

SDP

LAB-7

Name : Makwana Nisha Shaileshbhai

ID : 20CEUBG012

Roll no. : CE065

Code test:1 – Images

→ TO CREATE STATELESS WIDGET – SHORTCUT

→ Just type stless and press 'tab' key....it will create automatically a class of stateless widget...just change the class name according to your requirements.

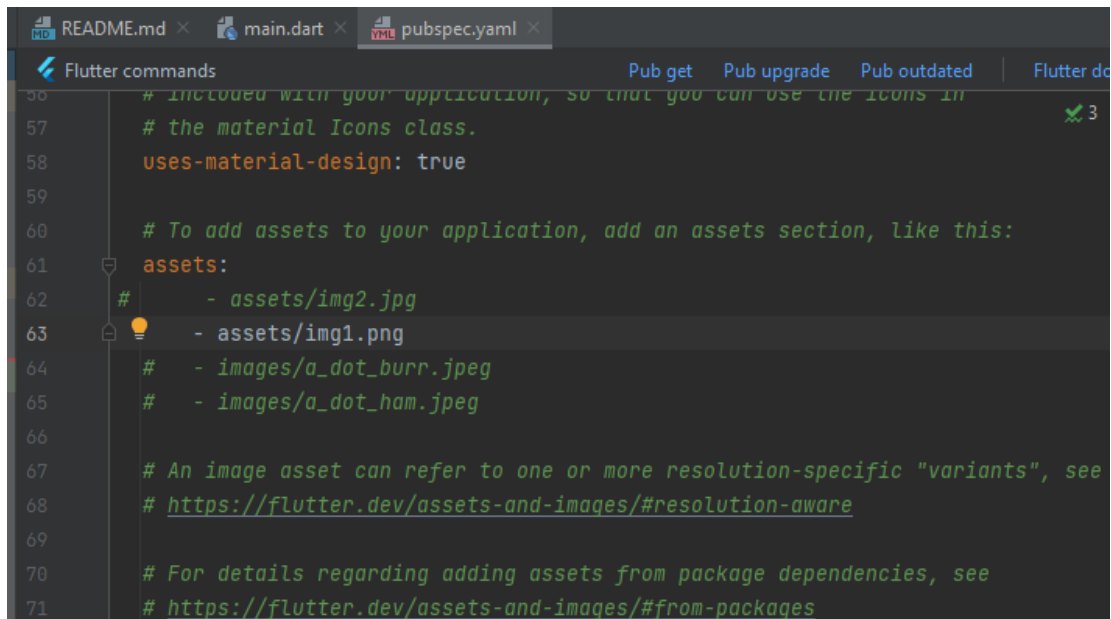
→ ADDING IMAGES IN PROJECT....

→ Similar to add font we can add images. Follow the same steps...

→ create a directory named 'assets'.

