Mobile Application

Flutter Layouts

Chapter-6

Part-3

Responsive

- Responsive designs adapt to different screen sizes and orientations, so whether you’re using an iPad in landscape mode or iPhone in portrait mode, the layout and content of the app will adjust to deliver the same level of user experience. Adopting responsive design ensures that apps are accessible and usable on a wide range of devices.

Flutter Widget – MediaQuery

- MediaQuery provides a higher-level view of your app’s screen size and can provide more detailed information about user layout preferences.
- The MediaQuery class in Flutter is a crucial tool for creating responsive layouts. It provides access to the media query data from the BuildContext context. This data is used to determine the physical characteristics of the device's screen size, such as screen width, screen height, device pixel ratio, and more.

What is the best screen size to design for?

- There’s no one best screen size to design for. Websites should transform responsively and fast at all screen resolutions on different browsers and platforms. Accessible. Mobile-friendly. Design for your audience, first. Design from 360×640 through 1920×1080.

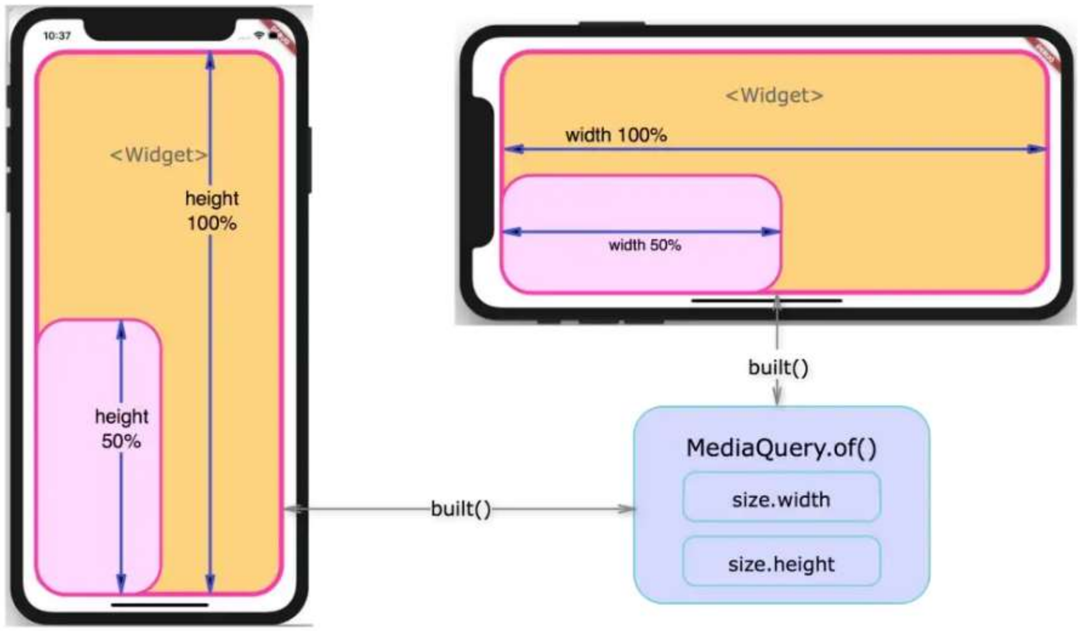
✓ **desktop** displays from **1280×720** through **1920×1080**

✓**tablet** displays from **601×962** through **1280×800**

✓**mobile** displays from **360×640** through **414×896**

Flutter Widget – MediaQuery

- Media Query can be used to get the real-time size (width/height) and orientation (portrait/landscape) of the window screen. It suggests the orientation and size of the app.

