Mobile Application

Flutter Layouts

Part-1

Flutter Layouts

The main concept of the layout mechanism is the widget. We know that flutter assume everything as a widget. So the image, icon, text, and even the layout of your app are all widgets. Here, some of the things you **do not see** on your app UI, such as **rows**, **columns**, and **grids** that arrange, constrain, and align the visible widgets are also the widgets.

Flutter Layouts

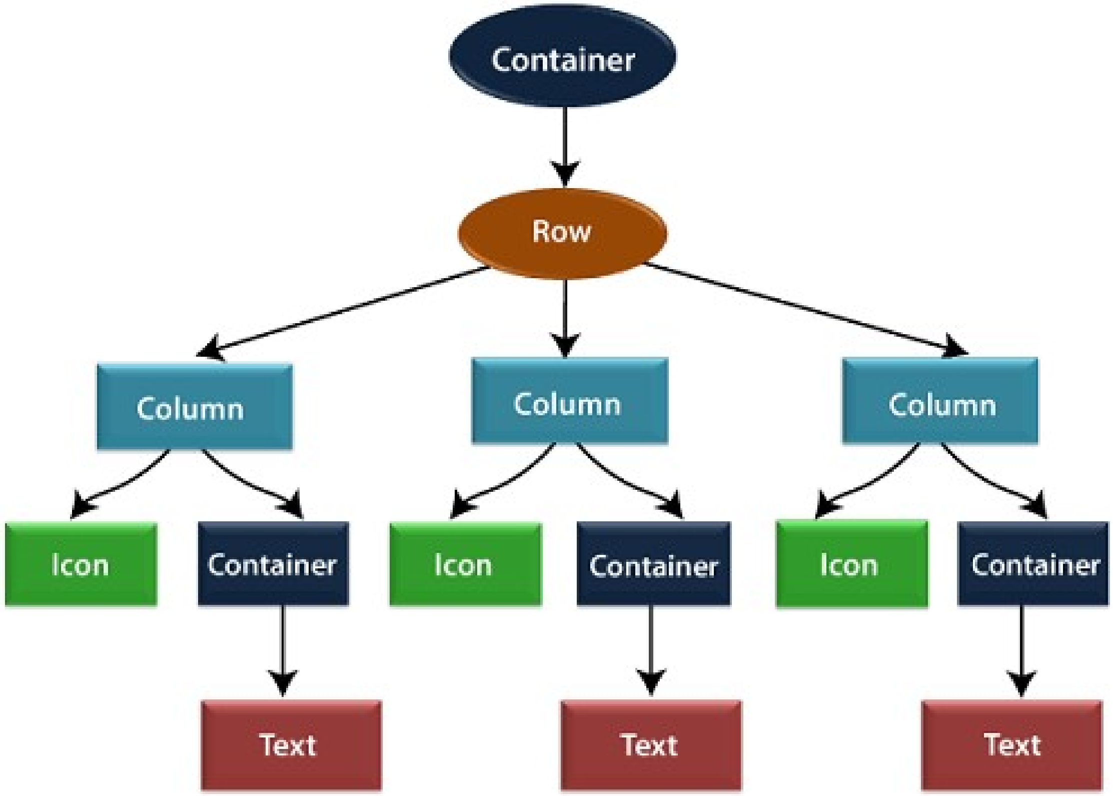
Flutter allows us to create a layout by composing multiple widgets to build more complex widgets.

For example, we can see the below image that shows three icons with a label under each one.

CALL
ROUTE
SHARE

Flutter Layouts

In the second image, we can see the visual layout of the above image. This image shows a row of three columns, and these columns contain an icon and label.



Layout a widget

The following steps show how to layout a widget:

Step 1: First, you need to select a Layout widget.

Step 2: Next, create a visible widget.

Step 3: Then, add the visible widget to the layout widget.

Step 4: Finally, add the layout widget to the page where you want to display.

Types of Layout Widgets

- We can categories the layout widget into two types:

1.Single Child Widget

2.Multiple Child Widget

Single Child Widgets

The single child layout widget is a type of widget, which can have only **one child widget** inside the parent layout widget. These widgets can also contain special layout functionality. Flutter provides us many single child widgets to make the app UI attractive. If we use these widgets appropriately, The list of different types of single child widgets are:

Single Child Widgets

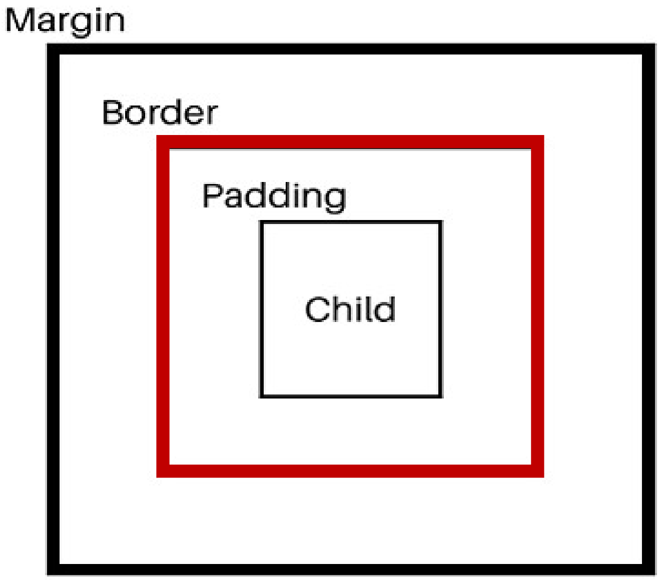
- **Center:** This widget allows you to center the child widget within itself.

```
body: Center(
   child: Text ('Center layout’) ),
```

