import 'package:flutter/material.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  static const String \_title = 'Flutter Stateful Clicker Counter';

  // This widget is the root of your application.

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: \_title,

      theme: ThemeData(

        // useMaterial3: false,

        primarySwatch: Colors.blue,

      ),

      home: const MyHomePage(),

    );

  }

}

class MyHomePage extends StatefulWidget {

  const MyHomePage({super.key});

  @override

  \_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

  int \_counter = 0;

  void \_incrementCounter() {

    setState(() {

      \_counter++;

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text('Flutter Demo Click Counter'),

      ),

      body: Center(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: <Widget>[

            const Text(

              'You have pushed the button this many times:',

            ),

            Text(

              '$\_counter',

              style: const TextStyle(fontSize: 25),

            ),

          ],

        ),

      ),

      floatingActionButton: FloatingActionButton(

        onPressed: \_incrementCounter,

        tooltip: 'Increment',

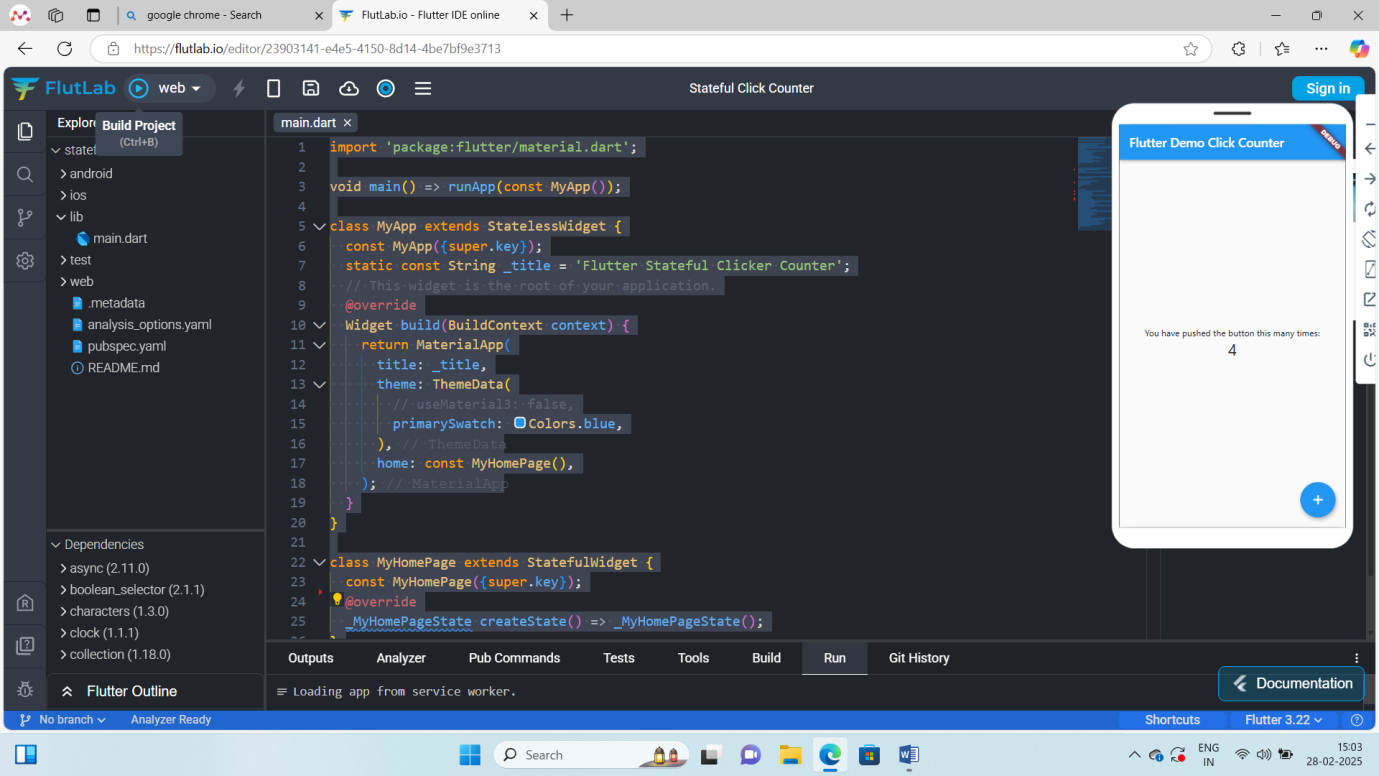
        child: const Icon(Icons.add),

      ),

    );

  }

}



Stateless widget

* Stateless widget are the widgets that don’t change.
* Its appearance and properties remain unchanged throughout the lifetime of the widgets.
* To create stateless widget ,we have to override the build()method.
* Eg:Text , Raised Button

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

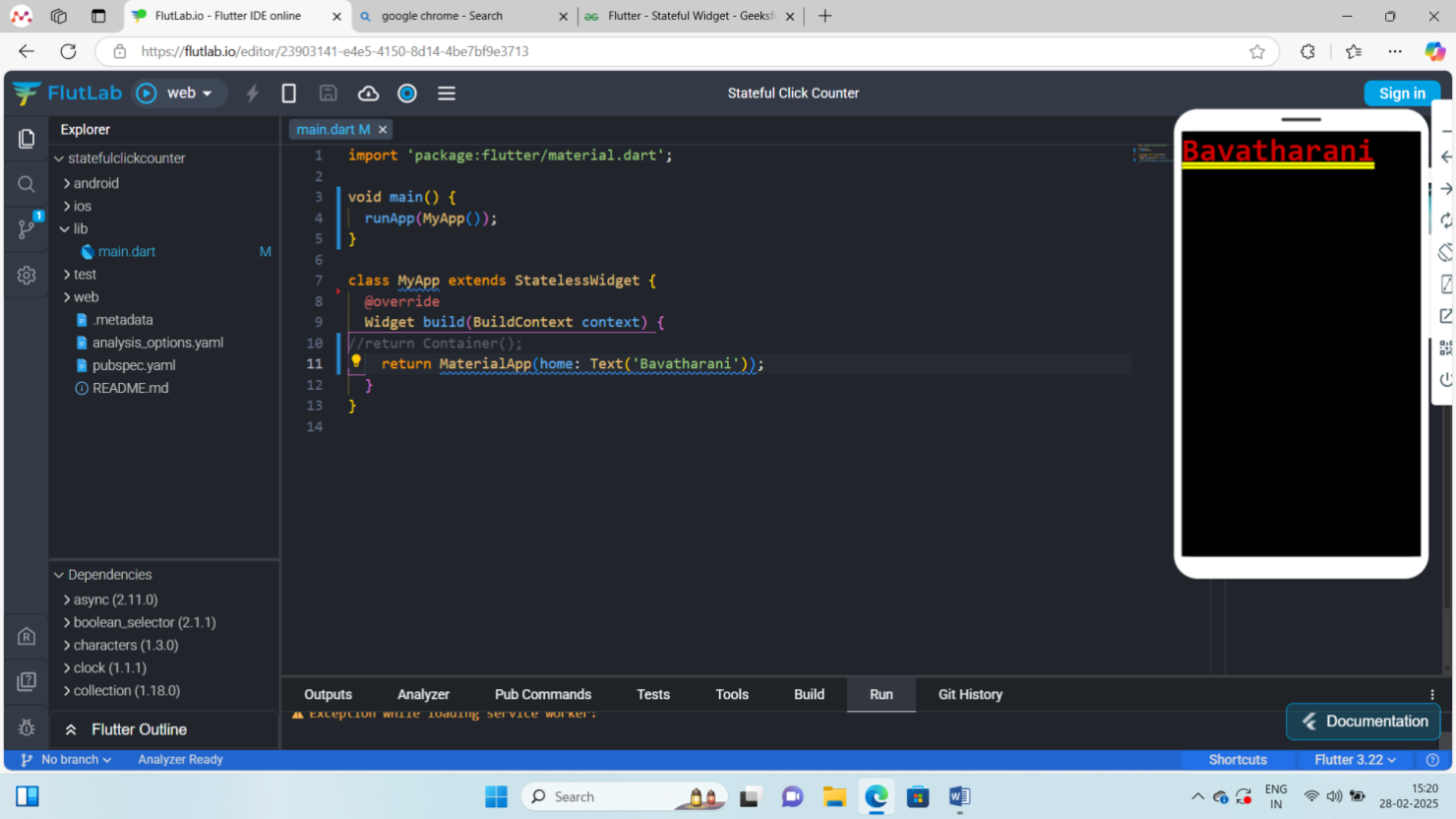
  Widget build(BuildContext context) {

//return Container();

    return MaterialApp(home: Text('Bavatharani'));