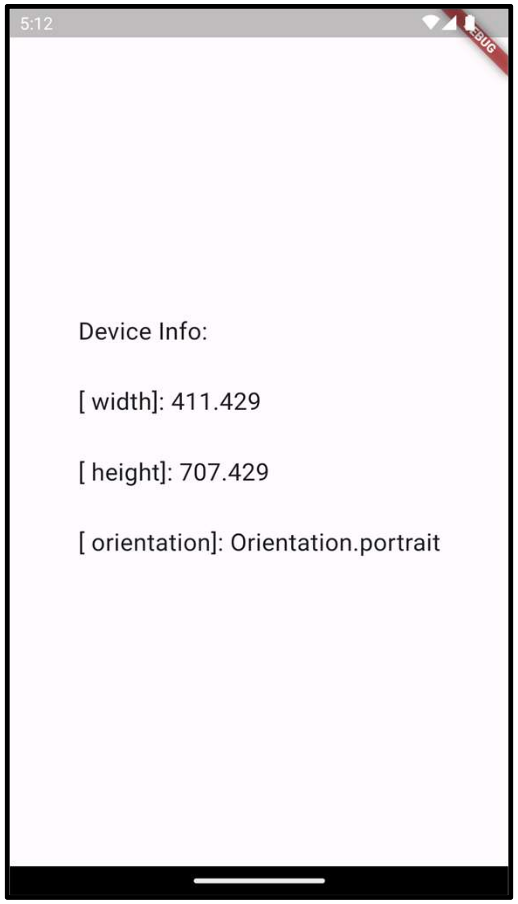


Flutter Widget – MediaQuery – Example1

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    double width = MediaQuery.of(context).size.width;

    double height = MediaQuery.of(context).size.height;

    Orientation orientation = MediaQuery.of(context).orientation;
```

Flutter Widget – MediaQuery – Example1