- widthFactor = 10.0,   // **double**
- heightFactor = 6.5,   // **double**

Single Child Widgets

- **Padding:** It is a widget that is used to arrange its child widget by the given padding. It contains **EdgeInsets** and **EdgeInsets.fromLTRB** for the desired side where you want to provide padding.
- **Margin:** this property is used to **set empty space** around an object or Widget in Flutter.



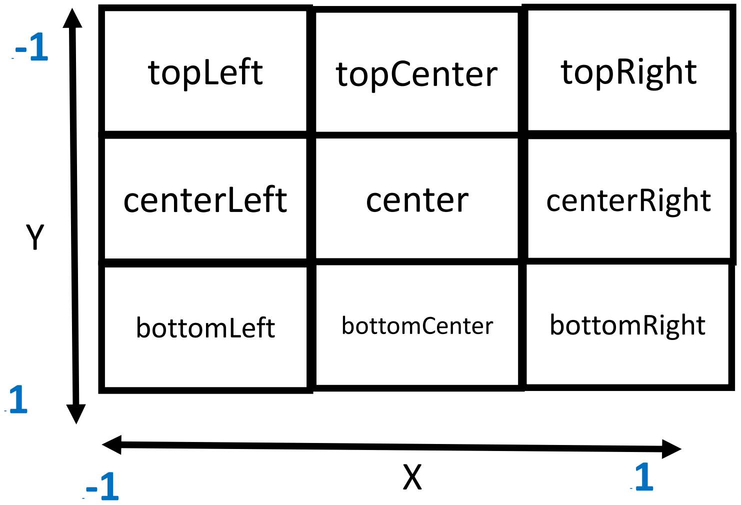
Example for margin or padding

```
//equal margin (padding) in all

margin: EdgeInsets.all(10),

 //symmetric margin (padding)

margin: EdgeInsets.symmetric(horizontal: 10, vertical: 20),

//margin (padding) on only one side

margin: EdgeInsets.only(top: 20),

//different margin (padding) from all side

margin: EdgeInsets.fromLTRB(10, 30, 10, 20),
```

Single Child Widgets

- **Align:** It is a widget, which aligns its child widget within itself and sizes it based on the child's size. It provides more control to place the child widget in the exact position where you need it.

```
Container(

  child: Align(
    alignment: Alignment(-1,-1), // (X , Y)
   //alignment: Alignment.bottomLeft,
    child: Text("Single child",
      style: TextStyle(
        fontSize: 12,
      ),),
  ),)
```

