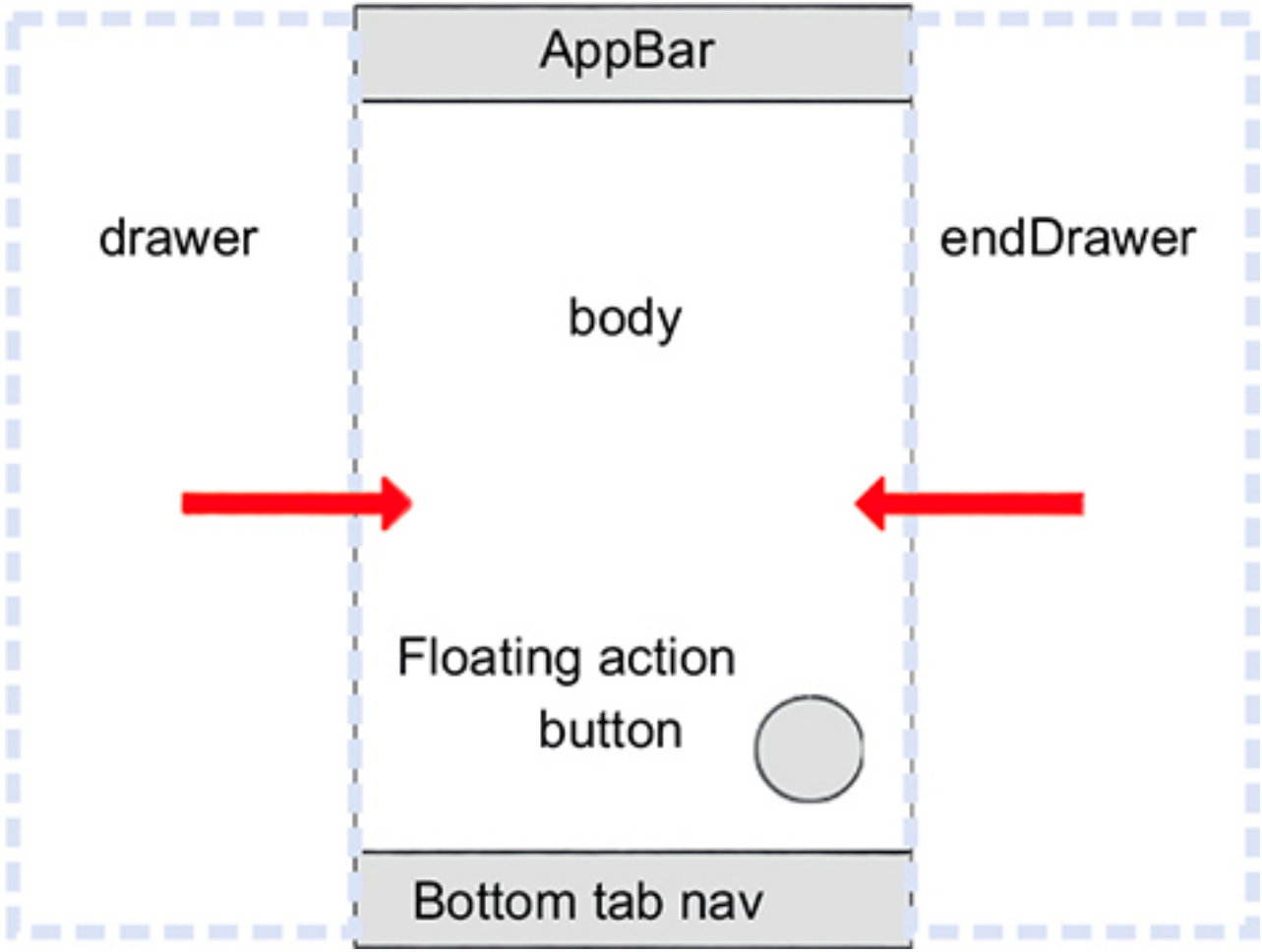



```dart
class BasicWidget extends StatelessWidget {
  const BasicWidget({super.key});
  @override
  Widget build(BuildContext context) {
    final fullNameController = TextEditingController();
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Basic Widget'),
        ),  // 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 (1)

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

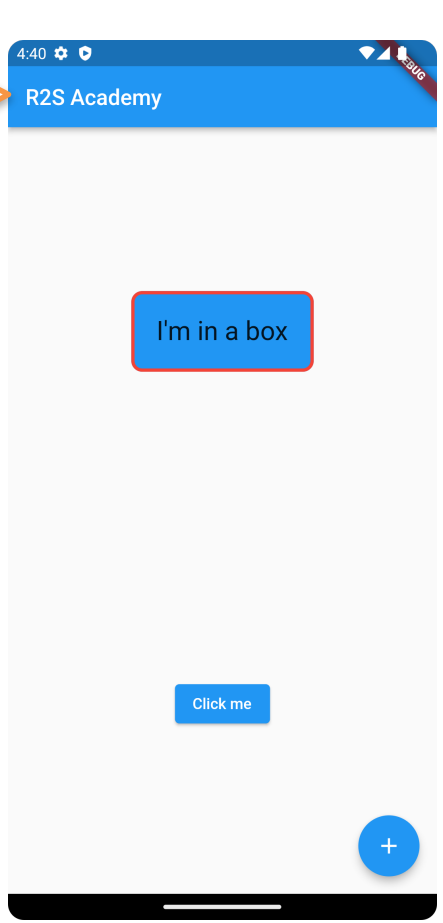
```dart
class BasicWidget extends StatelessWidget {
  const BasicWidget({super.key});
  @override
  Widget build(BuildContext context) {
    final fullNameController = TextEditingController();
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Basic Widget'),
        ),  // 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 (2)

- **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.