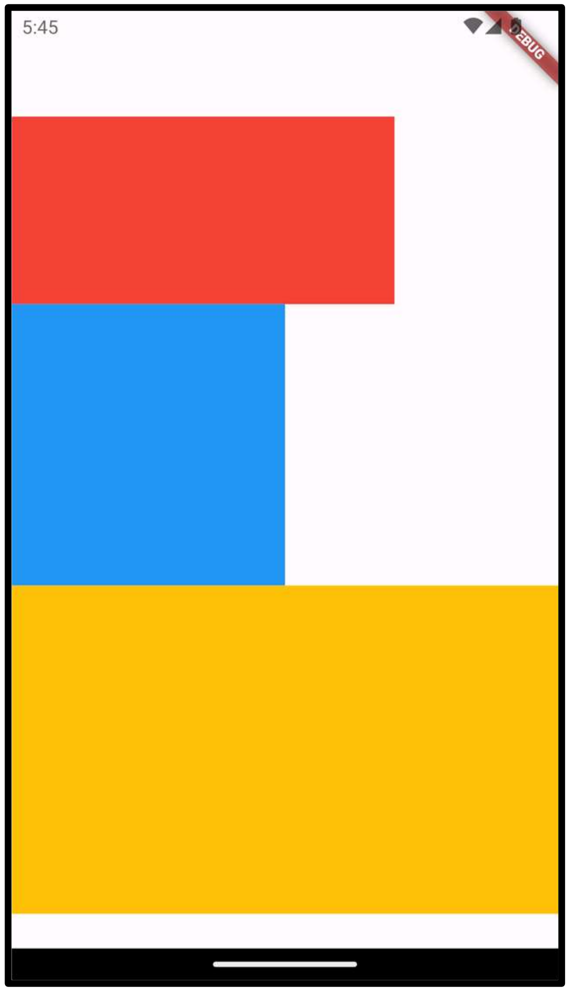
```
return MaterialApp(
home: Scaffold(
body: Center(
child: Text(
'Device Info:  \n\n' +
'[width]: ${width.toStringAsFixed(3)}\n\n' +
'[height]: ${height.toStringAsFixed(3)}\n\n' +
'[orientation]: $orientation',
style: TextStyle(fontSize: 20),
     ),
        ),   ),      );     }      }
```



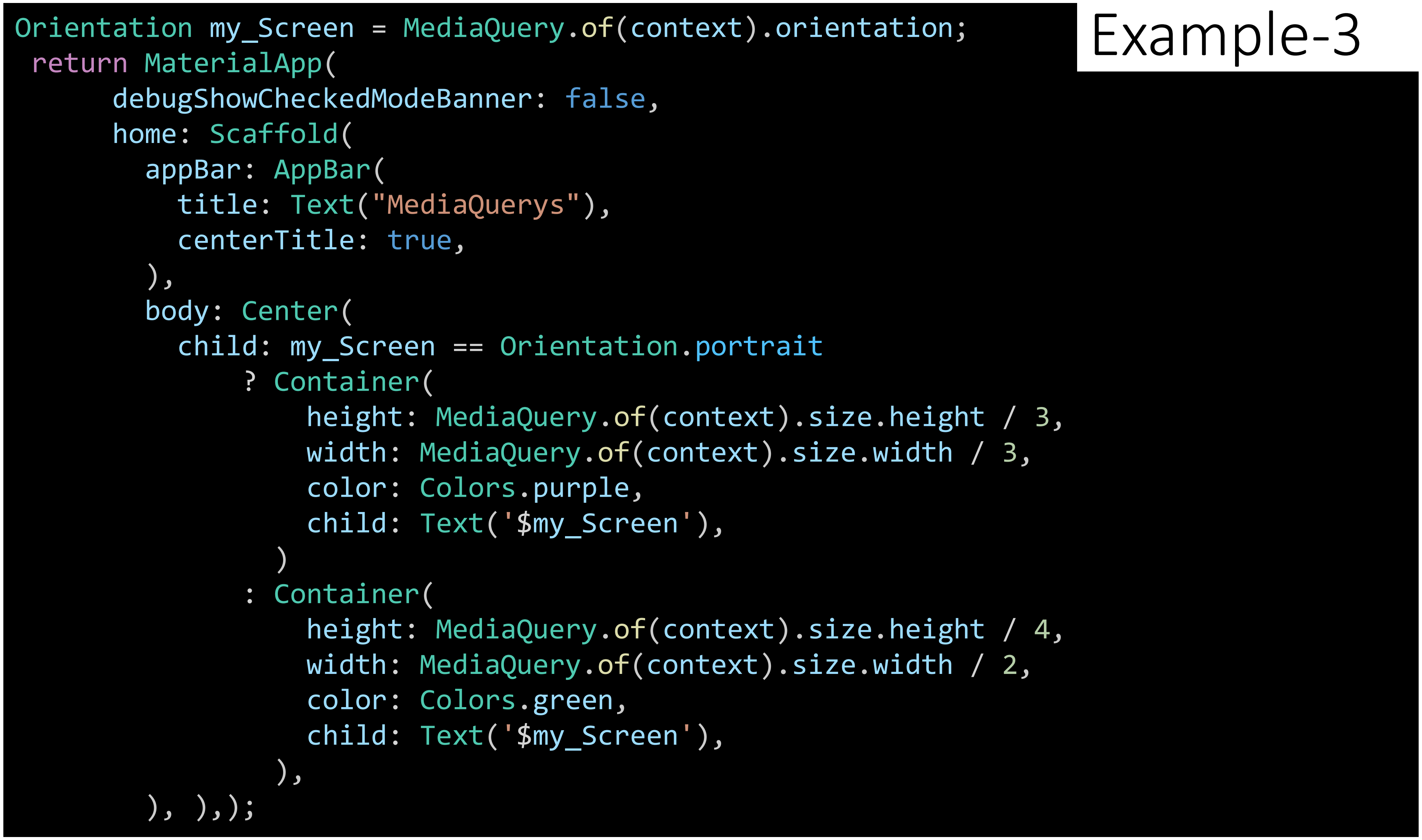
```
Device Info:

[ width]: 411.429

[ height]: 707.429

[ orientation]: Orientation.portrait
```

Flutter Widget – MediaQuery – Example2

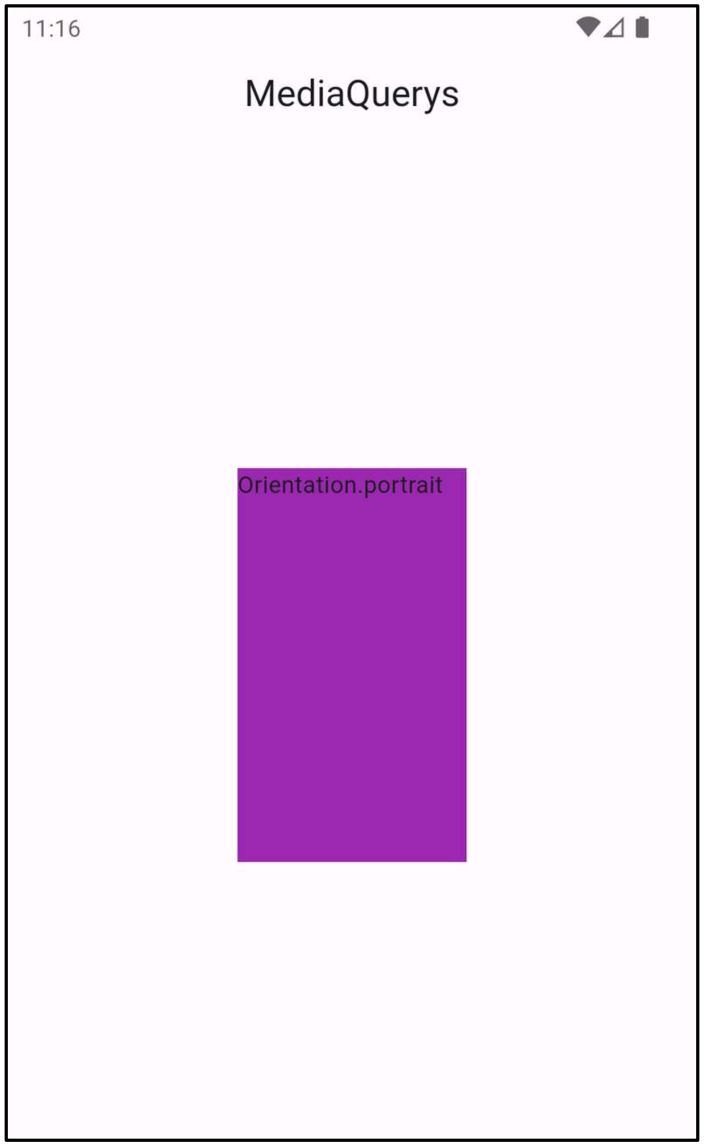
```
Scaffold(
        appBar: AppBar(),
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
                height: Screen_height * 0.2,
                width: Screen_width * 0.7,
                color: Colors.red),
            Container(
                height: Screen_height * 0.3,
                width: Screen_width / 2,
                color: Colors.blue),
            Container(
                height: Screen_height * 0.35,
                width: Screen_width,
                color: Colors.amber),
          ],
        ),
      ),
```



Example-3

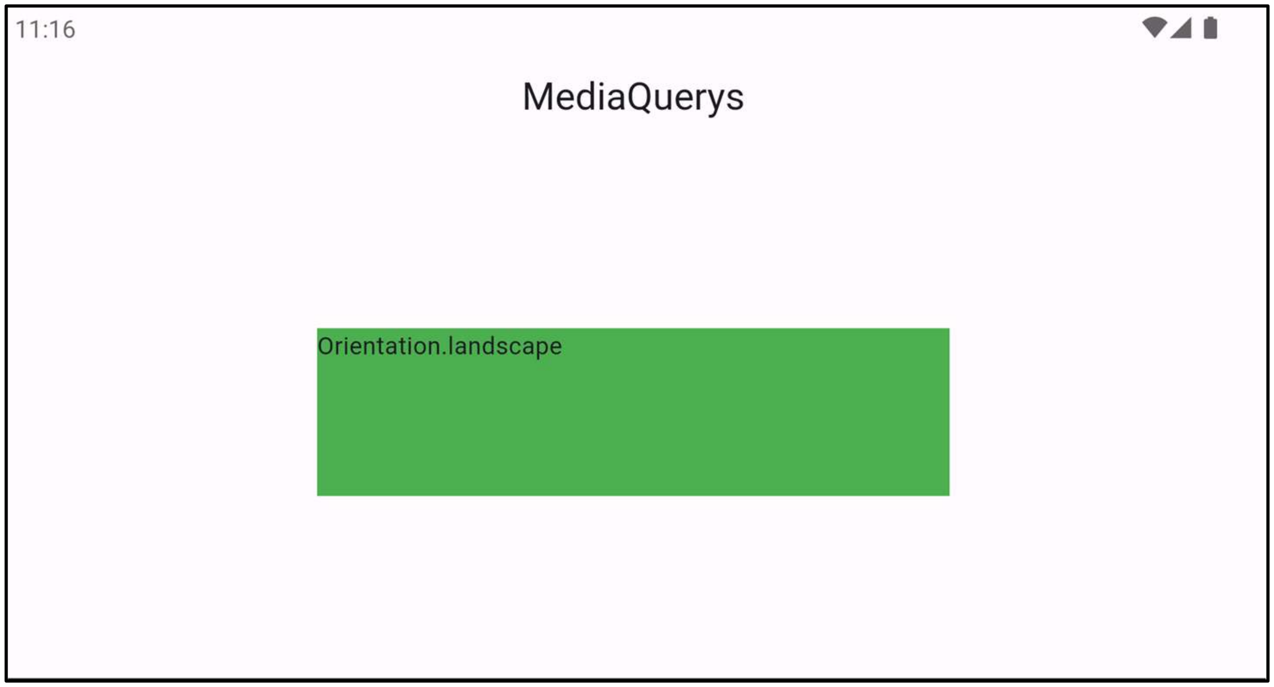