  }

}



**StateFul widget**

* Widgets that changes its properties during runtime are stateful widgets.
* To create a stateless widget ,we have to override the createState() method , which returns the state of the widgets.
* Eg:CheckBox, RadioButtoon

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  @override

  Widget build(BuildContext context) {

    //return Container();

    return MaterialApp(

      home: Column(

        children: <Widget>[

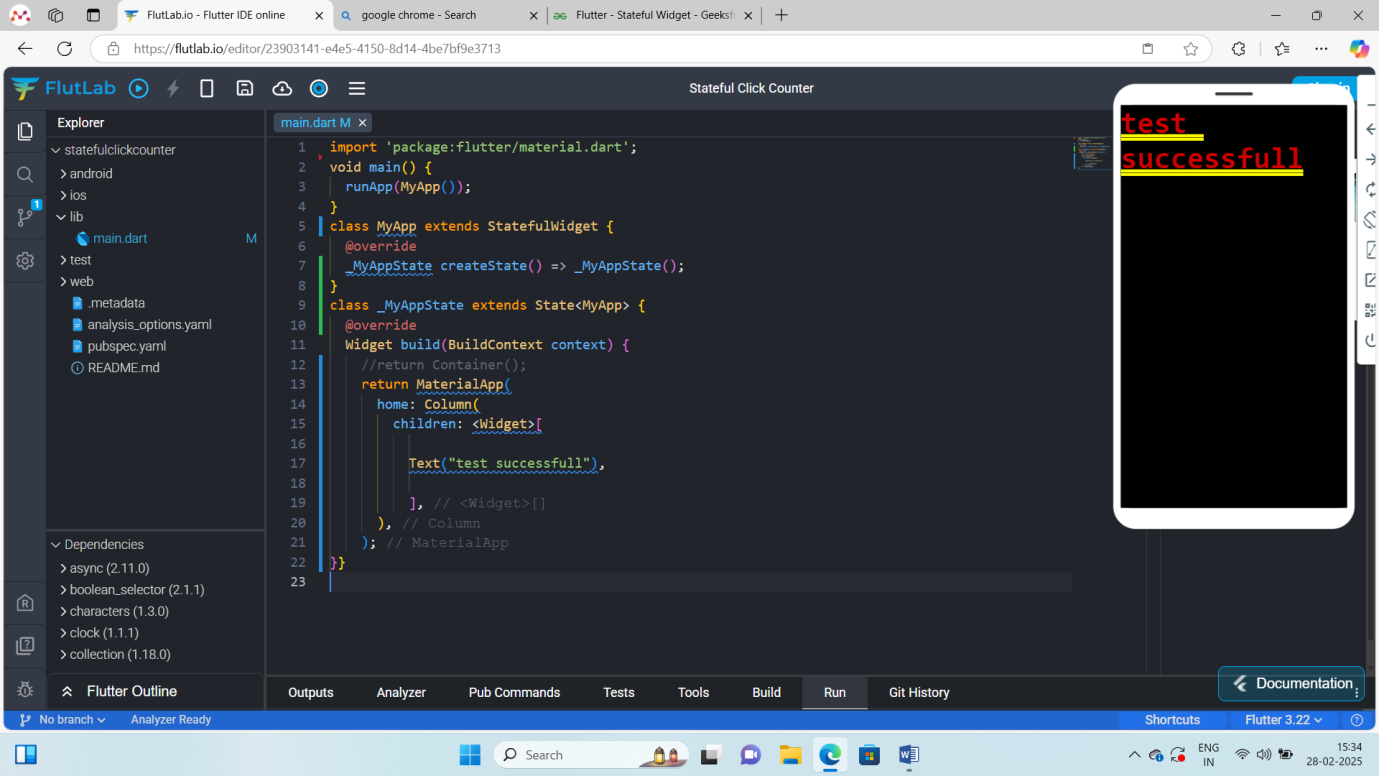
          Text("test successfull"),

          ],

      ),

    );

}}



**Onclick event**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  String value = 'Test';

  void clickme() {

    setState(() {

      value = "Bavatharani";

    });

  }

  @override

  Widget build(BuildContext context) {

    //return Container();

    return MaterialApp(

      home: Column(

        children: <Widget>[

          Text('$value'),

          FloatingActionButton(

            child: Icon(Icons.add),

            onPressed: clickme,

          )

        ],

      ),

    );