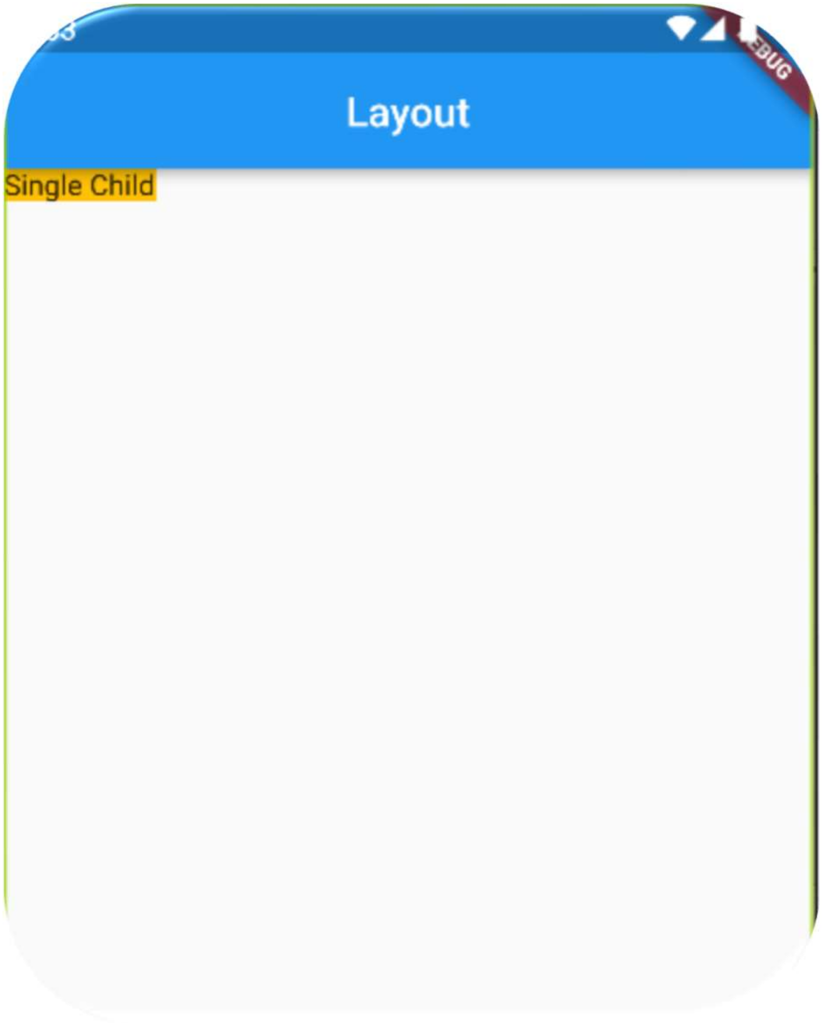


topLeft	topCenter	topRight
centerLeft	center	centerRight
bottomLeft	bottomCenter	bottomRight

Single Child Widgets

- **Container:** It is the most popular layout widget that provides customizable options for painting, positioning, and sizing of widgets.

```
body: Container(
        child: Text("Single Child"),

      ),
```



Layout

Single Child

Container

✓**child**: Container widget has a property ‘**child:**’. The child class can be any widget.

✓**color**: The color property sets the background color of the entire container.

✓**height and width:** By default, a container class takes the space that is required by the child. We can also specify the height and width of the container based on our requirements.

Container

✓**decoration**: The decoration property is used to decorate the box(e.g. give a border). This paints behind the child.

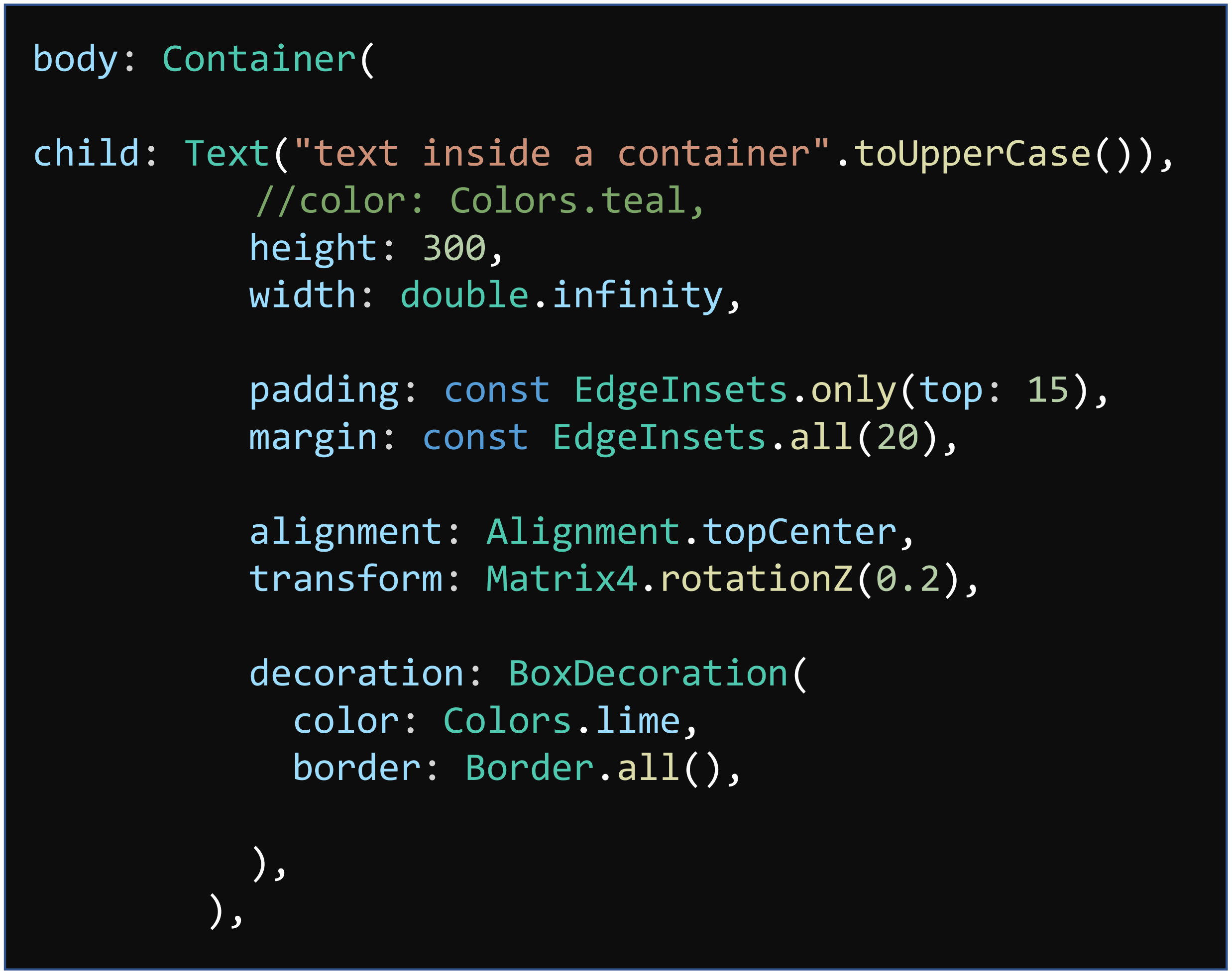
Note: change the color property to inside decoration