```
Orientation my_Screen = MediaQuery.of(context).orientation;
 return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("MediaQuerys"),
          centerTitle: true,
        ),
        body: Center(
          child: my_Screen == Orientation.portrait
              ? Container(
                  height: MediaQuery.of(context).size.height / 3,
                  width: MediaQuery.of(context).size.width / 3,
                  color: Colors.purple,
                  child: Text('$my_Screen'),
                )
              : Container(
                  height: MediaQuery.of(context).size.height / 4,
                  width: MediaQuery.of(context).size.width / 2,
                  color: Colors.green,
                  child: Text('$my_Screen'),
                ),
        ), ),);
```


11:16

MediaQuerys

Orientation.portrait


11:16

MediaQuerys

Orientation.landscape

Note:

- You can change the orientation by putting these code before the ***runApp()*** in your *main.dart file*

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


void main() {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp]);
  runApp(MyApp());
}
```

Layout Builder

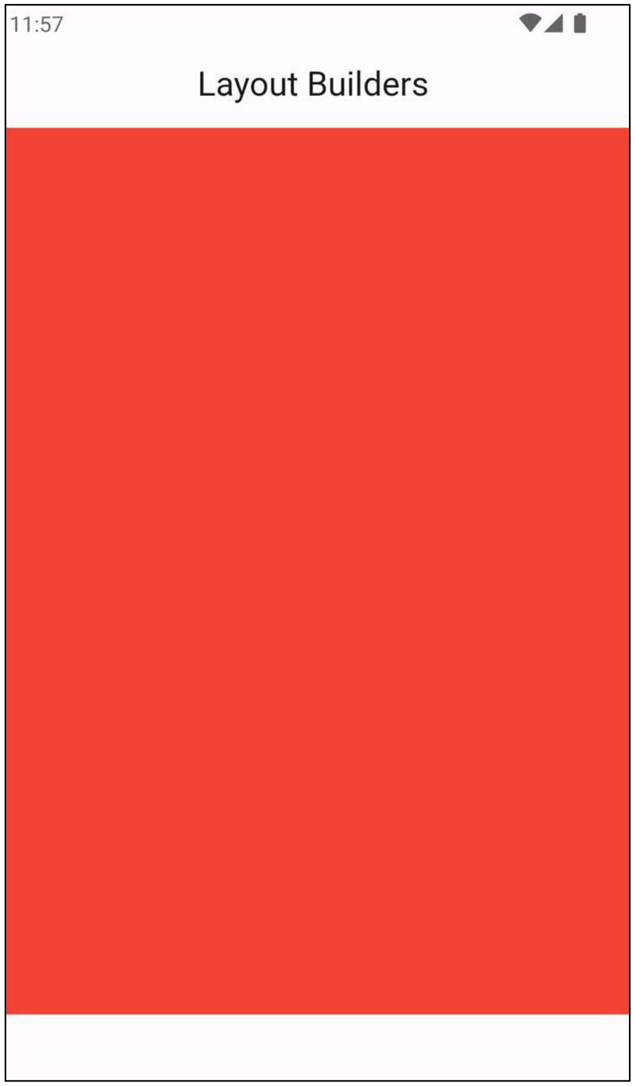
- Layout Builder is just a simplified version of Media Query. The main difference between Media Query and Layout Builder is that Media Query is based on the full context of the screen rather than just the size of a particular widget; on the other hand, Layout Builder determines the maximum width and height of a specific widget only.
- Layout Builder class provides the **Box Constraints** object that can be used for determining the **maxWidth** and **maxHeight** of the widget.