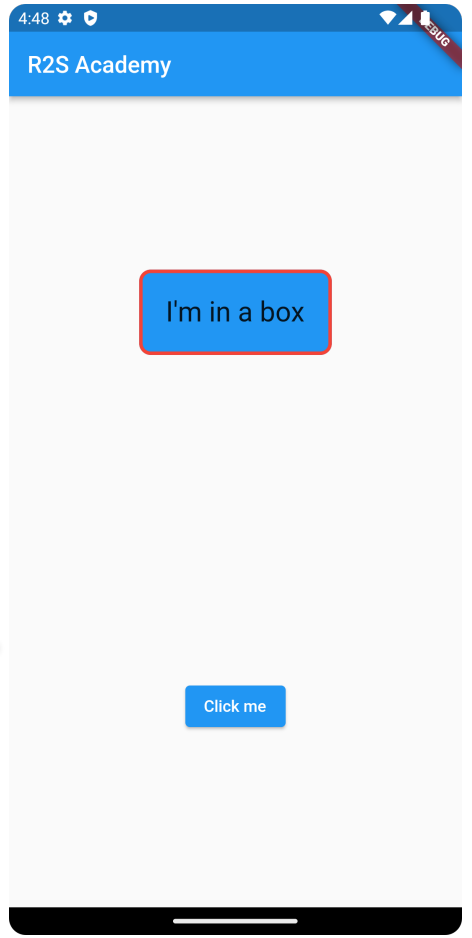
```
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('R2S Academy'),
      ),  // AppBar
      body: Center(
        child: Container(...),  // Container
      ),  // Center
      floatingActionButton: const FloatingActionButton(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



Scaffold (3)

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

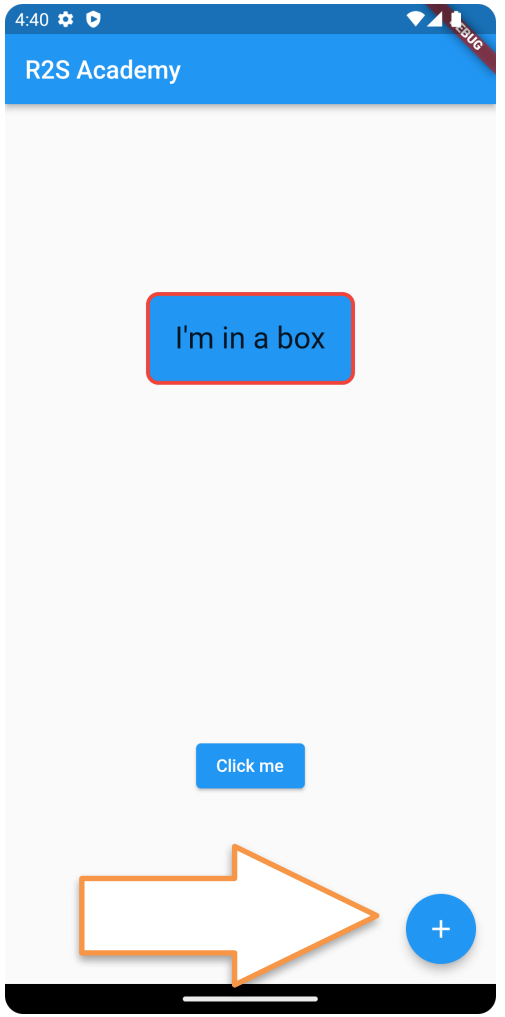
```
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('R2S Academy'),
      ),  // AppBar
      body: Center(...),  // Center
    ),  // Scaffold
  );  // MaterialApp
}
```



Scaffold (4)

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

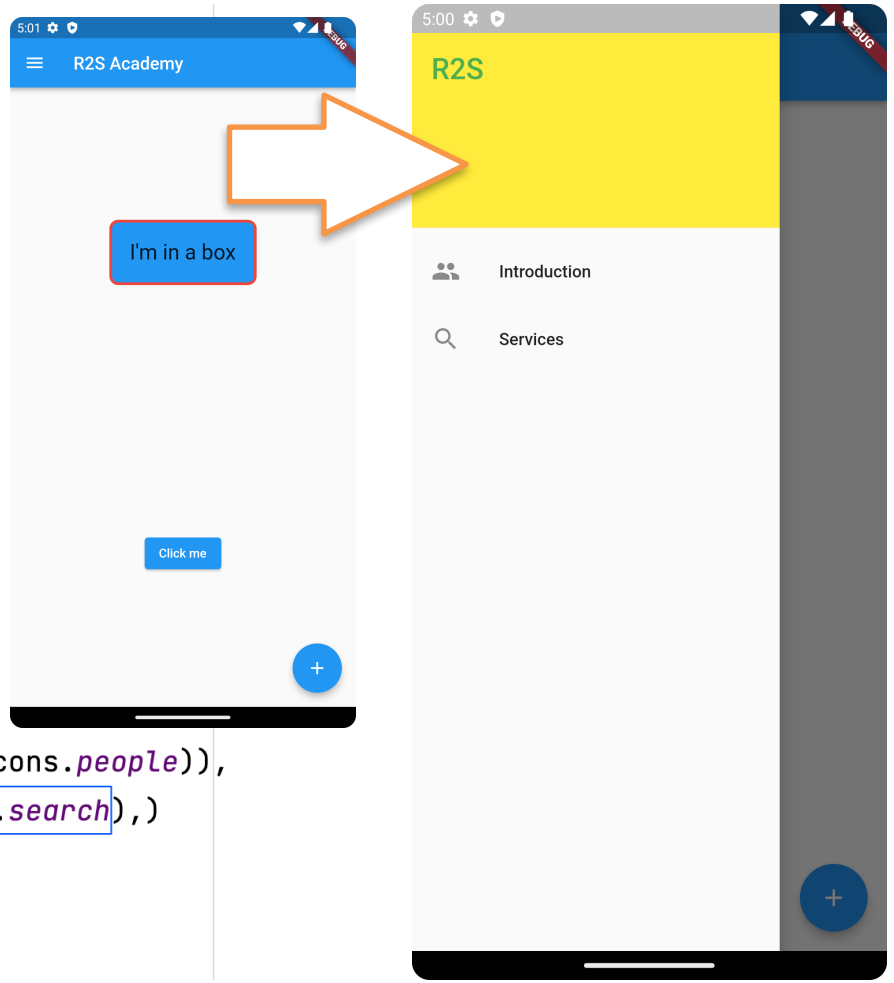
```
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('R2S Academy'),
      ),  // AppBar
      body: Center(
        child: Container(...),  // Container
      ),  // Center
      floatingActionButton: const FloatingActionButton(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



Scaffold (5)

- **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.