  }

}

Safe area

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

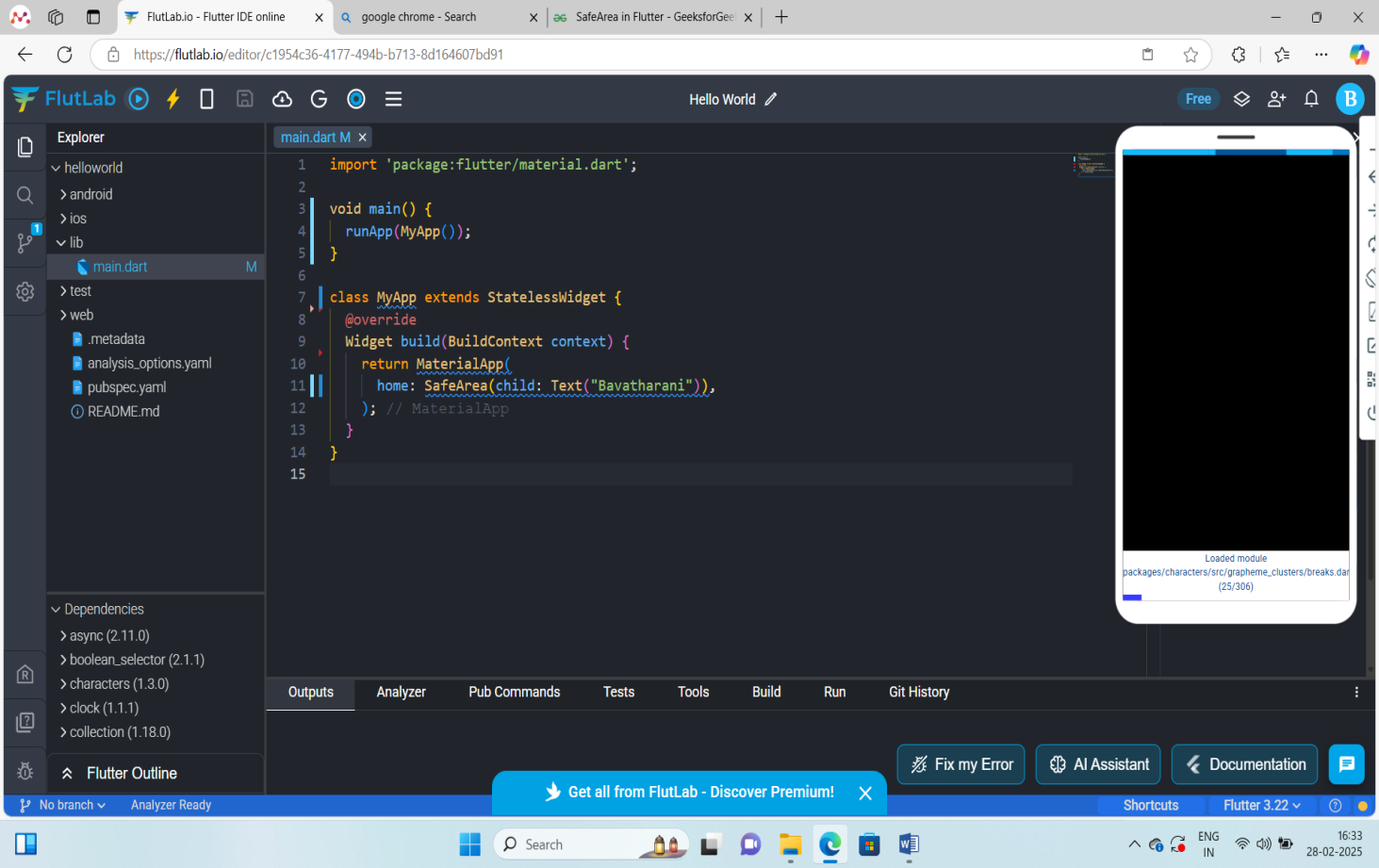
    return MaterialApp(

      home: SafeArea(child: Text("Bavatharani")),

    );

  }

}

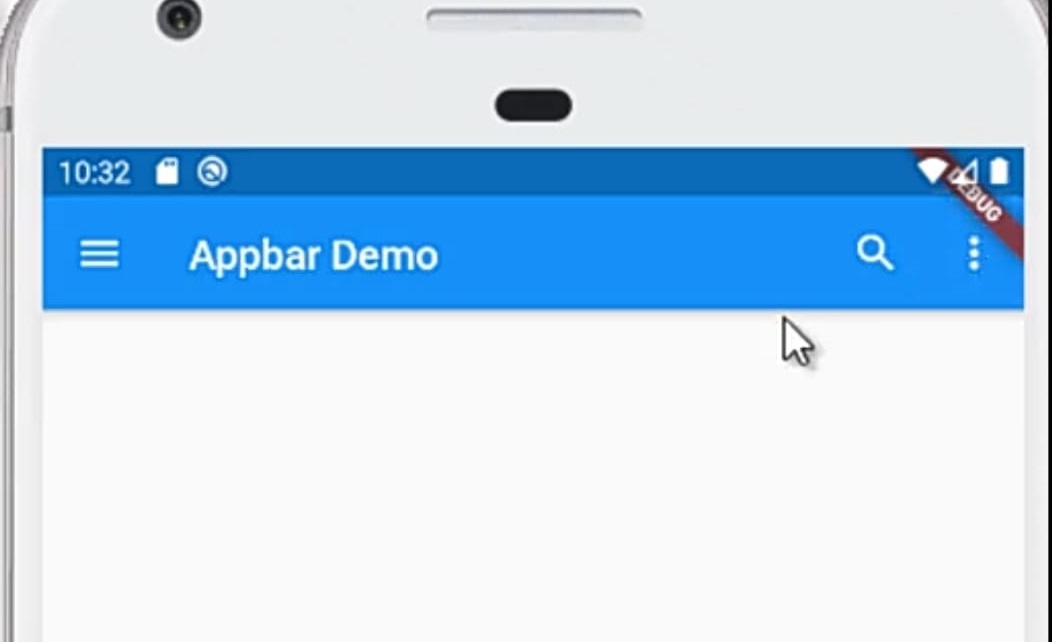


AppBar

AppBar is usually the topmost component of the app (or sometimes the bottom-most), it contains the toolbar and some other common action buttons.



import 'package:flutter/material.dart';  
  
void main() =>runApp(new MaterialApp(  
 home:new MyApp(),  
));  
class MyApp extends StatefulWidget  
{  
 \_MyAppState createState ()=>\_MyAppState();  
  
}  
  
class \_MyAppState extends State<MyApp>  
{  
 Widget build (BuildContext context)  
 {  
 return Scaffold(  
 appBar:AppBar(  
 leading:IconButton(  
 icon:Icon(Icons.*menu*),  
 onPressed: ()  
 {  
 print("Icon button click");  
 },  
 ),  
 title:Text('Appbar Demo'),  
 actions: <Widget>[  
 IconButton(  
 onPressed: (){  
 print("this is search");  
 },   
 icon: Icon(Icons.*search*)  
 ),  
 IconButton(  
 onPressed: (){  
 print("this is more\_vert");  
 },  
 icon: Icon(Icons.*more\_vert*)  
 ),  
 ],  
 ),  
  
 );  
 }  
}



import 'package:flutter/material.dart';  
  
void main() =>runApp(new MaterialApp(  
 home:new MyApp(),  
));  
class MyApp extends StatefulWidget  
{  
 \_MyAppState createState ()=>\_MyAppState();  
  
}  
  
class \_MyAppState extends State<MyApp>  
{  
 Widget build (BuildContext context)  
 {  
 return Scaffold(  
 appBar:AppBar(  
 leading:IconButton(  
 icon:Icon(Icons.*menu*),  
 onPressed: ()  
 {  
 print("Icon button click");  
 },  
 ),  
 title:Text('Appbar Demo'),  
 actions: <Widget>[  
 IconButton(  
 onPressed: (){  
 print("this is search");  
 },  
 icon: Icon(Icons.*search*)  
 ),  
 IconButton(  
 onPressed: (){  
 print("this is more\_vert");  
 },  
 icon: Icon(Icons.*more\_vert*)  
 )  
 ],  
 flexibleSpace: SafeArea(  
 child:Icon(  
 Icons.*camera*,  
 color:Colors.*white*,  
 size:55.0,  
 ),  
 ),  
   
 bottom:PreferredSize(  
 child:Container(  
 color: Colors.*grey*,  
 height:75.0,  
 width: double.*infinity*,  
 child:Text("Bavatharani",style: TextStyle(color: Colors.*white*,  
 ),  
 ),  
 ),  
 preferredSize:Size.fromHeight(75.0),  
 ),  
 ),  
  
 );  
 }  
}





Scaff fold example

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Scaffold Example',

home: MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

int \_selectedIndex = 0;

// Method to handle bottom navigation bar item taps

void \_onItemTapped(int index) {

setState(() {

\_selectedIndex = index;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Scaffold Example'),

foregroundColor: Colors.white,

backgroundColor: Colors.green,

),

body: Center(

child: Text(

'Selected Index: $\_selectedIndex',

style: TextStyle(fontSize: 24, color: Colors.black),

),

),

floatingActionButton: FloatingActionButton(

onPressed: () {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Floating Action Button Pressed')),

);

},

child: Icon(

Icons.add,

color: Colors.white,

),

backgroundColor: Colors.green,

),

drawer: Drawer(

child: ListView(

padding: EdgeInsets.zero,

children: <Widget>[

DrawerHeader(

decoration: BoxDecoration(

color: Colors.green,

),

child: Text(

'GeeksforGeeks',

style: TextStyle(

color: Colors.white,

fontSize: 24,

),

),

),

ListTile(

title: Text('Item 1'),

onTap: () {

// Handle item 1 tap

Navigator.pop(context);