✓**margin and padding**

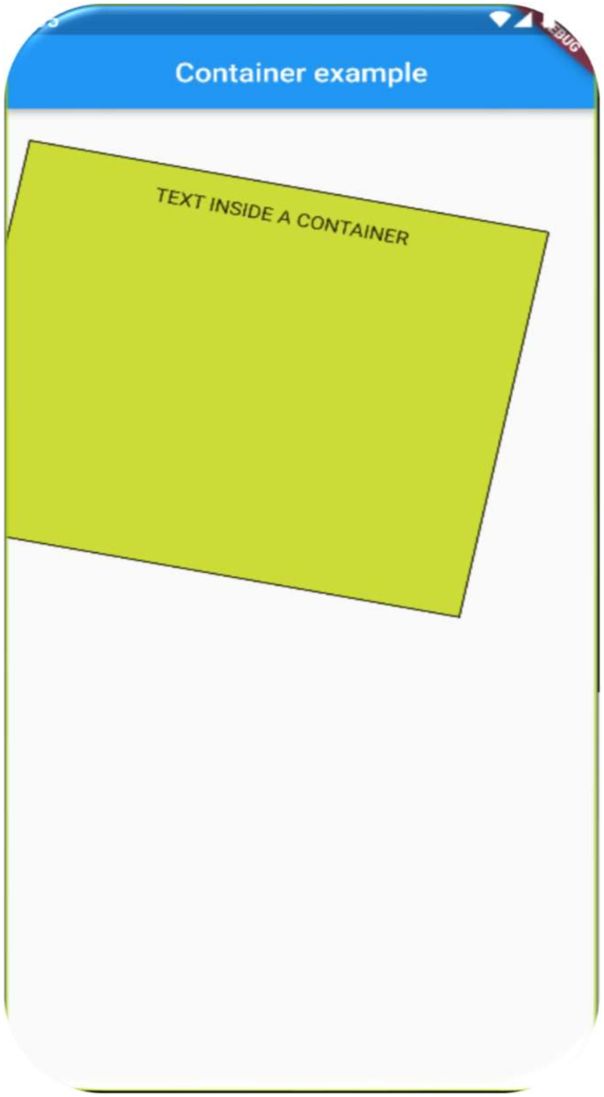
✓**Alignment**

✓**transform:** This property of the container helps us to rotate the container. We can rotate the container in any axis.