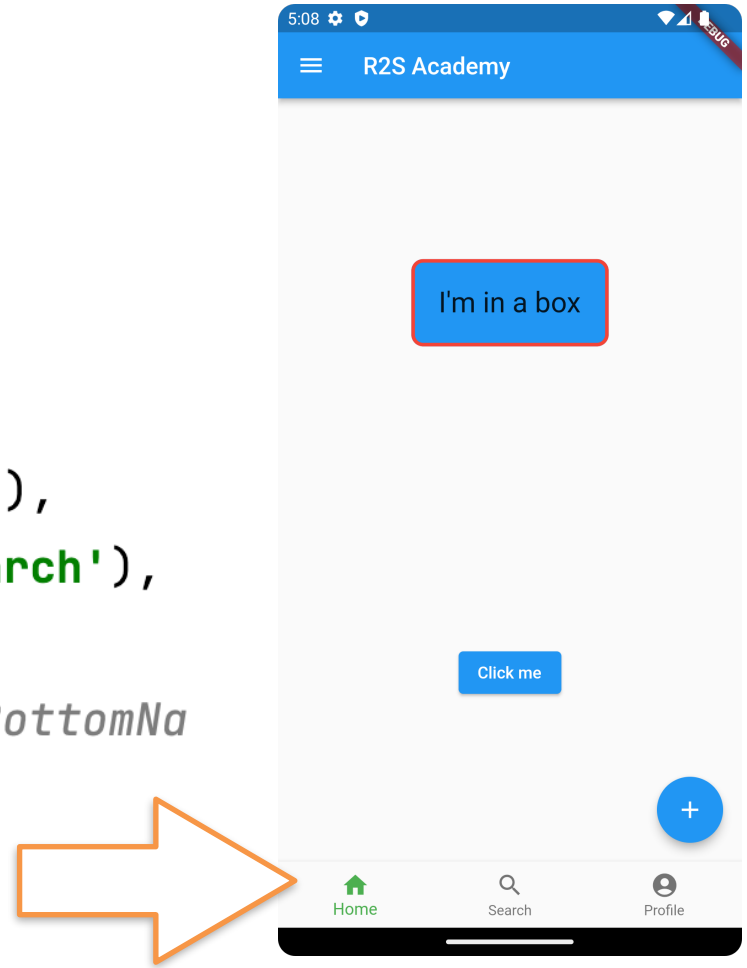
```
drawer: Drawer(
  child: ListView(
    children: const [
      DrawerHeader(
        decoration: BoxDecoration(
          color: Colors.yellow
        ),  // BoxDecoration
          child: Text(
              'R2S',
            style: TextStyle(
              color: Colors.green,
              fontSize: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



Scaffold (6)

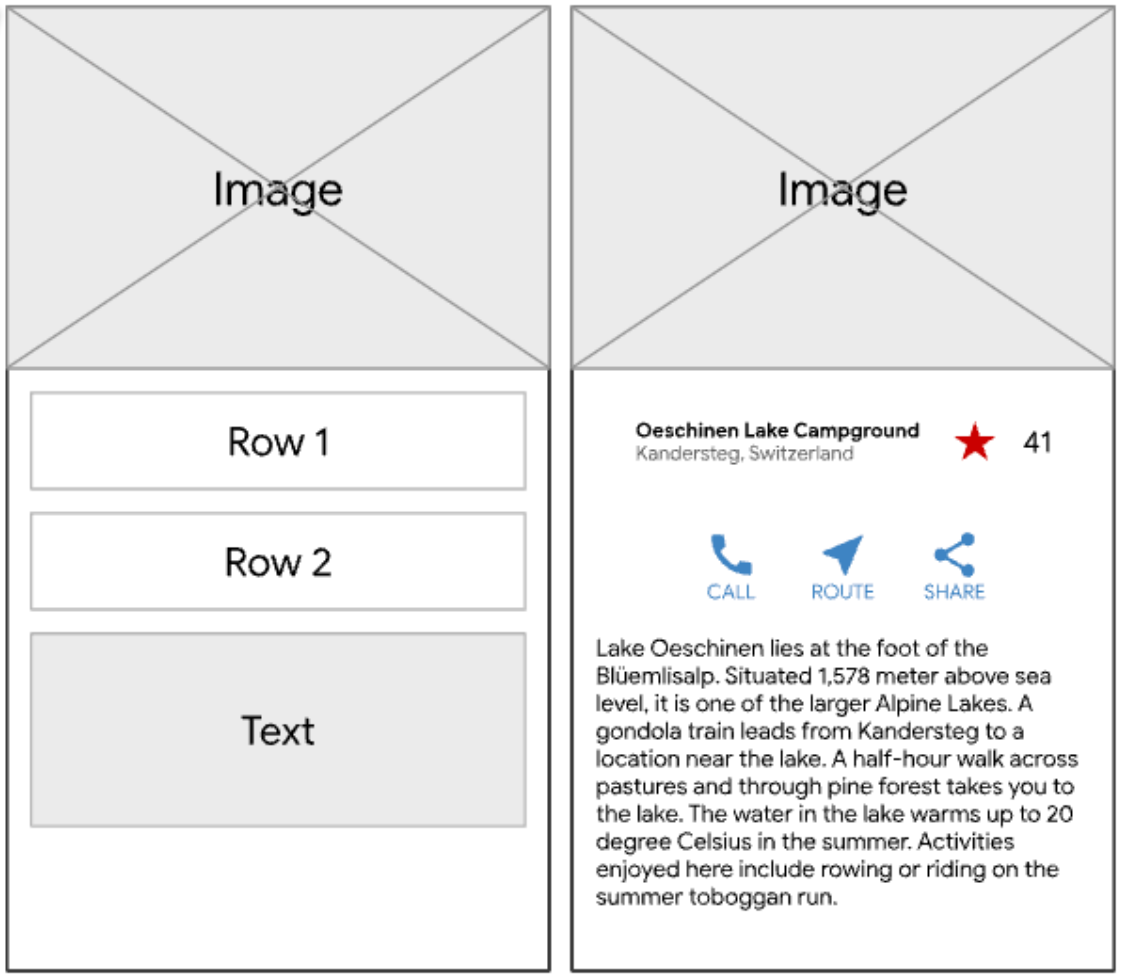
- **bottomNavigationBar**: It is like a menu at the bottom of the Scaffold. We have seen this navigationbar in most of the applications. We can add multiple icons or texts or both in the bar as items.