},

),

ListTile(

title: Text('Item 2'),

onTap: () {

// Handle item 2 tap

Navigator.pop(context);

},

),

],

),

),

endDrawer: Drawer(

child: ListView(

padding: EdgeInsets.zero,

children: <Widget>[

DrawerHeader(

decoration: BoxDecoration(

color: Colors.blue,

),

child: Text(

'End Drawer',

style: TextStyle(

color: Colors.white,

fontSize: 24,

),

),

),

ListTile(

title: Text('End Drawer Item 1'),

onTap: () {

Navigator.pop(context);

},

),

],

),

),

bottomNavigationBar: BottomNavigationBar(

items: const <BottomNavigationBarItem>[

BottomNavigationBarItem(

icon: Icon(Icons.home),

label: 'Home',

),

BottomNavigationBarItem(

icon: Icon(Icons.search),

label: 'Search',

),

BottomNavigationBarItem(

icon: Icon(Icons.person),

label: 'Profile',

),

],

currentIndex: \_selectedIndex,

selectedItemColor: Colors.green,

onTap: \_onItemTapped,

),

persistentFooterButtons: <Widget>[

TextButton(

onPressed: () {},

child: Text('Footer Button 1'),

),

TextButton(

onPressed: () {},

child: Text('Footer Button 2'),

),

],

backgroundColor: Colors.white,

resizeToAvoidBottomInset: false,

drawerScrimColor: Colors.black.withOpacity(0.5),

);

}

}

**Text widget**

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

// This widget is

//the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'RichText',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: const MyHomePAGE(),

debugShowCheckedModeBanner: false,

);

}

}

class MyHomePAGE extends StatefulWidget {

const MyHomePAGE({Key? key}) : super(key: key);

@override

\_MyHomePAGEState createState() => \_MyHomePAGEState();

}

class \_MyHomePAGEState extends State<MyHomePAGE> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('GeeksforGeeks'),

foregroundColor: Colors.white,

backgroundColor: Colors.green,),

body: Center(

child: RichText(

// Controls visual overflow

overflow: TextOverflow.clip,

// Controls how the text should be aligned horizontally

textAlign: TextAlign.end,

// Control the text direction

textDirection: TextDirection.rtl,

// Whether the text should break at soft line breaks

softWrap: true,

// Maximum number of lines for the text to span

textScaler: TextScaler.linear(1),

maxLines: 1,

text: TextSpan(

text: 'Hello ',

style: DefaultTextStyle.of(context).style,

children: const <TextSpan>[

TextSpan(

text: 'Geeks', style: TextStyle(fontWeight: FontWeight.bold)),

],

),

)),

backgroundColor: Colors.lightBlue[50],

);

}

}

Row Widget

import 'package:flutter/material.dart';

//function to trigger build

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'GeeksForGeeks',

theme: ThemeData(

primarySwatch: Colors.green,

),// ThemeData

home: const MyHomePage(),

debugShowCheckedModeBanner: false,

);// MaterialApp

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key}) : super(key: key);

@override

// ignore: library\_private\_types\_in\_public\_api

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("GeeksForGeeks"),

),// AppBar

// App body consists of single Row

// Row consists of three children widgets

body: Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: <Widget>[

Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10), color: Colors.green),

child: const Padding(

padding: EdgeInsets.all(8.0),

child: Text(

"Geeks",

style: TextStyle(color: Colors.white, fontSize: 25),

),

),

),

Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10), color: Colors.green),

child: const Padding(

padding: EdgeInsets.all(8.0),

child: Text(

"For",

style: TextStyle(color: Colors.white, fontSize: 25),

),

),

),

Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10), color: Colors.green),

child: const Padding(

padding: EdgeInsets.all(8.0),

child: Text(

"Geeks",

style: TextStyle(color: Colors.white, fontSize: 25),

),

),

)

],

),

);

}

}

TEXT BUTTON

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_State createState() => \_State();

}

class \_State extends State<MyApp> {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text('My Text Button'),

        ),

        body: Center(

          child: TextButton(

            style: TextButton.styleFrom(

              backgroundColor: Colors.red, // Background color

              padding: EdgeInsets.all(15.0),

            ),

            child: Text(

              "Click Me",

              style: TextStyle(color: Colors.white), // Text color

            ),

            onPressed: () {

              print("hai");

            },

          ),

        ),

      ),

    );

  }

}

ELEVATED BUTTON

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text('Elevated Button Example'),

          backgroundColor: Colors.blue,

        ),

        body: Center(

          child: ElevatedButton(

            style: ElevatedButton.styleFrom(

              backgroundColor: Colors.green, // Button background color

              padding: EdgeInsets.symmetric(horizontal: 20, vertical: 15),

              textStyle: TextStyle(fontSize: 18),

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(10), // Rounded corners

              ),

            ),

            child: Text(

              "Click Me",

              style: TextStyle(color: Colors.white), // Text color

            ),

            onPressed: () {

              print("Elevated Button Clicked!");

            },

          ),

        ),

      ),

    );

  }

}

ICON BUTTON WIDGET

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text('Icon Button Example'),

          backgroundColor: Colors.blue,

        ),

        body: Center(

          child: IconButton(

            icon: Icon(Icons.thumb\_up), // Icon inside the button

            color: Colors.green, // Icon color

            iconSize: 50.0, // Size of the icon

            tooltip: 'Like', // Tooltip when long-pressed

            onPressed: () {

              print("Icon Button Clicked!");

            },

          ),

        ),

      ),

    );

  }

}

Volume increase and decrease example

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  double \_volume = 1.0; // Initial volume level

  void \_increaseVolume() {

    setState(() {

     \_volume += 1; // Increase volume, max is 10

    });

  }

  void \_decreaseVolume() {

    setState(() {

      if (\_volume > 0) \_volume -= 1; // Decrease volume, min is 0

    });

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text('Volume Control'),

          backgroundColor: Colors.blue,

        ),

        body: Center(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              Text(

                "Volume: $\_volume", // Display volume level

                style: TextStyle(fontSize: 24),

              ),

              SizedBox(height: 20),

              Row(

                mainAxisAlignment: MainAxisAlignment.center,

                children: [

                  IconButton(

                    icon: Icon(Icons.volume\_down),

                    color: Colors.red,

                    iconSize: 30,

                    onPressed: \_decreaseVolume,

                  ),

                  SizedBox(width: 20),

                  IconButton(

                    icon: Icon(Icons.volume\_up),

                    color: Colors.green,

                    iconSize: 30,

                    onPressed: \_increaseVolume,

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

}

Network image or online image upload

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text("Network Image Example"),

          backgroundColor: Colors.blue,

        ),

        body: Center(

          child: Image.network(

            'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTtDZ3-Yw3bgKq-QSpxuvlERsUfa77gz2JJcw&s', // Network Image URL

            width: 300,

            height: 500,

            fit: BoxFit.cover, // Adjust how the image is displayed

            loadingBuilder: (context, child, loadingProgress) {

              if (loadingProgress == null) {

                return child; // Image is fully loaded

              } else {

                return Center(

                  child: CircularProgressIndicator(), // Show loading indicator

                );

              }

            },

            errorBuilder: (context, error, stackTrace) {

              return Center(

                child: Icon(Icons.error, size: 50, color: Colors.red), // Error icon

              );

            },

          ),

        ),

      ),

    );