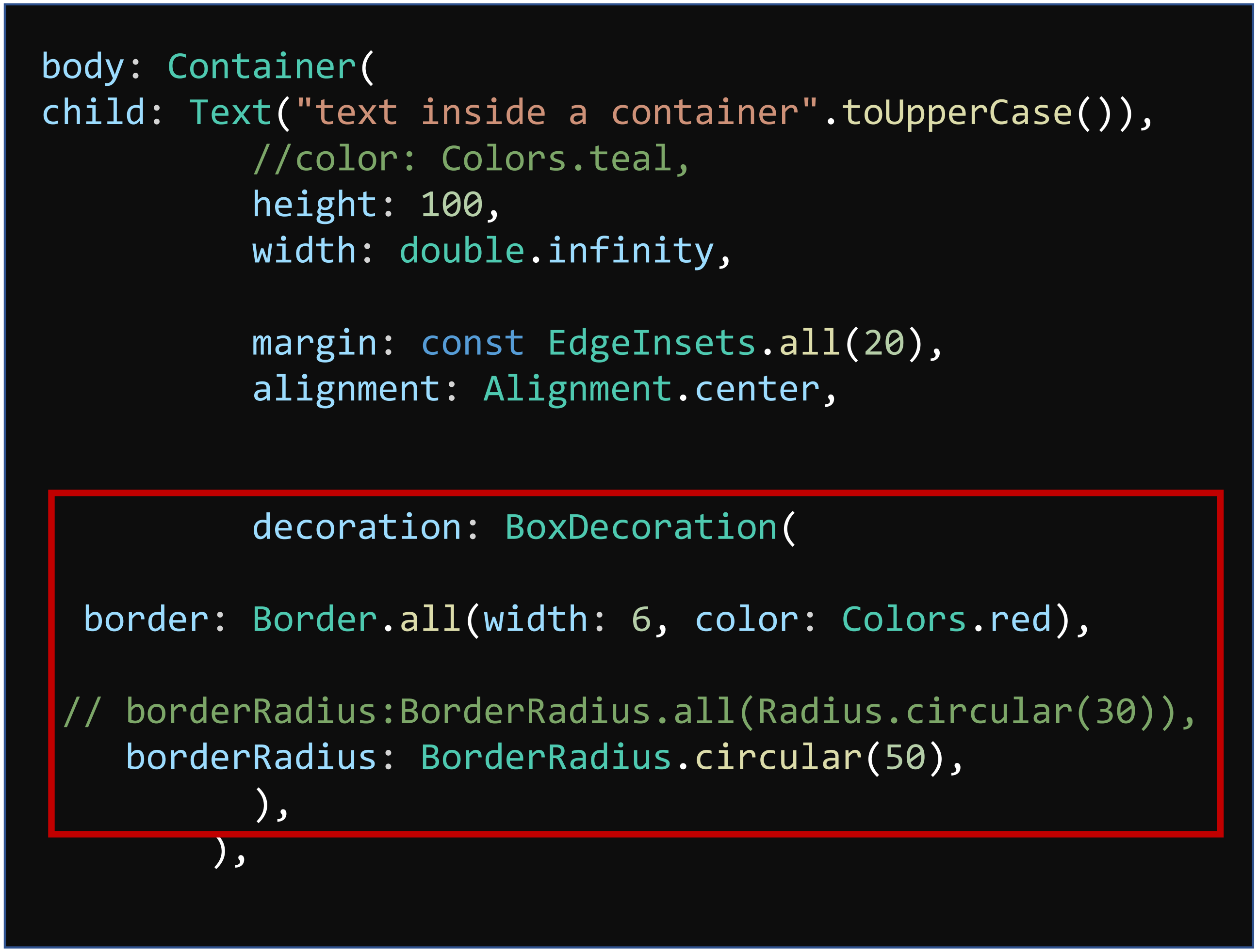
Example-1



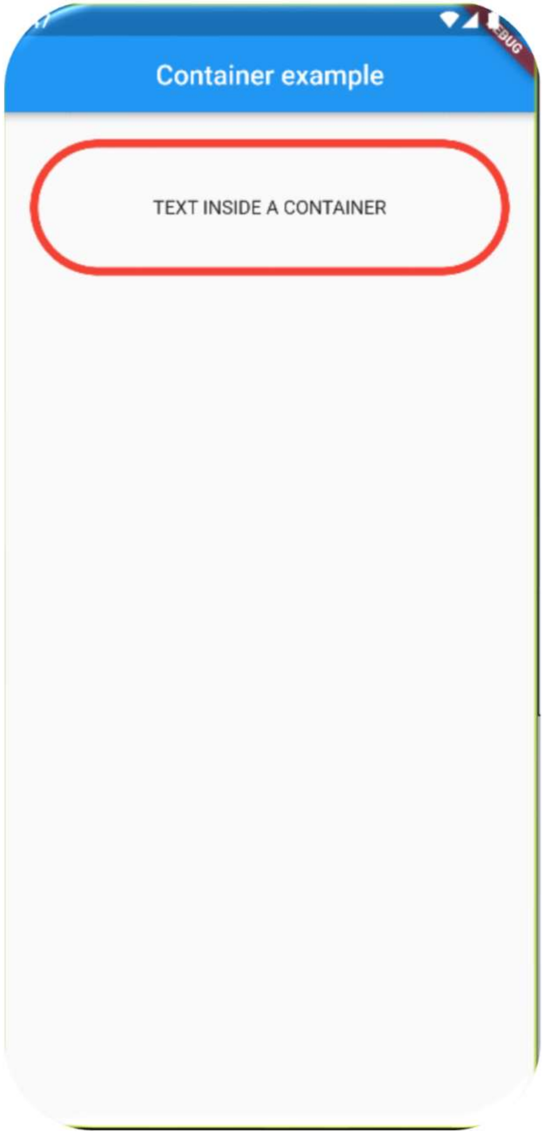
```
body: Container(

child: Text("text inside a container".toUpperCase()),
          //color: Colors.teal,
          height: 300,
          width: double.infinity,

          padding: const EdgeInsets.only(top: 15),
          margin: const EdgeInsets.all(20),

          alignment: Alignment.topCenter,
          transform: Matrix4.rotationZ(0.2),

          decoration: BoxDecoration(
            color: Colors.lime,
            border: Border.all(),

          ),
        ),
```



Example-2

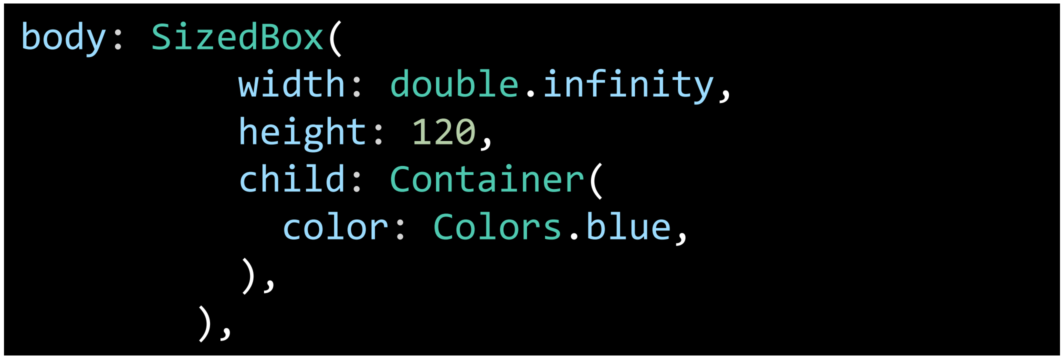


```
body: Container(
child: Text("text inside a container".toUpperCase()),
          //color: Colors.teal,
          height: 100,
          width: double.infinity,

          margin: const EdgeInsets.all(20),
          alignment: Alignment.center,


          decoration: BoxDecoration(

  border: Border.all(width: 6, color: Colors.red),

 // borderRadius:BorderRadius.all(Radius.circular(30)),
    borderRadius: BorderRadius.circular(50),
          ),
        ),
```