```
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
```



Demo (1)

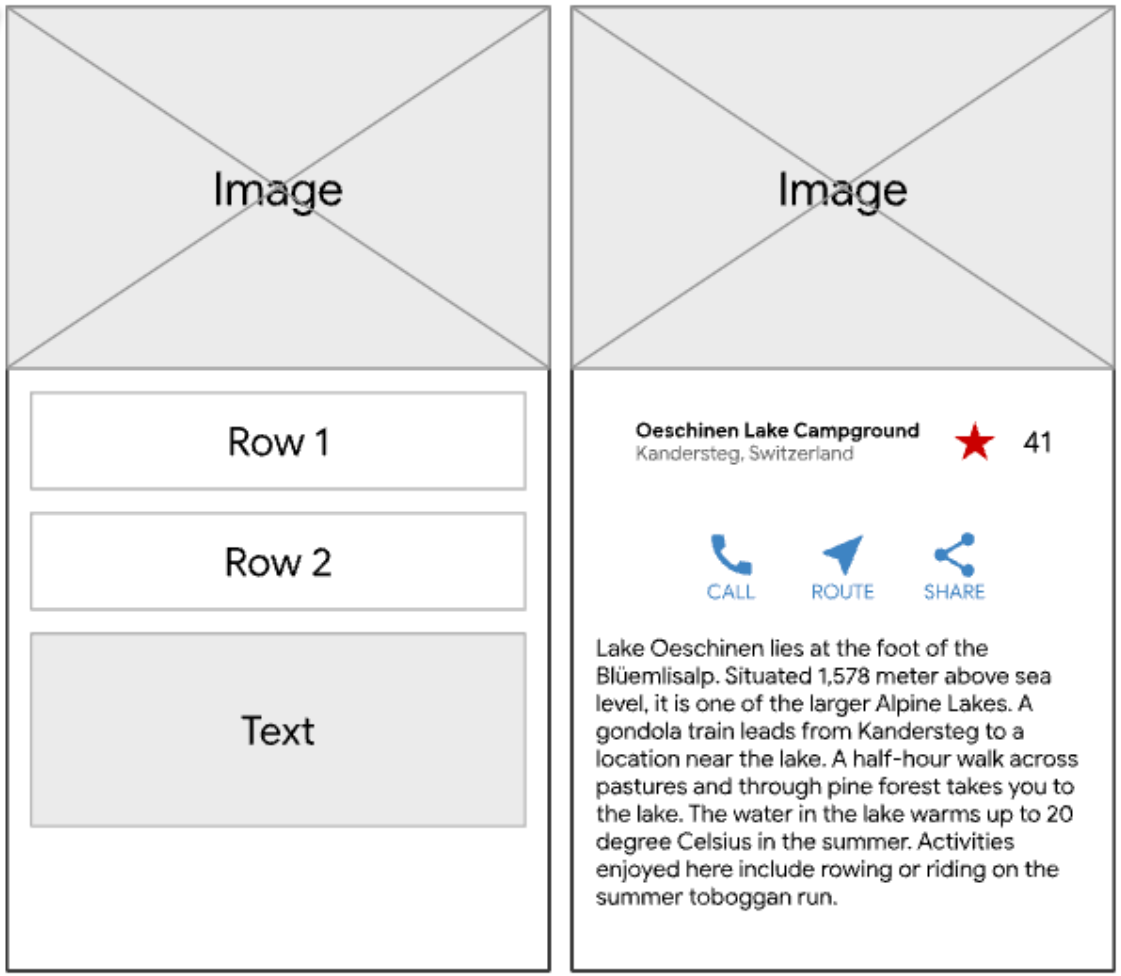
Column



Favorite page: The text block as sketch and prototype UI

Demo (2)

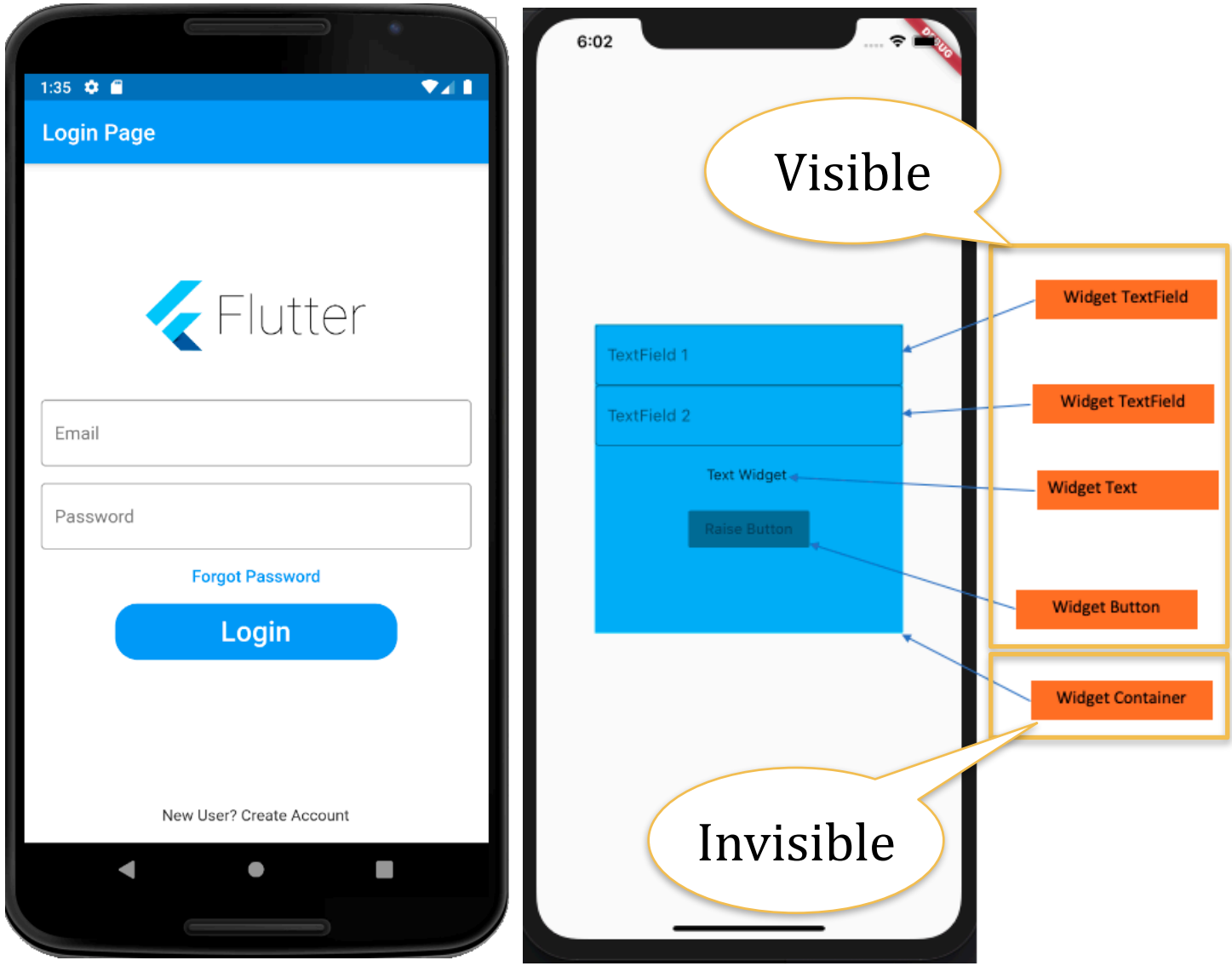
Column



Favorite page: The text block as sketch and prototype UI

Widgets in Flutter (1)

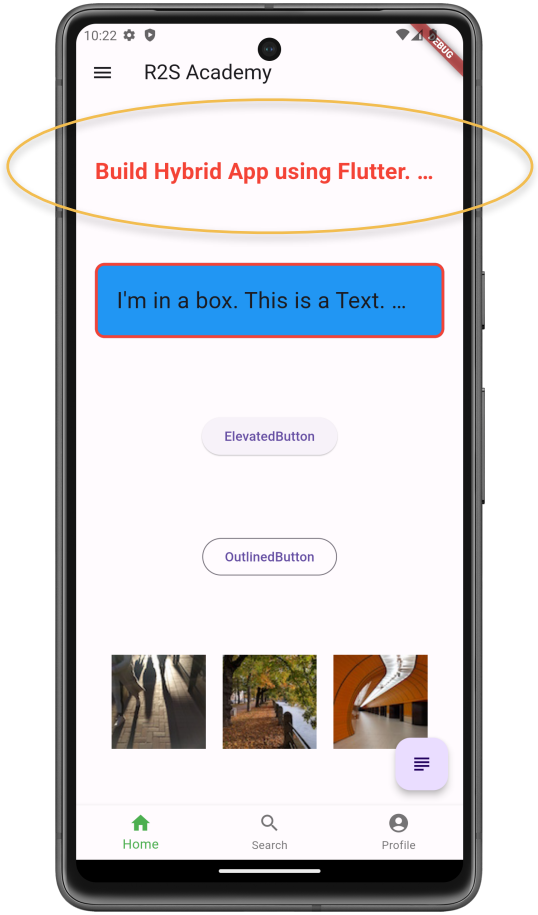
- Each **element** on a **screen** of the Flutter app is a **widget**.
- Types of Widget:
  - **Visible** (Input and Output)
  - **Invisible** (Layout)



Visible Widget - Text (1)

- It allows us to display a string of text with a single line in our application.