  }

}

Card widget in flutter

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text("Card Widget Example"),

          backgroundColor: Colors.blue,

        ),

        body: Center(

          child: Card(

            elevation: 8, // Adds shadow effect

            shape: RoundedRectangleBorder(

              borderRadius: BorderRadius.circular(15), // Rounded corners

            ),

            child: Container(

              width: 300,

              padding: EdgeInsets.all(20),

              child: Column(

                mainAxisSize: MainAxisSize.min,

                children: [

                  Icon(Icons.account\_circle, size: 50, color: Colors.blue),

                  SizedBox(height: 10),

                  Text(

                    "John Doe",

                    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),

                  ),

                  Text("Flutter Developer"),

                  SizedBox(height: 10),

                  ElevatedButton(

                    onPressed: () {

                      print("Card Button Clicked!");

                    },

                    child: Text("Follow"),

                  ),

                ],

              ),

            ),

          ),

        ),

      ),

    );

  }

}

Single child scrollview widget

Vertical Scrolling

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(title: Text("SingleChildScrollView Example")),

        body: SingleChildScrollView(

          child: Column(

            children: List.generate(20, (index) => ListTile(

              leading: Icon(Icons.star, color: Colors.blue),

              title: Text("Item ${index + 1}"),

            )),

          ),

        ),

      ),

    );

  }

}

Horizontal Scrolling

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(title: Text("Horizontal Scrolling Example")),

        body: SingleChildScrollView(

          scrollDirection: Axis.horizontal,

          child: Row(

            children: List.generate(10, (index) => Padding(

              padding: EdgeInsets.all(8.0),

              child: Container(

                width: 100,

                height: 100,

                color: Colors.blueAccent,

                child: Center(child: Text("Box ${index + 1}", style: TextStyle(color: Colors.white))),

              ),

            )),

          ),

        ),

      ),

    );

  }

}

Simple calculator

import 'package:flutter/material.dart';

void main() {

  runApp(CalculatorApp());

}

class CalculatorApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: CalculatorScreen(),

    );

  }

}

class CalculatorScreen extends StatefulWidget {

  @override

  \_CalculatorScreenState createState() => \_CalculatorScreenState();

}

class \_CalculatorScreenState extends State<CalculatorScreen> {

  TextEditingController num1Controller = TextEditingController();

  TextEditingController num2Controller = TextEditingController();

  double result = 0;

  void calculate(String operation) {

    double num1 = double.tryParse(num1Controller.text) ?? 0;

    double num2 = double.tryParse(num2Controller.text) ?? 0;

    setState(() {

      if (operation == '+') {

        result = num1 + num2;

      } else if (operation == '-') {

        result = num1 - num2;

      } else if (operation == '\*') {

        result = num1 \* num2;

      } else if (operation == '/') {

        result = num2 != 0 ? num1 / num2 : 0;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Simple Calculator')),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            TextField(

              controller: num1Controller,

              keyboardType: TextInputType.number,

              decoration: InputDecoration(labelText: 'Enter first number'),

            ),

            TextField(

              controller: num2Controller,

              keyboardType: TextInputType.number,

              decoration: InputDecoration(labelText: 'Enter second number'),

            ),

            SizedBox(height: 20),

            Row(

              mainAxisAlignment: MainAxisAlignment.spaceEvenly,

              children: [

                ElevatedButton(onPressed: () => calculate('+'), child: Text('+')),

                ElevatedButton(onPressed: () => calculate('-'), child: Text('-')),

                ElevatedButton(onPressed: () => calculate('\*'), child: Text('\*')),

                ElevatedButton(onPressed: () => calculate('/'), child: Text('/')),

              ],

            ),

            SizedBox(height: 20),

            Text('Result: $result', style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),

          ],

        ),

      ),

    );

  }

}

Screen Navigation in flutter App with data handling

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: FirstPage(),

    );

  }

}

class FirstPage extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("page routing"),

      ),

      body: Center(

        child: Text(

          'page-1',

          style: TextStyle(

            fontSize: 25.0,

            fontWeight: FontWeight.bold,

          ),

        ),

      ),

      floatingActionButton: FloatingActionButton(

        child: Icon(Icons.navigate\_next),

        onPressed: () {

          Navigator.of(context)

              .push(MaterialPageRoute(builder: (context) => SecondPage()));

        },

      ),

    );

  }

}

class SecondPage extends StatelessWidget {

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("page routing"),

      ),

      body: Center(

        child: Text(

          'page-2',

          style: TextStyle(

            fontSize: 25.0,

            fontWeight: FontWeight.bold,

          ),

        ),

      ),

      floatingActionButton: FloatingActionButton(

        child: Icon(Icons.navigate\_next),

        onPressed: () {

          Navigator.of(context)

              .pop(MaterialPageRoute(builder: (context) => FirstPage()));

        },

      ),

    );

  }

}

Data table

One column one row

import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(

home: MyApp(),

));

}

class MyApp extends StatefulWidget {

  \_State createState() => \_State();

}

class \_State extends State<MyApp>

{

  Widget build(BuildContext context)

  {

    return Scaffold(

      appBar: AppBar(

        title:Text('Data Table'),

      ),

      body:DataTable(

        columns:[DataColumn(label: Text("Roll No",style:TextStyle(color: Colors.red)))],

        rows:[

         DataRow(cells:[DataCell(Text("A1209027"))]),

         DataRow(cells:[DataCell(Text("A1209028"))]),

         DataRow(cells:[DataCell(Text("A1209029"))]),

         DataRow(cells:[DataCell(Text("A1209030"))]),

        ],

      ),

    );

  }

}

Mutliple row and multiple column

import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(

home: MyApp(),

));

}

class MyApp extends StatefulWidget {

  \_State createState() => \_State();

}

class \_State extends State<MyApp>

{

  Widget build(BuildContext context)

  {

    return Scaffold(

      appBar: AppBar(

        title:Text('Data Table'),

      ),

      body:DataTable(

        columns:[

          DataColumn(label: Text("Roll No",style:TextStyle(color: Colors.red))),

          DataColumn(label: Text("Name",style:TextStyle(color: Colors.red))),

        ],

        rows:[

         DataRow(cells:[DataCell(Text("A1209027")),DataCell(Text("bava"))]),

         DataRow(cells:[DataCell(Text("A1209028")),DataCell(Text("thara"))]),

         DataRow(cells:[DataCell(Text("A1209029")),DataCell(Text("xyz"))]),

         DataRow(cells:[DataCell(Text("A1209030")),DataCell(Text("abc"))]),

        ],

      ),

    );

  }

}

CheckBox in flutter

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  bool \_isChecked = false;

  void \_toggleCheckbox(bool? value) {

    setState(() {

      \_isChecked = value!;

    });

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: Text('Checkbox Example'),

        ),

        body: Center(

          child: Row(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              Checkbox(

                value: \_isChecked,

                onChanged: \_toggleCheckbox,

              ),

              Text('I agree to the terms and conditions'),

            ],

          ),

        ),

      ),

    );

  }

}

Slider Widget in flutter

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: SliderExample(),

    );

  }

}

class SliderExample extends StatefulWidget {

  @override

  \_SliderExampleState createState() => \_SliderExampleState();

}

class \_SliderExampleState extends State<SliderExample> {

  double \_currentValue = 20;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Flutter Slider Example')),

      body: Center(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            Text(

              'Value: ${\_currentValue.toStringAsFixed(1)}',

              style: TextStyle(fontSize: 20),

            ),

            Slider(

              value: \_currentValue,

              min: 0,

              max: 100,

              divisions: 10, // Optional: Creates discrete steps

              label: \_currentValue.round().toString(), // Shows label on hover

              onChanged: (double value) {

                setState(() {

                  \_currentValue = value;

                });

              },

              activeColor: Colors.blue, // Color of active track

              inactiveColor: Colors.grey, // Color of inactive track

            ),

          ],

        ),

      ),

    );