SizedBox

- It is a simple box with a specified size. It can be used to set size constraints to the child widget, put an empty *SizedBox* between the two widgets to get some space in between, or something else. It is somewhat similar to a **Container widget** with fewer properties.



```
body: SizedBox(
          width: double.infinity,
          height: 120,
          child: Container(
            color: Colors.blue,
          ),
        ),
```

Multiple Child widgets

- The multiple child widgets are a type of widget, which contains **more than one child widget**, and the layout of these widgets are **unique**. For example, Row widget laying out of its child widget in a horizontal direction, and Column widget laying out of its child widget in a vertical direction. If we combine the **Row** and **Column** widget, then it can build any level of the complex widget.

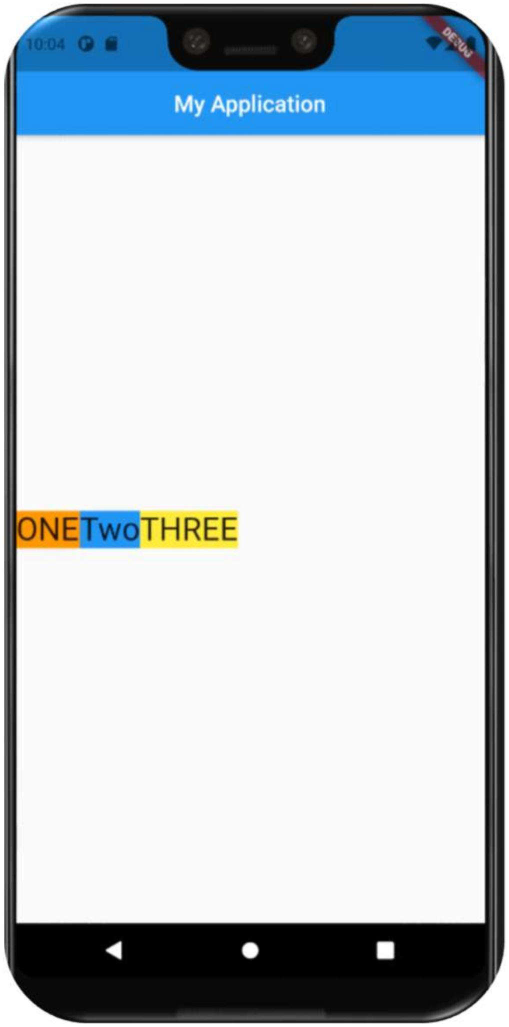
Multiple Child widgets

Row: It allows to arrange its child widgets in a horizontal direction.

Column: It allows to arrange its child widgets in a vertical direction.

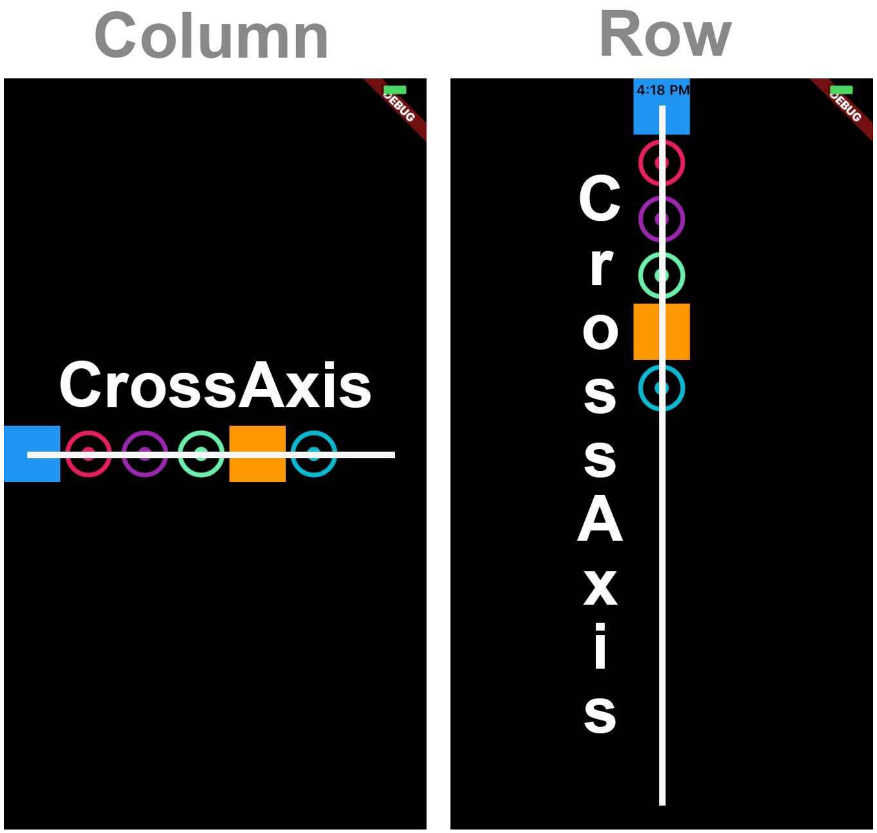
If we wished to display three text widgets within a row we can create a Row widget like below:

```
body: Center(
    child:Row(
      children: [
        Container(
          color: Colors.orange,
          child: Text('ONE',style: TextStyle(fontSize: 28),) ),
        Container(
          color: Colors.blue,
          child:Text('Two',style: TextStyle(fontSize: 28),) ),
        Container(
          color: Colors.yellow,
          child:Text('THREE',style: TextStyle(fontSize: 28),)
        ),
      ],
    )
),
```



