

Flutter Basics

1. What is Flutter?

Flutter is an open-source UI toolkit by Google used to build cross-platform applications using a single codebase.

Platforms supported:

- Android
- iOS
- Web
- Desktop

Why Flutter?

- Single codebase
- Hot Reload
- Rich UI using widgets
- High performance

2. What is Dart?

Flutter uses Dart programming language.

Features:

- Object-oriented
- Easy to learn
- Async support

Example:

```
void main() {  
    print("Hello Flutter");  
}
```

3. Flutter Architecture

Layers:

- Framework (Widgets)
- Engine
- Platform (Android/iOS)

4. Everything is a Widget

Text, Button, Image, Layout – everything is a widget.

5. main() Function

Entry point of Flutter app.

runApp() loads root widget.

6. StatelessWidget

Used for static UI.

Requires only build() method.

7. Scaffold

Provides basic screen structure.

Includes AppBar, Body, FAB, Drawer.

8. Layout Widgets

- Row – Horizontal
- Column – Vertical
- Center – Center alignment
- Container – Box with decoration

9. Basic Widgets

Text Widget – displays text

Image Widget – displays images

Icon Widget – displays icons

10. Buttons and Interaction

ElevatedButton handles user interaction.

11. StatefulWidget

Used for dynamic UI.

Uses setState() to update UI.

12. Navigation

Navigator.push() used to move between screens.

13. Project Structure

lib/

main.dart

screens/

widgets/

STEP 1: COMPLETE DEFAULT FLUTTER CODE (GIVEN)

```
void main() {  
  runApp(const MyApp());  
}  
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      // top-level widget of the App  
    );  
  }  
}
```

→ entry point

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

↳ starts the flutter application

↳ defines the UI of this widget

first screen of the app

load when App start

```

home: MyHomePage(),
);
}
}

```

```

class MyHomePage extends StatelessWidget {

```

```

  @override

```

```

  Widget build(BuildContext context) {

```

```

    return Scaffold(

```

```

      appBar: AppBar(

```

```

        title: Text('Flutter Demo'),

```

```

      ),

```

```

      body: Center(

```

```

        child: Text('Hello World'),

```

```

      ),

```

```

    );

```

```

  }

```

```

}

```

STEP 2: WHAT IS ALREADY GIVEN BY FLUTTER

- main() function
- runApp()
- StatelessWidget structure
- build() method
- MaterialApp

- Scaffold
- appBar
- body structure

These parts form the ready-made skeleton of the app.

STEP 3: MAIN EDITABLE AREA

appBar title text

Text inside body

Widgets inside body

- Text
- Icons
- Images
- Buttons
- Layout (Row, Column)

STEP 5: REPLACING BODY CONTENT

Replace Text with Icon:

child: Icon(Icons.star)

Replace Text with Image:

child: Image.asset('assets/image.png')

STEP 6: COMBINING WIDGETS

```
body: Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Text('My App'),  
    Icon(Icons.star),  
    ElevatedButton(  
      onPressed: () {},  
      child: Text('Click Me'),  
    ),  
  ],  
)
```

Commands

Flutter (DART)

Category	Command / Widget	Purpose
Basic Structure	import 'package:flutter/material.dart';	Imports Flutter Material UI library
Basic Structure	void main()	Entry point of Flutter application
Basic Structure	runApp()	Starts the Flutter application
Basic Structure	StatelessWidget	Widget with no state (static UI)
Basic Structure	build(BuildContext context)	Builds UI on the screen
App Structure	MaterialApp()	Root widget of the Flutter app
App Structure	title	Sets application title
App Structure	theme	Defines overall app theme
App Structure	home	Sets first screen of app
App Structure	Scaffold()	Provides basic screen layout
App Structure	AppBar()	Top application bar
App Structure	body	Main content area
Text & Display	Text()	Displays text on screen
Text & Display	TextStyle()	Styles the text
Text & Display	fontSize	Sets text size
Text & Display	fontWeight.bold	Makes text bold
Text & Display	FontStyle.italic	Makes text italic
Text & Display	color	Sets text color
Alignment	Center()	Centers widget on screen
Alignment	Align()	Aligns widget to specific position
Alignment	Alignment.topLeft	Top-left alignment
Alignment	Alignment.topRight	Top-right alignment
Alignment	Alignment.bottomLeft	Bottom-left alignment
Alignment	Alignment.bottomRight	Bottom-right alignment
Layout	Row()	Arranges widgets horizontally
Layout	Column()	Arranges widgets vertically
Layout	mainAxisAlignment.center	Aligns widgets at center
Colors & Theme	Colors.red / blue / green	Predefined colors
Colors & Theme	backgroundColor	Sets background color
Colors & Theme	ThemeData()	Defines theme data
Colors & Theme	ThemeData.dark()	Dark theme
Spacing	Padding()	Adds inner spacing

Spacing	SizedBox()	Adds fixed space or size
Container	Container()	Box model widget
Container	BoxDecoration()	Decoration styling
Interaction	ElevatedButton()	Button widget
Interaction	onPressed	Button click action
Scrolling	ListView()	Scrollable content