```
Text(
  data: 'Build Hybrid App using Flutter. My name is Dart',
  style: TextStyle(
    fontSize: 24,
    color: Colors.red,
    fontWeight: FontWeight.bold
  ),
  overflow: TextOverflow.ellipsis,
)
```

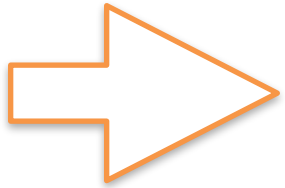


Visible Widget - Text (2)

- The Text widget has an **overflow** property that handles how text should be displayed when it overflows the space allocated to it
- TextOverFlow parameters
  1. **Ellipsis**: This will display an ellipsis (…) to indicate that the text has overflowed. It's commonly used to signify that there is more text that isn't being displayed.
  2. **Fade**: This will fade the overflowing text to transparent
  3. **Visible**: This will render the text outside the container, which means the overflowed text will be visible, potentially overlapping other widgets.
  4. **Clip**: This will clip the overflowing text to fix its container.

Visible Widget - Text (3)

```
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
```

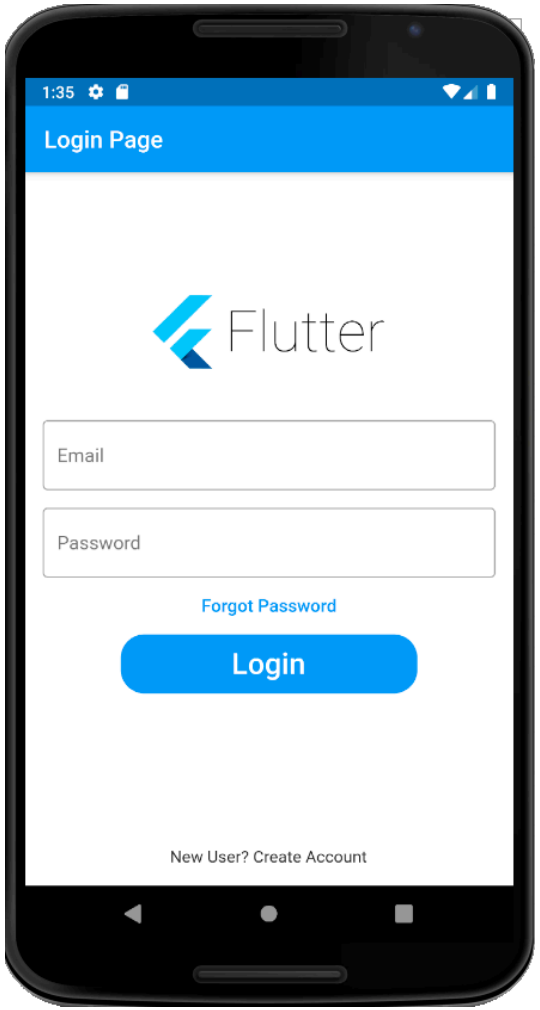


Hello *beautiful* **world**

Visible Widget - Button

- This widget allows you to perform some action on click
- We uses a type of buttons like blow

Old Widget	New Widget
FlatButton	**TextButton**
RaisedButton	**ElevatedButton**
OutlineButton	**OutlinedButton**

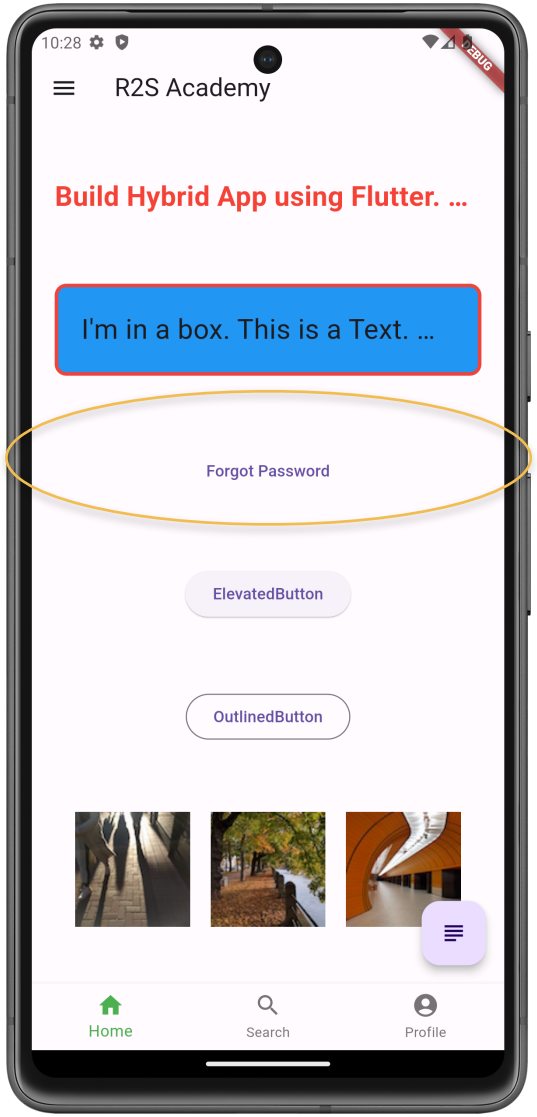


Visible Widget - TextButton

- Basic usage