Multiple Child widgets

- **Column** and **Row** have the same properties. So in the examples below we are working in the same time with both widgets.
- What is the **CrossAxis** in Row and Column?



CrossAxisAlignment Propery

- We can use the ***crossAxisAlignment*** property to align our child widget in the desired direction, for example, crossAxisAlignment.**start** would place the children with their start edge aligned with the start side of the cross axis.

```
crossAxisAlignment: CrossAxisAlignment.start
```

Column and **Row**

- TextDirection Propery

Determines the order to lay children out horizontally and how to interpret **start** and **end** in the horizontal direction.

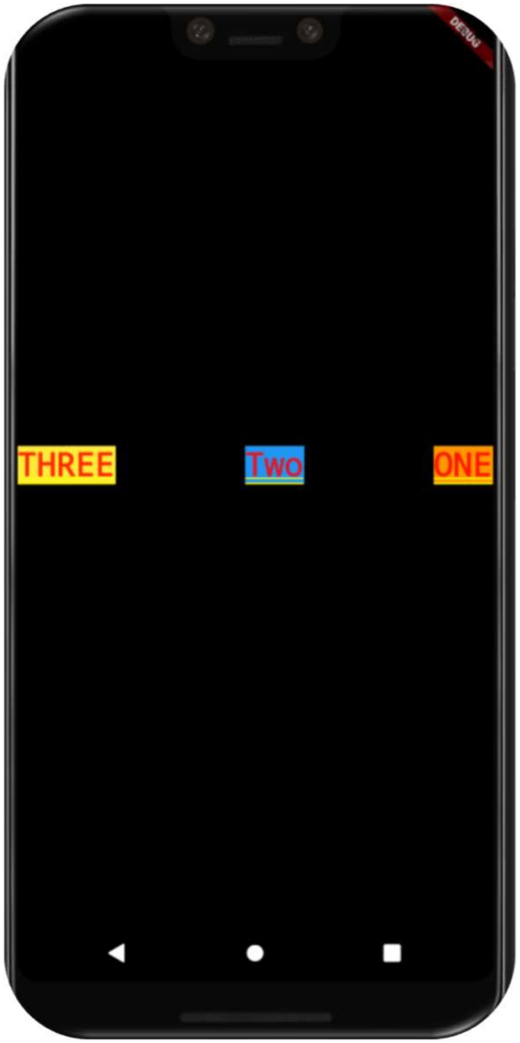
```
textDirection: TextDirection.rtl
```

Column and **Row**

- **MainAxisAlignment Property**

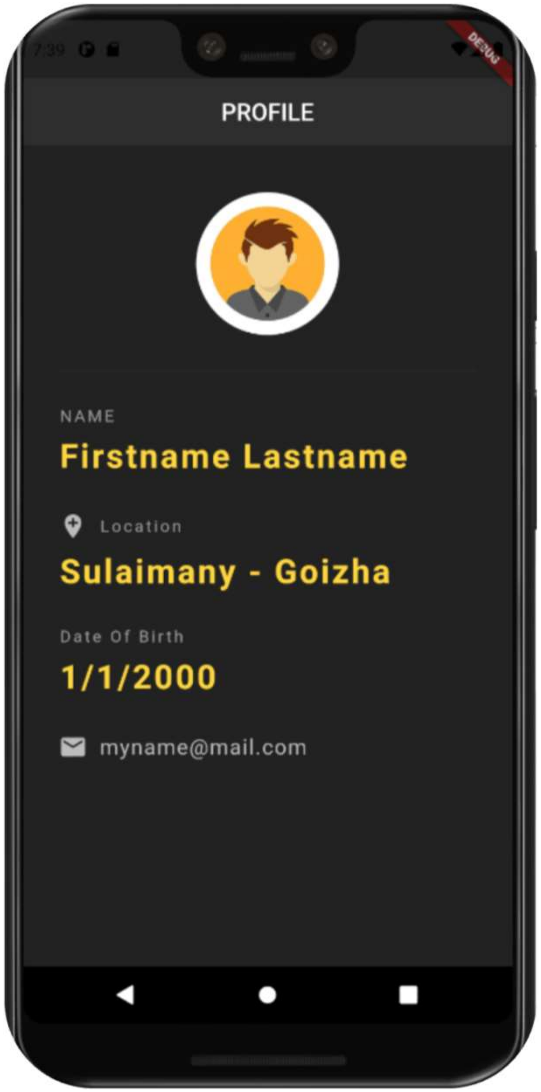
The positioning of the child widgets on the main axis

```
mainAxisAlignment: MainAxisAlignment. spaceBetween,
```