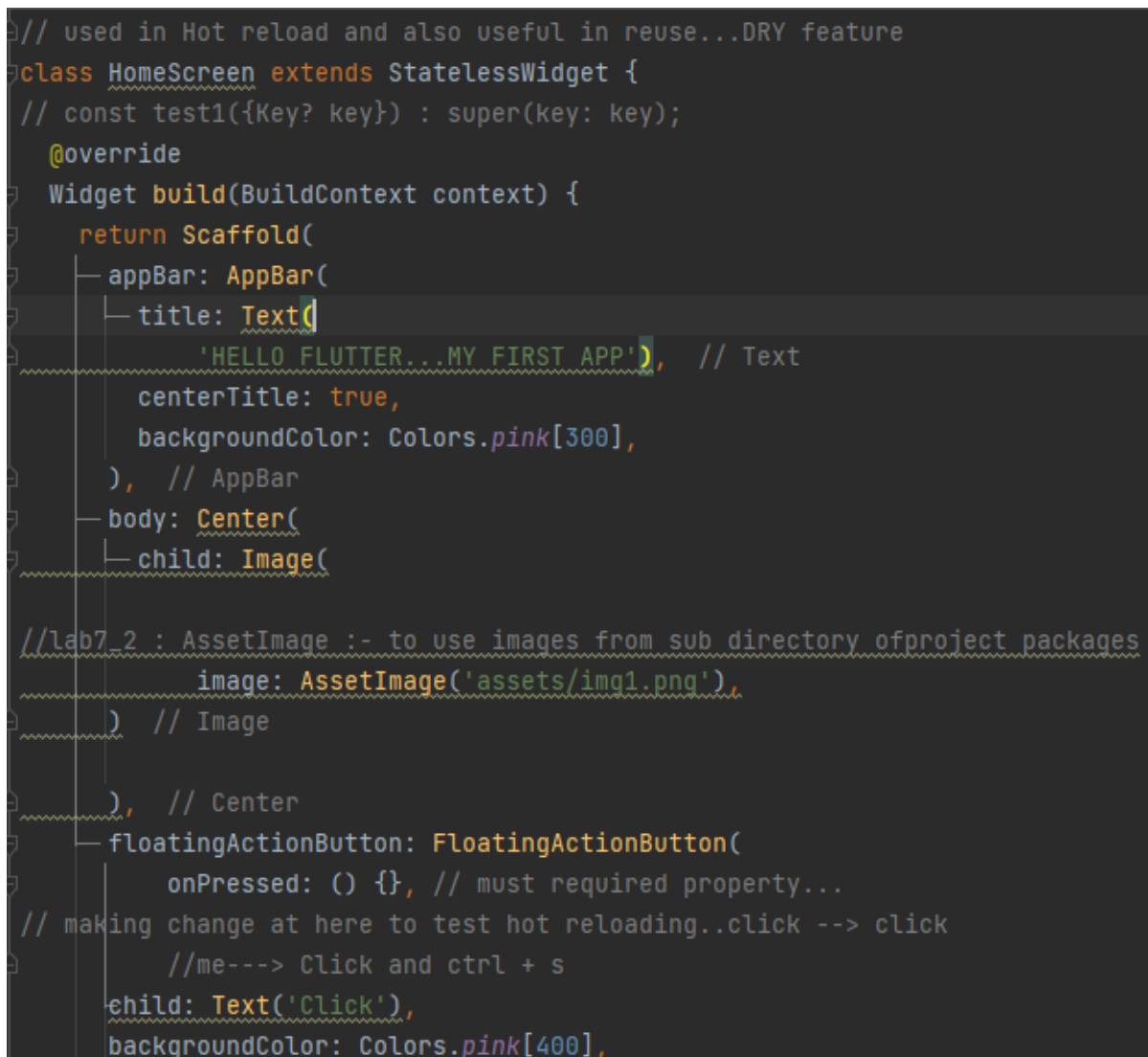
→ Put images in that directory which you want to display.

→ Do changes in 'pubspec.yaml' file like following.



```
56 # included with your application, so that you can use the icons in
57 # the material Icons class.
58 uses-material-design: true
59
60 # To add assets to your application, add an assets section, like this:
61 assets:
62 #   - assets/img2.jpg
63 #   - assets/img1.png
64 #   - images/a_dot_burr.jpeg
65 #   - images/a_dot_ham.jpeg
66
67 # An image asset can refer to one or more resolution-specific "variants", see
68 # https://flutter.dev/assets-and-images/#resolution-aware
69
70 # For details regarding adding assets from package dependencies, see
71 # https://flutter.dev/assets-and-images/#from-packages
```

Code:



```
// used in Hot reload and also useful in reuse...DRY feature
class HomeScreen extends StatelessWidget {
  // const test1({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'HELLO FLUTTER...MY FIRST APP'), // Text
        centerTitle: true,
        backgroundColor: Colors.pink[300],
      ), // AppBar
      body: Center(
        child: Image(
          //lab7_2 : AssetImage :- to use images from sub directory of project packages
          image: AssetImage('assets/img1.png'),
        ), // Image
      ), // Center
      floatingActionButton: FloatingActionButton(
        onPressed: () {}, // must required property...
        // making change at here to test hot reloading..click --> click
        //me---> Click and ctrl + s
        child: Text('Click'),
        backgroundColor: Colors.pink[400],
      ),
    );
  }
}
```

Output:

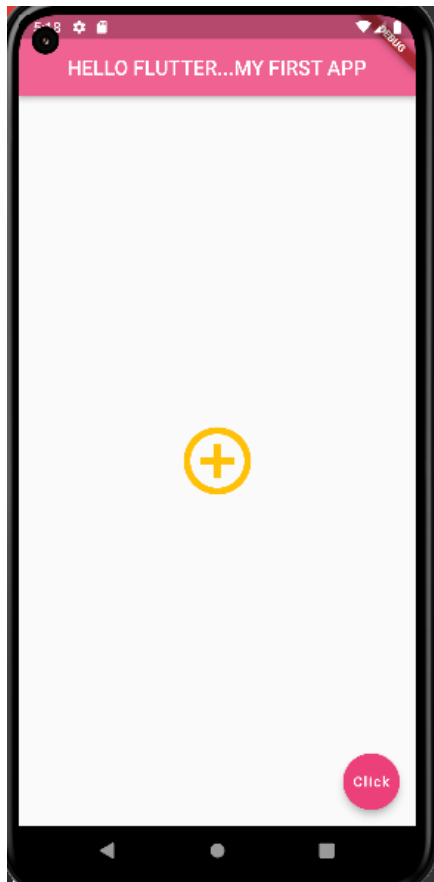


Code test : 2 – Icon

For using this class you must ensure that you have set `used-material-design: true` in the `pubspec.yml` file of your object.

```
body: Center(  
  child: Icon(  
    Icons.add_circle_outline,  
    color: Colors.amber,  
    size: 80.0,  
  ), // Icon
```

Output:



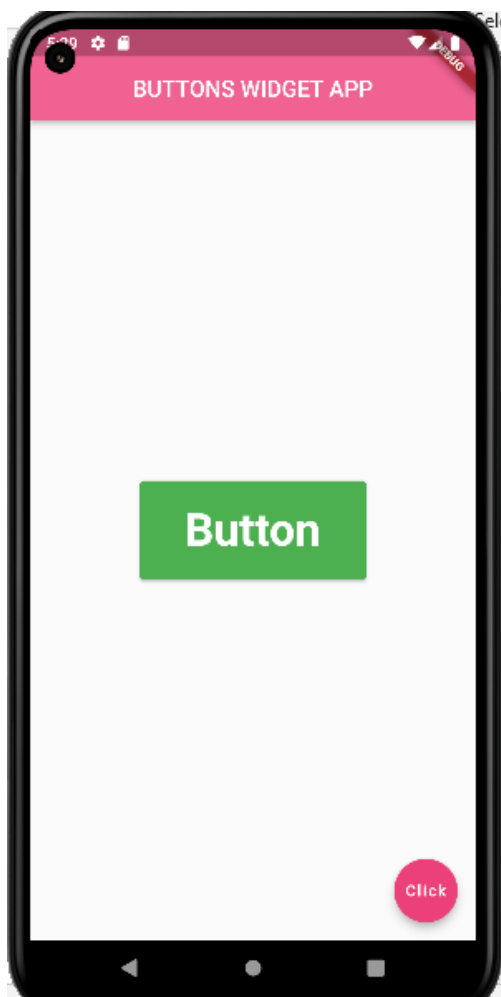
Code test 3 : Buttons

→Elevated Button:

- Use elevated buttons to add dimension to otherwise mostly flat layouts, e.g. in long busy lists of content, or in wide spaces. Avoid using elevated buttons on already-elevated content such as dialogs or cards.

```
child: ElevatedButton(  
  child: Text('Button'),  
  onPressed: () {},  
  style: ElevatedButton.styleFrom(  
    primary: Colors.green[500],  
    padding: EdgeInsets.symmetric(horizontal: 40, vertical: 20),  
    textStyle: TextStyle(  
      fontSize: 40,  
      fontWeight: FontWeight.bold)), // TextStyle  
), // ElevatedButton
```

Output:

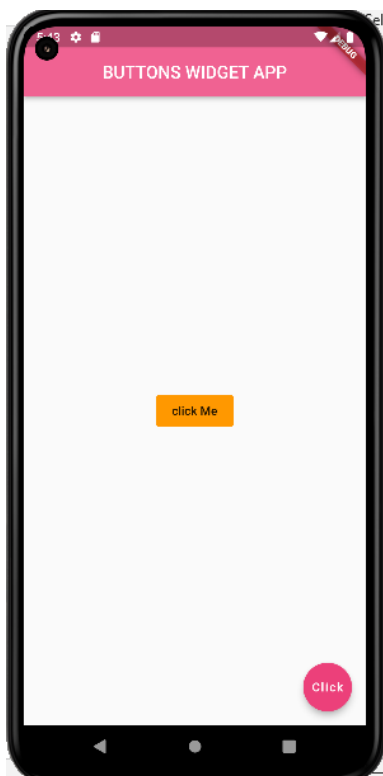


→ Flat Button:

- In flutter, we will use **FlatButton widget** to display a simple button. FlatButton is just a text button, because it has no style and border. It has no elevation, button color and text color like raised button.
- 'FlatButton' is deprecated and shouldn't be used. Use TextButton instead.

```
), */  
child: FlatButton(  
  onPressed: () {  
    print('print on console');  
  },  
  child: Text('click Me'),  
  color: Colors.orange,  
) // FlatButton  
/*
```

Output:

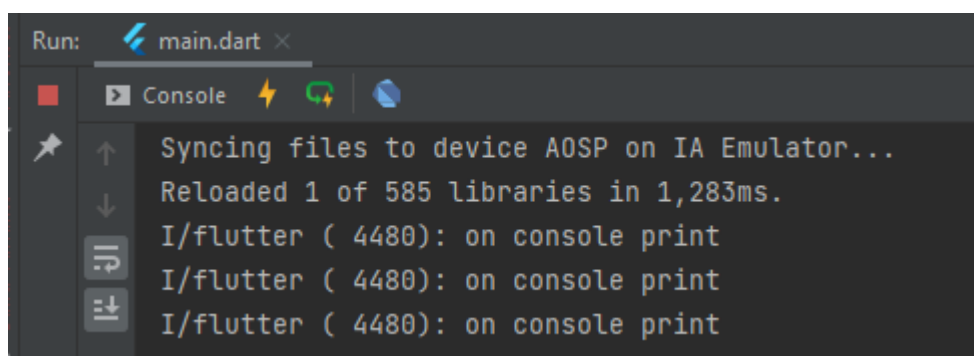


→Icon Button:

- An icon button is a picture printed on a Material widget that reacts to touches by filling with color (ink).
- Generally we will use IconButton in the AppBar.actions field, but we can use them in many other places as well.
- If the **onPressed** callback is null, then the button will be disabled and will not react to touch.

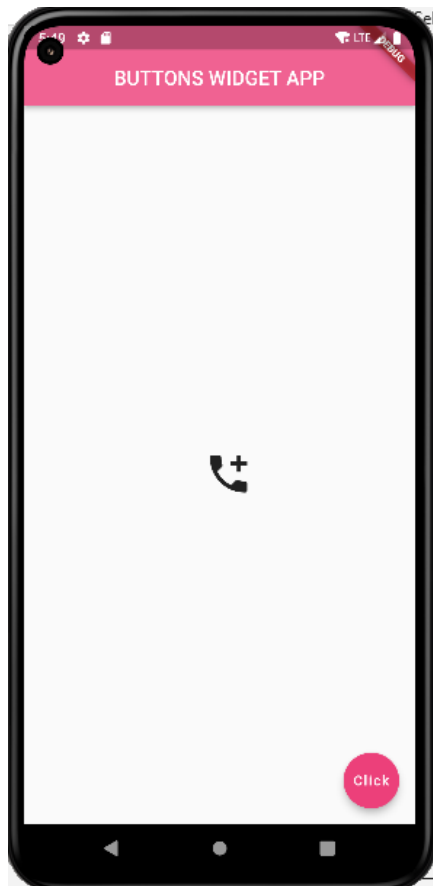
```
), */
child: IconButton(
  icon: Icon(
    Icons.add_call,
    size: 50.0,
  ), // Icon
  tooltip: 'send mail me',
  onPressed: () {
    print('on console print');
  },
), // IconButton
```

Output:



The screenshot shows the Flutter IDE's console window. At the top, it says 'Run: main.dart'. Below that, the console output is displayed. It starts with 'Syncing files to device AOSP on IA Emulator...' and 'Reloaded 1 of 585 libraries in 1,283ms.' followed by three lines of output: 'I/flutter (4480): on console print'.

```
Run: main.dart x
Console
Syncing files to device AOSP on IA Emulator...
Reloaded 1 of 585 libraries in 1,283ms.
I/flutter ( 4480): on console print
I/flutter ( 4480): on console print
I/flutter ( 4480): on console print
```