```
TextButton(
 onPressed: () {
            var msg = 'This is OutlinedButton';
            showMessage(msg);
        },
 child: const Text('Forgot Password'),
)
```

```
void showMessage(String msg) {
  // body
}
```

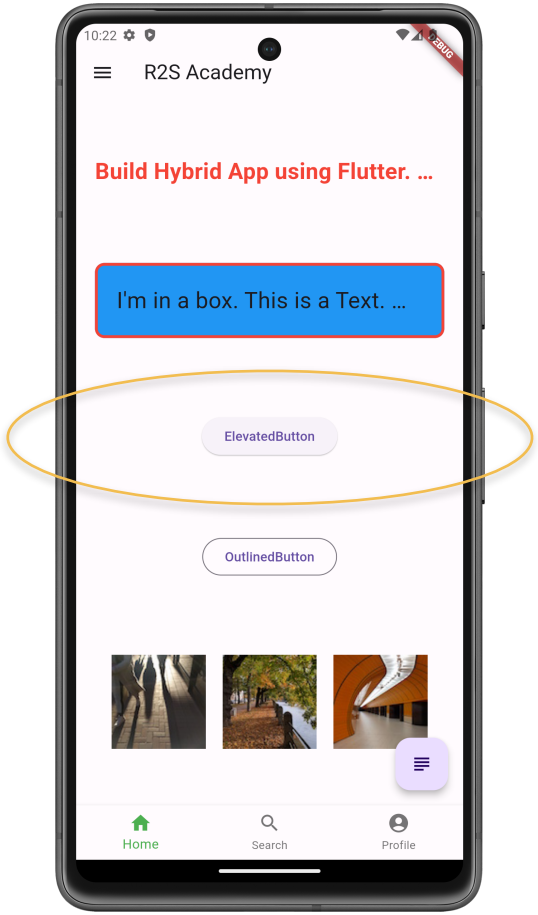


Visible Widget - ElevatedButton

- Let's take a look at a basic use of ElevatedButton

```
ElevatedButton(
   onPressed: () => showMessage('This is ElevatedButton'),
   child: const Text('ElevatedButton')
)
```

```
void showMessage(String msg) {
  // body
}
```

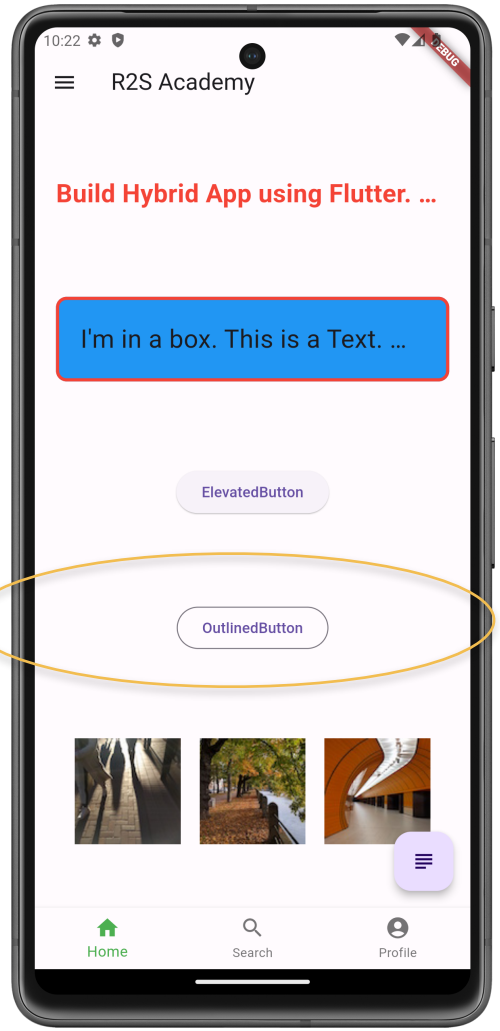


Visible Widget - OutlinedButton

- Let's take a look at a basic use of OutlinedButton

```
OutlinedButton(
  onPressed: saveOrder,
  child: const Text('Save')
)
```

```
void saveOrder() {
  // body
}
```



Visible Widget - Button

- Key Differences

```
ElevatedButton(
   onPressed: () => showMessage('This is ElevatedButton'),
   child: const Text('ElevatedButton')
)
```

```
OutlinedButton(
  onPressed: saveOrder,
  child: const Text('Save')
)
```

```
TextButton(
 onPressed: () {
                var msg = 'This is OutlinedButton';
                showMessage(msg);
            },
 child: const Text('Forgot Password'),
)
```

- Conclusion:
  - (**onPressed: saveOrder)** when the function signature matches and no additional parameters.
  - **(onPressed: () => showMessage())** when you need to pass parameters to the function.
  - **(onPressed: () { })** when you have multiple statements.

Visible Widget - Image

- This widget holds the image which can fetch it from multiple sources like from the asset folder or directly from the URL.