  }

}

Alert Dialog widget in Flutter

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: AlertDialogExample(),

    );

  }

}

class AlertDialogExample extends StatelessWidget {

  void \_showAlertDialog(BuildContext context) {

    showDialog(

      context: context,

      builder: (BuildContext context) {

        return AlertDialog(

          title: Text("Alert"),

          content: Text("This is a simple alert dialog in Flutter."),

          actions: [

            TextButton(

              onPressed: () {

                Navigator.of(context).pop(); // Close the dialog

              },

              child: Text("OK"),

            ),

          ],

        );

      },

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text("AlertDialog Example")),

      body: Center(

        child: ElevatedButton(

          onPressed: () => \_showAlertDialog(context),

          child: Text("Show AlertDialog"),

        ),

      ),

    );

  }

}

Image carousel

import 'package:flutter/material.dart';

import 'package:carousel\_slider/carousel\_slider.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: "Carousel Slider",

      debugShowCheckedModeBanner: false,

      theme: ThemeData(primarySwatch: Colors.green),

      home: HomePage(),

    );

  }

}

class HomePage extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        foregroundColor: Colors.white,

        backgroundColor: Colors.green,

        title: Text("GFG Slider"),

      ),

      body: ListView(

        children: [

          CarouselSlider(

            items: [

              buildImage("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRlTvWSI0lmgrwHWGlMWzFDsI9bIoZGKqy0ZA&s"),

              buildImage("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3KLsKw0jLKi6EOWlMs2QnOvqlopxW-8i54w&s"),

              buildImage("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3KLsKw0jLKi6EOWlMs2QnOvqlopxW-8i54w&s"),

              buildImage("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3KLsKw0jLKi6EOWlMs2QnOvqlopxW-8i54w&s"),

              buildImage("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3KLsKw0jLKi6EOWlMs2QnOvqlopxW-8i54w&s"),

            ],

            options: CarouselOptions(

              height: 180.0,

              enlargeCenterPage: true,

              autoPlay: true,

              aspectRatio: 16 / 9,

              autoPlayCurve: Curves.fastOutSlowIn,

              enableInfiniteScroll: true,

              autoPlayAnimationDuration: Duration(milliseconds: 800),

              viewportFraction: 0.8,

            ),

          ),

        ],

      ),

    );

  }

  Widget buildImage(String url) {

    return Container(

      margin: EdgeInsets.all(6.0),

      decoration: BoxDecoration(

        borderRadius: BorderRadius.circular(8.0),

        image: DecorationImage(

          image: NetworkImage(url),

          fit: BoxFit.cover,

        ),

      ),

    );

  }

}

pageViewer

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root

// of your application.

@override

Widget build(BuildContext context) {

  return MaterialApp(

  title: 'PageView',

  theme: ThemeData(

    primarySwatch: Colors.blue,

  ),

  debugShowCheckedModeBanner: false,

  home: MyHomePage(),

  );

}

}

class MyHomePage extends StatefulWidget {

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

PageController controller=PageController();

List<Widget> \_list=<Widget>[

  new Center(child:new Pages(text: "Page 1",)),

  new Center(child:new Pages(text: "Page 2",)),

  new Center(child:new Pages(text: "Page 3",)),

  new Center(child:new Pages(text: "Page 4",))

];

int \_curr=0;

@override

Widget build(BuildContext context) {

  return Scaffold(

  backgroundColor: Colors.grey,

  appBar:AppBar(

    title: Text("GeeksforGeeks"),

    backgroundColor: Colors.green,

    actions: <Widget>[

    Padding(

      padding: const EdgeInsets.all(3.0),

      child: Text(

      "Page: "+(

        \_curr+1).toString()+"/"+\_list.length.toString(),textScaleFactor: 2,),

    )

    ],),

  body: PageView(

    children:

    \_list,

    scrollDirection: Axis.horizontal,

    // reverse: true,

    // physics: BouncingScrollPhysics(),

    controller: controller,

    onPageChanged: (num){

    setState(() {

      \_curr=num;

    });

    },

  ),

  floatingActionButton:Row(

    mainAxisAlignment: MainAxisAlignment.spaceEvenly,

    children:<Widget>[

    FloatingActionButton(

      onPressed: () {

      setState(() {

      \_list.add(

        new Center(child: new Text(

        "New page", style: new TextStyle(fontSize: 35.0))),

      );

      });

      if(\_curr!=\_list.length-1)

        controller.jumpToPage(\_curr+1);

        else

        controller.jumpToPage(0);

      },

  child:Icon(Icons.add)),

  FloatingActionButton(

    onPressed: (){

    \_list.removeAt(\_curr);

    setState(() {

      controller.jumpToPage(\_curr-1);

    });

    },

  child:Icon(Icons.delete)),

    ]

  )

  );

}

}

class Pages extends StatelessWidget {

final text;

Pages({this.text});

@override

Widget build(BuildContext context) {

return Center(

  child: Column(

    mainAxisAlignment: MainAxisAlignment.center,

    children:<Widget>[

    Text(text,textAlign: TextAlign.center,style: TextStyle(

      fontSize: 30,fontWeight:FontWeight.bold),),

    ]

  ),

  );

}

}

Refresh indicator:

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: RefreshExample(),

    );

  }

}

class RefreshExample extends StatefulWidget {

  @override

  \_RefreshExampleState createState() => \_RefreshExampleState();

}

class \_RefreshExampleState extends State<RefreshExample> {

  List<String> items = ["Item 1", "Item 2", "Item 3", "Item 4", "Item 5"];

  // Simulate a refresh function

  Future<void> \_refreshData() async {

    await Future.delayed(Duration(seconds: 2)); // Simulate network call

    setState(() {

      items.add("New Item ${items.length + 1}");

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text("Pull to Refresh Example")),

      body: RefreshIndicator(

        onRefresh: \_refreshData,

        child: ListView.builder(

          itemCount: items.length,

          itemBuilder: (context, index) {

            return ListTile(

              title: Text(items[index]),

              leading: Icon(Icons.refresh),

            );

          },

        ),

      ),

    );

  }

}

Dart Programming language

It is open source purpose programming language developed by google.

It supports application development on both server and client side.

Variables

It is used to store the given data.

Syntax:

Type variable\_name

Example

void main()

{

int a=10;

double b=5.6;

bool c=false;

String d="Sathyabama";

print(a);

print(b);

print(c);

print(d);

}

Keywords

Reserved words

Final keyword

It can only be set once and its initialized when accessed.

We cannot override it

void main()

{

final int a=10;

a=11;

print(a);

}

Const in dart

Its value must be known before the program runs.

void main()

{

const  a=10;

print(a);

}

Operators in dart

Arithmetic operator

void main()

{

int a=11;

int b=2;

print(a+b);

print(a-b);

print(a\*b);

print(a/b);

print(a%b);

}

Relational operator

void main()

{

int a=11;

int b=20;

print(a<b);

print(a>b);

print(a<=b);

print(a>=b);

print(a==b);

print(a!=b);

}

Assignment operator

void main()

{

int a=11;

int b=20;

a\*=5;//a=a+5

print(a);

}

Logical operator

Logical AND

void main()

{

int a=10;

int b=20;

print(a<b && a==b);

print(a<b && a!=b);

print(a>b && a==b);

print(a>b && a!=b);

}

Logical OR

void main()

{

int a=10;

int b=20;

print(a<b || a==b);

print(a<b || a!=b);

print(a>b || a==b);

print(a>b || a!=b);

}

Conditional operator

void main()

{

int a=100;

int b=20;

String c=a>b?"a is greater than b":"a is less than b";

print(c);

}

Control statement

It controls the flow of the program.

void main()

{

int a=100;

int b=20;

if(a<b)

{

    print("a is less than b");

}

else

{

    print("a is greater than b");

}

}

Else if statement

void main()

{

int a=100;

int b=20;

if(a<b)

{

    print("a is less than b");

}

else if(a>b)

{

    print("a is greater than b");

}

else if(a==b)

{

    print("a is equal to b");

}

else

{

    print("None of these");

}

}

Loops

It repeats the same process until its satisfy the condition.