→Text Button:

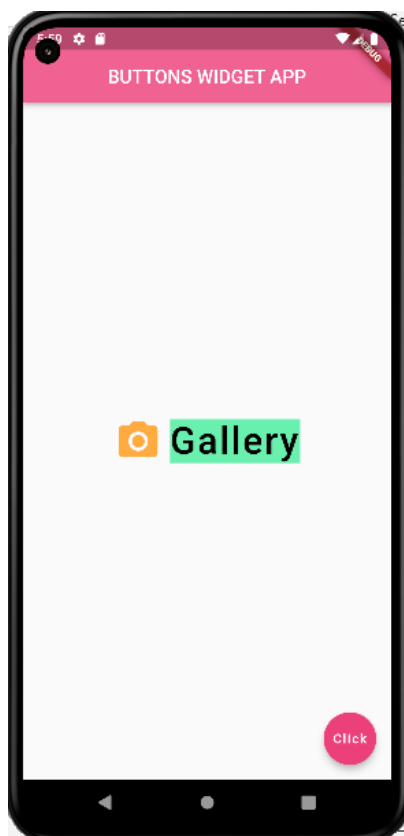
- In **flutter**, we will use **TextButton widget** to display a simple button. A text button is a label (child) displayed on a (zero elevation) Material widget. We can style the label's Text and Icon widgets using the style property of the individual widgets. The button reacts to touches by filling with the style's `ButtonStyle.backgroundColor`.


```

body: Center(
  child: TextButton.icon(
    icon: Icon(
      Icons.photo_camera,
      color: Colors.orangeAccent,
      size: 50.0,
    ), // Icon
    label: Text(
      "Gallery",
      style: TextStyle(
        color: Colors.black,
        fontSize: 40.0,
        letterSpacing: 2.0,
        backgroundColor: Colors.greenAccent,
      ), // TextStyle
      textAlign: TextAlign.start,
    ), // Text
    onPressed: () {},
  ), // TextButton.icon
),

```

Output:

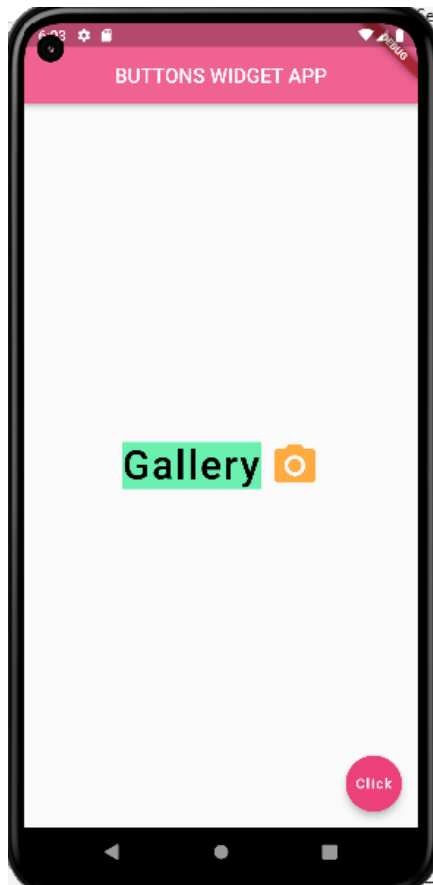


→ Directionality:

- Creates a widget that determines the directionality of text and text-direction-sensitive render objects.
- The `textDirection` and `child` arguments must not be null.

```
body: Center(  
  child: Directionality(  
    textDirection: TextDirection.rtl,  
    child: TextButton.icon(  
      icon: Icon(  
        Icons.photo_camera,  
        color: Colors.orangeAccent,  
        size: 50.0,  
      ), // Icon  
      label: Text(  
        "Gallery",  
        style: TextStyle(  
          color: Colors.black,  
          fontSize: 40.0,  
          letterSpacing: 2.0,  
          backgroundColor: Colors.greenAccent,  
        ), // TextStyle  
        textAlign: TextAlign.start,  
      ), // Text  
      onPressed: () {},  
    ),  
  ),  
)
```

Output:



Code link :

<https://github.com/nishu-mns/SDP/lab7>