```
class MyHomePage extends StatelessWidget {

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(this.title),
    ),
    body: Center(
      child: Image.asset('images/computer.png'),
    ),
  );
 }
}
```

Visible Widget - Icon

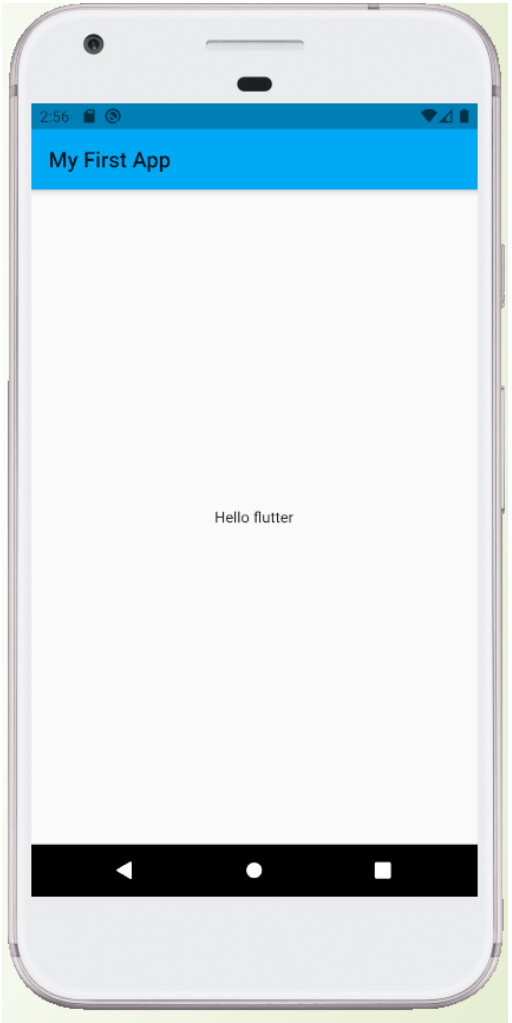
- This widget acts as a container for storing the Icon in the Flutter

```
Icon(
        Icons.audiotrack,
        size: 100.0,
        color: Colors.green,
),
Icon(
        Icons.account_balance,
        size: 100.0,
        color: Colors.red,
),
Icon(
        Icons.airplanemode_active,
        size: 200.0,
        color: Colors.deepPurple
),
```

Invisible Widget - Center

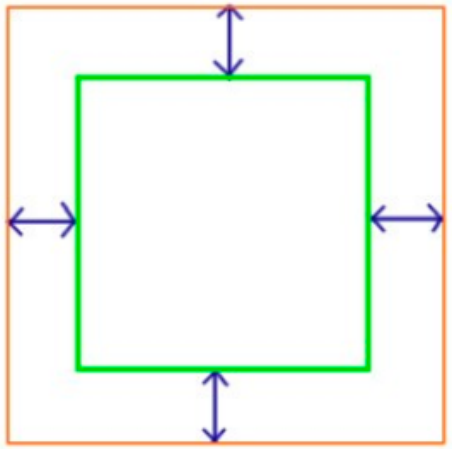
- This widget is used to center the child widget, which comes inside it

```dart
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
  }
}
```

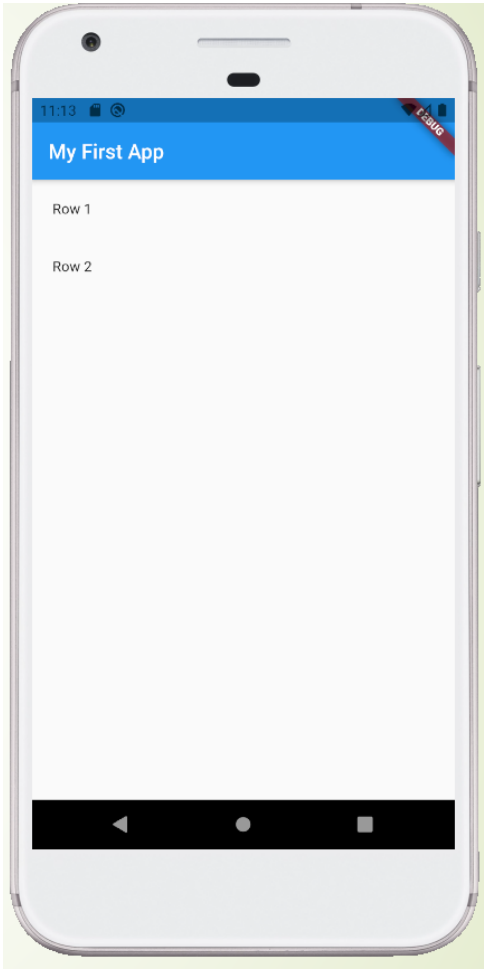


Invisible Widget - Padding

- This widget wraps other widgets to give them padding in specified directions. You can also provide padding in all directions.

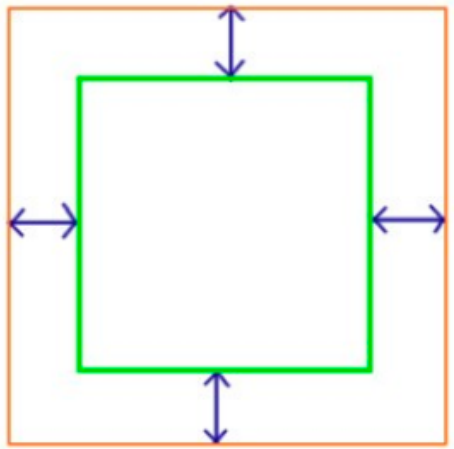


```
return MaterialApp(
    title: 'Flutter Demo',
    home: Scaffold(
      appBar: AppBar(
        title: Text('My First App'),
      ), // AppBar
      body: Column(
        children: [
          Padding(
            padding: EdgeInsets.only(top: 20, bottom: 20),
            child: Text('Row 1'),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



Padding - EdgeInsets

- EdgeInsets.**all**(double value): Creates insets where all the offsets are value
- EdgeInsets.**symmetric**({double **vertical** = 0.0, double **horizontal** = 0.0}): Creates insets with symmetrical vertical and horizontal offsets.
- 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.