Basic Syntax of Layout Builder Widget

```
LayoutBuilder(
  builder: (BuildContext context, BoxConstraints constraints) {
// Return a widget based on the Layout Builder
return YourWidget();
 },
)
```

Example

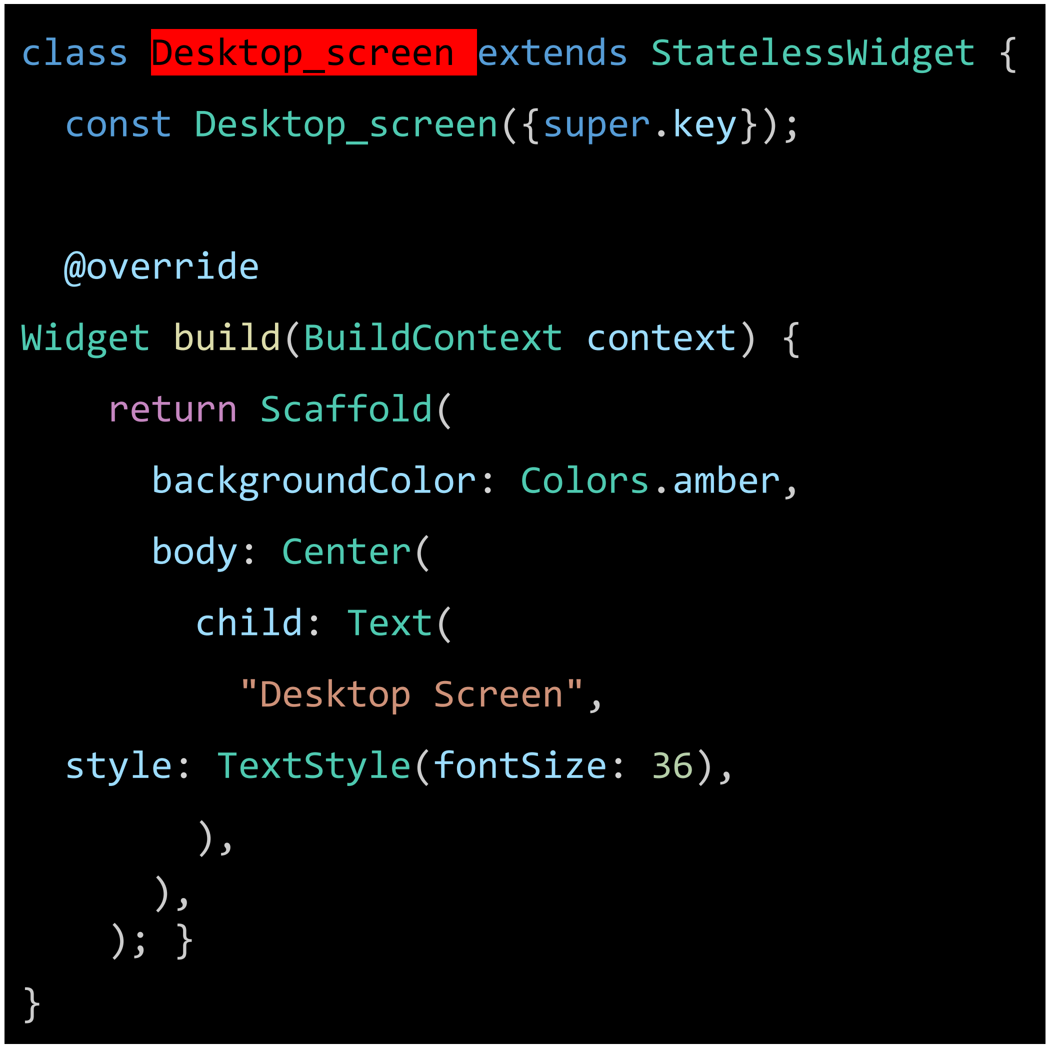
```
Scaffold(
        appBar: AppBar(
          title: Text("Layout Builders"),
          centerTitle: true,
        ),
body: LayoutBuilder(
  builder: (context, constraints) {
    return Container(
              color: Colors.red,
  height: constraints.maxHeight - 50,
  width: constraints.maxWidth,
             );
          },
        ),
      ),
```