For loop

void main()

{

for(int i=1;i<=10;i++)

{

    print(i);

}

}

For in loop

void main()

{

    var A=[10,20,30,40,50,60,70,80,90,100];

for(int i in A)

{

    print(i);

}

}

For each loop

void main()

{

    var A=[10,20,30,40,50,60,70,80,90,100];

A.forEach((var i)=> print(i));

}

While loop

void main()

{

    int i=1;

    while(i<=10)

    {

        print(i);

        i++;

    }

}

Do while loop

void main()

{

    int i=11;

    do

    {

        print(i);

        i++;

    }while(i<=10);

}

Break statement

void main()

{

for(int i=1;i<=10;i++)

{

   if(i==5)

   {

    break;

   }

   else

   {

    print(i);

   }

}

}

Continue statement

void main()

{

for(int i=1;i<=10;i++)

{

   if(i==5)

   {

    continue;

   }

   else

   {

    print(i);

   }

}

}

Dart function

It is used to do a specific task.

Function call

Function definition

Example

int add(int a,int b)

{

return a+b;

}

void main()

{

int x=add(10,20);

print(x);

}

Lambda function

void add()=>print("Hello World");

void main()

{

add();

}

Class and Object

Class

It is blueprint of an object

It is a virtual thing or common thing.

Example :Human

Object

It is a physical thing ,we can see it

Example: male ,female

class first

{

    void add()

    {

print("Hello guys good evening");

    }

}

void main()

{

first obj=new first();

obj.add();

}

class first

{

    String a="Welcome";

    void add()

    {

print("Hello guys good evening");

    }

}

void main()

{

first obj=new first();

print(obj.a);

obj.add();

}

**Inheritance**

One class can inherit another class

**Single inheritance**

class first

{

void add()

{

print("this is first class");

}

}

class second extends first

{

void sub()

{

    print("this is second class");

}

}

void main()

{

second obj=new second();

obj.add();

obj.sub();

}

Multilevel inheritance

class first

{

void add()

{

print("this is first class");

}

}

class second extends first

{

void sub()

{

    print("this is second class");

}

}

class third extends second

{

void mul()

{

    print("this is multiplication function");

}

}

void main()

{

third obj=new third();

obj.add();

obj.sub();

obj.mul();

}

**Hierarchical inheritance**

class first

{

void add()

{

print("this is first class");

}

}

class second extends first

{

void sub()

{

    print("this is second class");

}

}

class third extends first

{

void mul()

{

    print("this is multiplication function");

}

}

void main()

{

third obj=new third();

obj.add();

obj.mul();

}

**Constructor**

It is a special method

Initialize of an object

Default constructor

class first

{

first()

{

print("this is constructor");

}

}

void main()

{

first obj=new first();

}

**Parameterized constructor**

class first

{

first(String a)

{

print("this is constructor"+a);

}

}

void main()

{

first obj=new first("dora");

}

**Polymorphism**

One in many form.

**Example:** teacher

Method overriding

Two different classes but same function name with different arguments.

class first

{

void sum()

{

print("this is first class");

}

}

class second extends first

{

void sum()

{

print("this is second class");

}

}

void main()

{

second obj=new second();

obj.sum();

}

**Method overloading**

Since dart is a dynamically typed language , it doesnot support function overloading.because function overloading require static data type.

This keyword

class mob

{

String mobile="";

Car(String mobile)

{

this.mobile=mobile;

print("this mobile is:${mobile}");

}

}

void main()

{

mob m1=new mob();

m1.Car("volvo");

}

Exception

void main()

{

try{

var b=10/0;

print(b);

}

catch(e)

{

print(e);

}

}

**Asynchronous programming: futures, async, await**

What is future ?

A future (lower case "f") is an instance of the [Future](https://api.dart.dev/dart-async/Future-class.html) (capitalized "F") class. A future represents the result of an asynchronous operation, and can have two states: uncompleted or completed.

### Uncompleted

When you call an asynchronous function, it returns an uncompleted future. That future is waiting for the function's asynchronous operation to finish or to throw an error.

### Completed

If the asynchronous operation succeeds, the future completes with a value. Otherwise, it completes with an error.

**Example :**

Future<void> fetchUserOrder() {

// Imagine that this function is fetching user info from another service or database.

return Future.delayed(const Duration(seconds: 2), () => print('Large Latte'));

}

void main()

{

fetchUserOrder();

print('Fetching user order...');

}

## async and await

The async and await keywords provide a declarative way to define asynchronous functions and use their results. Remember these two basic guidelines when using async and await:

* To define an async function, add async before the function body:
* The await keyword works only in async functions.

|  |  |
| --- | --- |
| synchronous function String createOrderMessage() {  var order = fetchUserOrder();  return 'Your order is: $order';  }  Future<String> fetchUserOrder() =>  // Imagine that this function is  // more complex and slow.  Future.delayed(const Duration(seconds: 2), () => 'Large Latte');  void main() {  print('Fetching user order...');  print(createOrderMessage());  } | Asynchronous function Future<String> createOrderMessage() async {  var order = await fetchUserOrder();  return 'Your order is: $order';  }  Future<String> fetchUserOrder() =>  // Imagine that this function is  // more complex and slow.  Future.delayed(const Duration(seconds: 2), () => 'Large Latte');  Future<void> main() async {  print('Fetching user order...');  print(await createOrderMessage());  } |

Stream

A **Stream** is a sequence of **asynchronous events.** It is like an asynchronous Iterable—where, instead of getting the next event when you ask for it, the stream tells you that there is an event when it is ready.

In other words, streams are a source of asynchronous events delivered sequentially. There are data events, which are sometimes referred to as elements of the stream due to a stream’s similarity to a list, and there are error events, which are notifications of failure. Once all data elements have been emitted, a special event signaling the stream is done will notify any listeners that there is no more.

* A stream is like a pipe, you put a value on the one end, and if there’s a listener on the other end that listener will receive that value.
* You can process a stream using either await for or listen() from the Stream API.

**Example**

Future<int> sumStream(Stream<int> stream) async {

var sum=0;

await for(var value in stream) {

sum += value;

}

return sum;

}

Future<void> main() async {

final stream = Stream<int>.fromIterable([1,2,3,4,5]);

final sum = await sumStream(stream);

print('Sum: $sum');

}

In this code simply receives each event of a stream of integer events, adds them up, and returns (a future of) the sum. When the loop body ends, the function is paused until the next event arrives or the stream is done.

In function, we have used the **async keyword,** which is required when using the **await for** a loop.

**Flutter – Working with Layouts**

all the images, icons, labels and text, etc are technically widgets of different types and layouts.



The common layout widget can be divided into two categories:

1. Standard Widgets
2. Material Widgets

### ****Container:****

The container is the most used widget in flutter. We can add padding, margin, border, and background color-like properties in this widget and we can customize it according to our requirement. It contains only a single widget or child.