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:

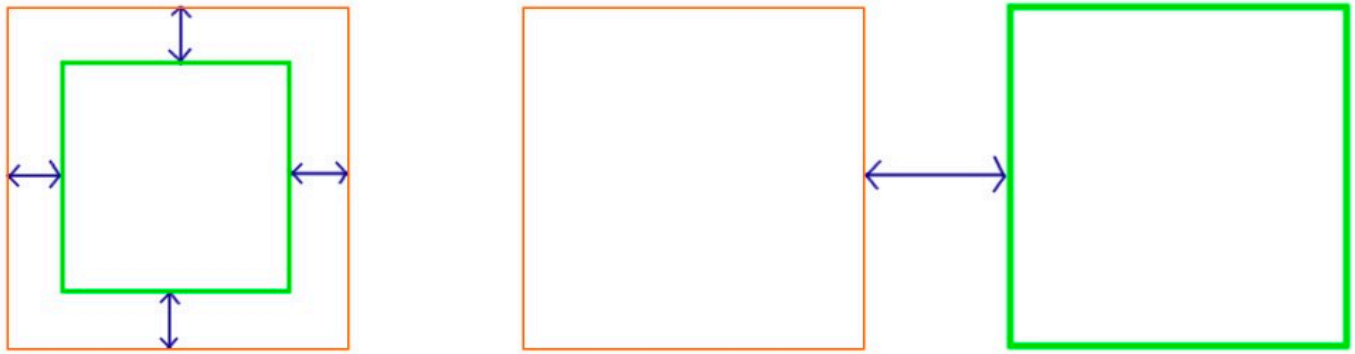
```
const EdgeInsets.symmetric(vertical: 8.0)
```

Left margin indent of 40 pixels:

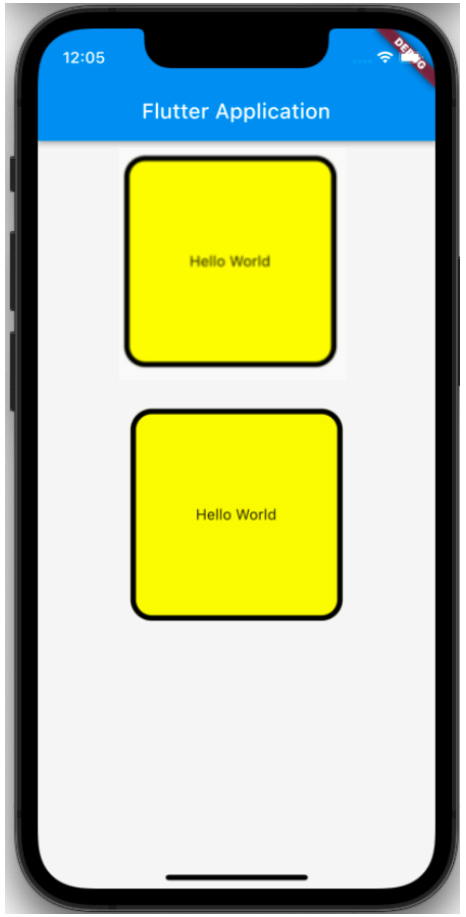
```
const EdgeInsets.only(left: 40.0)
```

Invisible Widget - Container (1)

- 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.



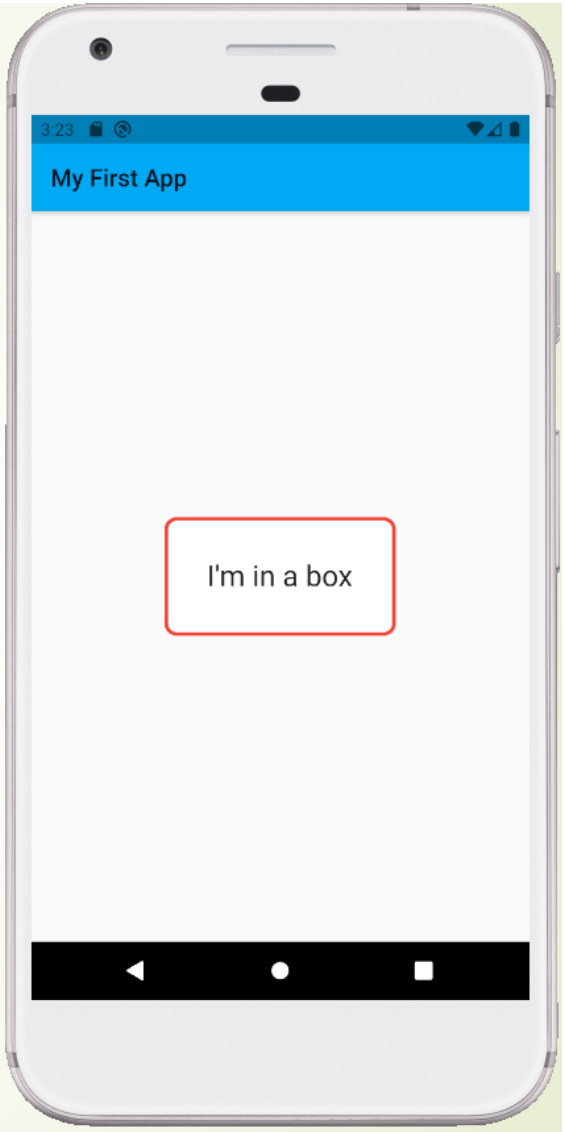
Padding **Margin**



Invisible Widget - Container (2)

- Example

```
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
```



*Keeping up those **inspiration** and the **enthusiasm** in the **learning path**.*
*Let confidence to bring it into **your career path** for getting gain the **success** as your expectation.*

Thank you

Contact

- Name: R2S Academy
- Email: daotao@r2s.edu.vn
- Phone/Zalo: 0919 365 363
- FB: https://www.facebook.com/r2s.tuyendung
- Website: www.r2s.edu.vn

Questions and Answers