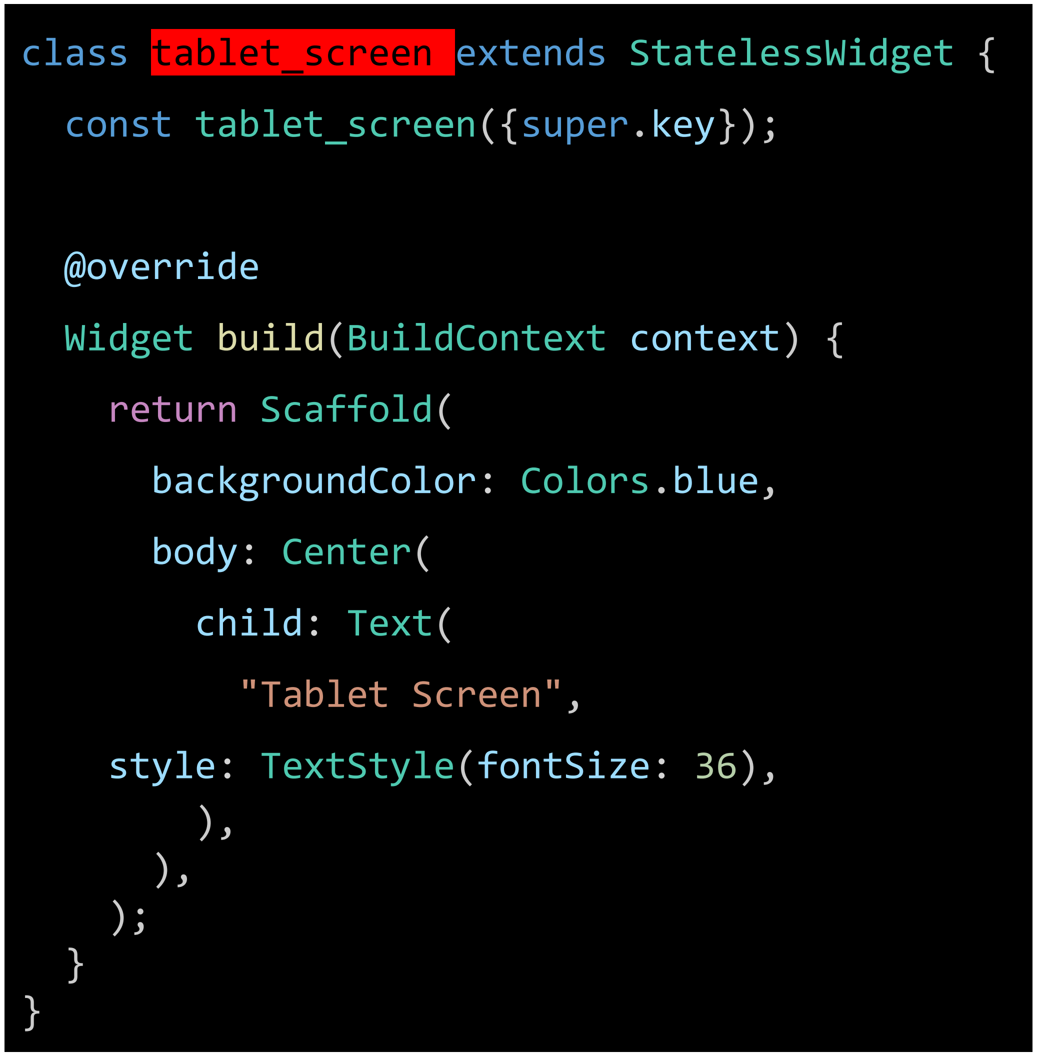
Example-2

```
 Scaffold(
        appBar: AppBar(
          title: Text("Layout Builders"),
          centerTitle: true,
        ),
  body: LayoutBuilder(builder:
(context, constraints) {
    if (constraints.maxWidth < 600) {
              return mobile_screen();
   } else if (constraints.maxWidth < 1200) {
              return tablet_screen();
   } else {
              return Desktop_screen();
            }
          },),
),
```

```
class mobile_screen extends
StatelessWidget {
  const mobile_screen({super.key});

  @override
  Widget build(BuildContext
context) {
    return Scaffold(
backgroundColor: Colors.green,
      body: Center(
        child: Text(
          "Mobile Screen",
          style:
TextStyle(fontSize: 36),
        ),
      ),
    );
  }
}
```



```dart
class Desktop_screen extends StatelessWidget {
  const Desktop_screen({super.key});

  @override
Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.amber,
      body: Center(
        child: Text(
          "Desktop Screen",
  style: TextStyle(fontSize: 36),
        ),
      ),
    ); }
}
```



```dart
class tablet_screen extends StatelessWidget {
  const tablet_screen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      body: Center(
        child: Text(
          "Tablet Screen",
    style: TextStyle(fontSize: 36),
        ),
      ),
    );
  }
}
```