**Example:**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: Home(title: 'Flutter App'),

    );

  }

}

class Home extends StatefulWidget {

  final String title;

  const Home({Key? key, required this.title}) : super(key: key);

  @override

  \_HomeState createState() => \_HomeState();

}

class \_HomeState extends State<Home> {

  @override

  Widget build(BuildContext context) {

    var size = MediaQuery.of(context).size;

    return Scaffold(

      appBar: AppBar(

        title: Text(widget.title), // Now `widget.title` is valid

      ),

      body: SafeArea(

        child: Container(

          height: 100.0,

          width: 100.0,

          color: Colors.green,

        ),

      ),

    );

  }

}

**List View**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: MyListScreen(),

    );

  }

}

class MyListScreen extends StatelessWidget {

  final List<Map<String, String>> names = [

    {'firstName': 'James', 'lastName': 'Thomas'},

    {'firstName': 'Ajay', 'lastName': 'Kumar'},

    {'firstName': 'Arun', 'lastName': 'Das'},

    {'firstName': 'Roxie', 'lastName': 'St'},

    {'firstName': 'Stanlee', 'lastName': 'Jr'},

    {'firstName': 'AMC', 'lastName': 'Hales'},

    {'firstName': 'Monty', 'lastName': 'Chopra'},

    {'firstName': 'Emmy', 'lastName': 'Ave'},

    {'firstName': 'Chaitanya', 'lastName': 'Kumar'},

    {'firstName': 'Rio', 'lastName': 'St'},

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        backgroundColor: Colors.green,

        title: Text("ListView Example"),

      ),

      body: ListView.separated(

        itemCount: names.length,

        itemBuilder: (context, index) {

          return name(names[index]['firstName']!, names[index]['lastName']!);

        },

        separatorBuilder: (context, index) => Divider(),

      ),

    );

  }

  ListTile name(String firstName, String lastName) {

    return ListTile(

      title: Text(

        firstName,

        style: TextStyle(fontWeight: FontWeight.w500, fontSize: 20),

      ),

      subtitle: Text(lastName),

      leading: Icon(Icons.person, color: Colors.blue[500]),

      trailing: Icon(Icons.arrow\_forward\_ios),

    );

  }

}

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: Home(title: 'Flutter App'),

    );

  }

}

class Home extends StatefulWidget {

  final String title;

  const Home({Key? key, required this.title}) : super(key: key);

  @override

  \_HomeState createState() => \_HomeState();

}

class \_HomeState extends State<Home> {

  Widget \_buildImageColumn() => Container(

        decoration: BoxDecoration(

          color: Colors.black12,

        ),

        child: Column(

          children: [

            Container(

              height: 100.0,

              width: 50.0,

              color: Colors.red,

            ),

            Container(

              height: 100.0,

              width: 50.0,

              color: Colors.yellow,

            ),

          ],

        ),

      );

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        backgroundColor: Colors.green,

        title: Text(widget.title), // Now `widget.title` works

      ),

      body: Center(

        child: \_buildImageColumn(),

      ),

    );

  }

}

**Flutter – Managing the MediaQuery Object**

During the process of developing an app for both phones and tablets, it is standard practice to have different UI layouts for different screen sizes for a better user experience. If the user has a preference set for different font sizes or wants to curtail animations. This is where MediaQuery comes into action, you can get information about the current device size, as well as user preferences, and design your layout accordingly. MediaQuery provides a higher-level view of the current app’s screen size and can also give more detailed information about the device and its layout preferences. In practice, MediaQuery is always there. It can simply be accessed by calling**MediaQuery.of**in the build method.

**Example:**

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

    @override

    Widget build(BuildContext context) {

        return MaterialApp(

            home: Home(),

            debugShowCheckedModeBanner: false,

        );

    }

}

class Home extends StatelessWidget {

var size,height,width;

@override

Widget build(BuildContext context) {

  // getting the size of the window

  size = MediaQuery.of(context).size;

  height = size.height;

  width = size.width;

  return Scaffold(

  appBar: AppBar(

    title: Text("welcome"),

    backgroundColor: Colors.green,

  ),

  body: Container(

    color: Colors.yellow,

    height: height/2,//half of the height size

    width: width/2,//half of the width size

  ),

  );

}

}

**Flutter – Gestures**

Gestures are used to interact with an application. It is generally used in touch-based devices to physically interact with the application. It can be as simple as a single tap on the screen to a more complex physical interaction like swiping in a specific direction to scrolling down an application.

import 'package:flutter/material.dart';

// Entry point of the application

void main() => runApp(const MyApp());

// Root widget of the application

class MyApp extends StatelessWidget {

  const MyApp({Key? key}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    const title = 'Gestures';

    return MaterialApp(

      title: title,

      home: const MyHomePage(title: title),

      debugShowCheckedModeBanner: false,

    );

  }

}

// Home page widget

class MyHomePage extends StatelessWidget {

  final String title;

  const MyHomePage({Key? key, required this.title}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text('Welcome'),

        backgroundColor: Colors.green,

        foregroundColor: Colors.white,

      ),

      body: const Center(child: MyButton()),

    );

  }

}

// Custom button widget with gesture detection

class MyButton extends StatelessWidget {

  const MyButton({Key? key}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    // The GestureDetector wraps the button.

    return GestureDetector(

      // Show snackbar on tap of child

      onTap: () {

        const snackBar = SnackBar(content: Text("You just Tapped on the Button"));

        ScaffoldMessenger.of(context).showSnackBar(snackBar);

      },

      // The tap button

      child: Container(

        padding: const EdgeInsets.all(12.0),

        decoration: BoxDecoration(

          color: Colors.green,

          borderRadius: BorderRadius.circular(8.0),

        ),

        child: const Text(

          'Tap Button',

          style: TextStyle(color: Colors.white),

        ),

      ),

    );

  }

}

**Display Text Over the Image**

import 'package:flutter/material.dart';

void main() {

runApp(RunMyApp());

}

class RunMyApp extends StatelessWidget {

const RunMyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

theme: ThemeData(primarySwatch: Colors.green),

home: Scaffold(

appBar: AppBar(

title: Text("Text Over Image"),

),

body: Center(

child: Stack(

children: [

Container(

alignment: Alignment.center,

child: Image.asset(

'assets/s2.png',

height: 200,

width: double.infinity,

fit: BoxFit.cover,

),

),

Container(

alignment: Alignment.center,

child: Text(

'Text Over the Image',

style: TextStyle(color: Colors.white,

fontWeight: FontWeight.bold,

fontSize: 24.0),

)),

],

),

),

) ,

);

}

}

**Animate Image Rotation**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

theme: ThemeData(

primarySwatch: Colors.green, // Set the app's primary theme color

),

debugShowCheckedModeBanner: false,

home: MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage>

with SingleTickerProviderStateMixin {

late AnimationController \_controller;

late Animation<double> \_animation;

@override

void initState() {

super.initState();

// Create an animation controller

\_controller = AnimationController(

vsync: this, // vsync is set to this for performance reasons

duration: Duration(seconds: 2), // Set the duration of the animation

);

// Create a Tween for the rotation angle

\_animation = Tween<double>(

begin: 0, // Start rotation angle

end: 2 \* 3.141, // End rotation angle (2 \* pi for a full circle)

).animate(\_controller);

// Repeat the animation indefinitely

\_controller.repeat();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Image Rotation Animation'),

),

body: Center(

child: AnimatedBuilder(

animation: \_animation,

builder: (context, child) {

// Use Transform.rotate to rotate the Image based on the animation value

return Transform.rotate(

angle: \_animation.value,

child: Image.asset(

'assets/1.png', // Replace with your image asset

width: 200,

height: 200,

),

);

},

),

),

);

}

@override

void dispose() {

// Dispose of the animation controller when the widget is disposed

\_controller.dispose();

super.dispose();

}

}