Example

Lets try the following codes to design this layout



PROFILE

NAME

Firstname Lastname

Location

Sulaimany - Goizha

Date Of Birth

1/1/2000

myname@mail.com

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

void main() => runApp(MaterialApp(
 home: RowCol(),
));

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


  return Scaffold(
   backgroundColor: Colors.grey[900],
   appBar: AppBar(
    title: Text('PROFILE'),
    centerTitle: true,
    backgroundColor: Colors.grey[850],

   ),
```

1

```
body: Padding(
 padding: const EdgeInsets.fromLTRB(30.0, 40.0, 30.0, 0),
 child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
   Center(
    child: CircleAvatar(
     radius: 60.0, // size of image or avatar
     backgroundImage: AssetImage('assets/pic7.jpg'),
    ), ),
      Divider(
    color: Colors.grey[800],
    height: 60.0,
   ),
   Text( 'NAME',
    style: TextStyle(
     color: Colors.grey,
     letterSpacing: 2.0,
    ),),
   SizedBox(height: 10.0),
```

2

```
Text(
  'Firstname Lastname',
  style: TextStyle(
    color: Colors.amberAccent[200],
    fontWeight: FontWeight.bold,
    fontSize: 28.0,
    letterSpacing: 2.0,
  ), ),
SizedBox(height: 30.0),
Row(
  children: [
    Icon(Icons.add_location,
      color: Colors.grey[400] ,),
    SizedBox(width: 10,),
    Text(
      'Location',
      style: TextStyle(
        color: Colors.grey,
        letterSpacing: 2.0,
      ),), ],
),
```

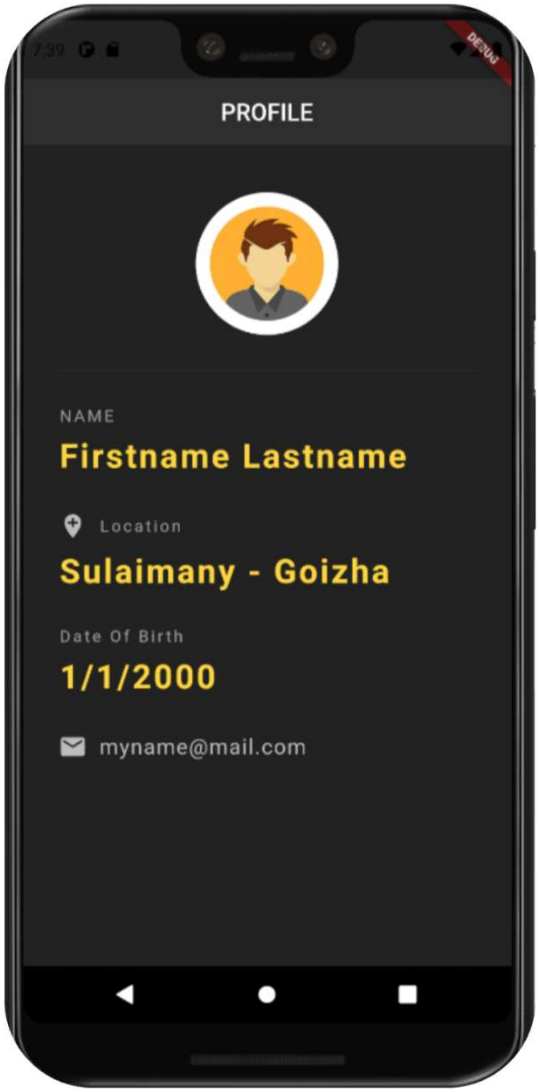
3

```
SizedBox(height: 10.0),
Text('Sulaimany - Goizha',
  style: TextStyle(
    color: Colors.amberAccent[200],
    fontWeight: FontWeight.bold,
    fontSize: 28.0,
    letterSpacing: 2.0,
  ),),
SizedBox(height: 30.0),
Text('Date Of Birth',
  style: TextStyle(
    color: Colors.grey,
    letterSpacing: 2.0, ),),
SizedBox(height: 10.0),
Text('1/1/2000',
  style: TextStyle(
    color: Colors.amberAccent[200],
    fontWeight: FontWeight.bold,
    fontSize: 28.0,
    letterSpacing: 2.0,
  ),),
```

4

```
          SizedBox(height: 30.0),
          Row(
            children: <Widget>[
              Icon(
                Icons.email,
                color: Colors.grey[400],
              ),
              SizedBox(width: 10.0),
              Text(
                'myname@mail.com',
                style: TextStyle(
                  color: Colors.grey[400],
                  fontSize: 18.0,
                  letterSpacing: 1.0,
                ),
              )
            ],
          ),
        ],
      ),
    ),
  );
}
```

5



PROFILE

NAME

Firstname Lastname

Location

Sulaimany - Goizha

Date Of Birth

1/1/2000

myname@mail.com

?