Orientation Builder

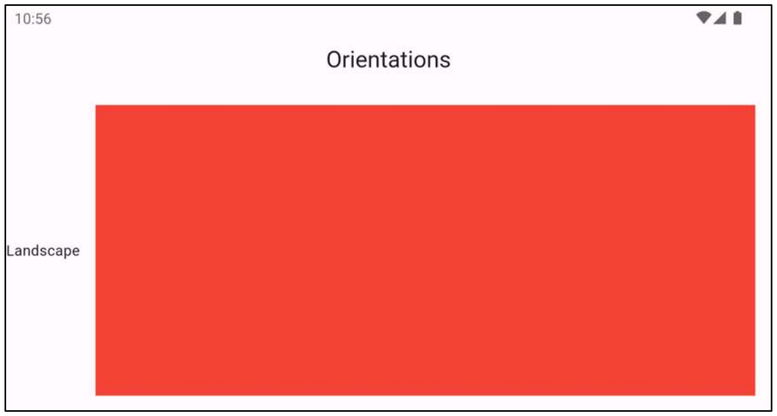
- Orientation Builder class can be used to determine a widget’s **current orientation**.
- OrientationBuilder is a Flutter widget that allows you to build a widget tree based on the orientation (**portrait** or **landscape**) of the device. It’s useful when you want to create a different UI layout or make adjustments based on the device’s orientation.

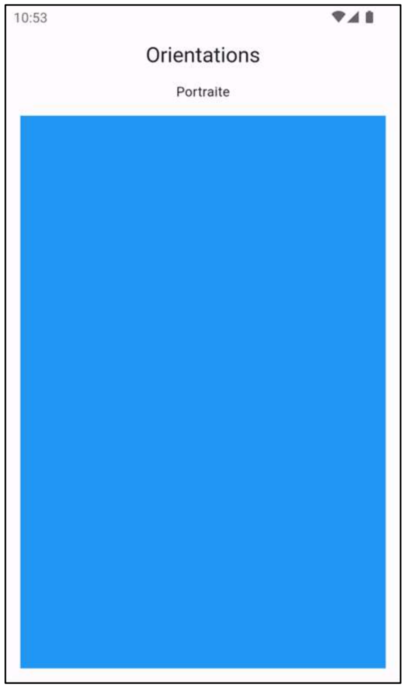
Basic Syntax of OrientationBuilder Widget

```
OrientationBuilder(
builder: (BuildContext context, Orientation orientation) {
// Return a widget tree based on the orientation
return YourWidget();
},
)
```

Example

```
MaterialApp(
debugShowCheckedModeBanner: false,
home: Scaffold(
appBar: AppBar(
title: Text("Orientations"),
centerTitle: true,
),
body: OrientationBuilder(
builder: (context, orientation) {
return orientation == Orientation.portrait
? portrait_Screen()
: landscape_Screen();
          },
),
      ),);
```





```dart
class portrait_Screen extends
StatelessWidget {
  const portrait_Screen({super.key});

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        Text("Portraite"),
        Expanded(
          child: Container(
            margin: EdgeInsets.all(15),
            color: Colors.blue,
          ),),
      ],
    );
  }
}
```

```dart
class landscape_Screen extends
StatelessWidget {
  const landscape_Screen({super.key});

  @override
  Widget build(BuildContext context) {
    return Row(
      children: [
        Text("Landscape"),
        Expanded(
          child: Container(
            margin: EdgeInsets.all(15),
            color: Colors.red,
          ),),
      ],
    );
  }
}
```