Chapter 3

Basic Widgets

Basic Widgets

Everything in Flutter is a widget.

But how do you know which widget to use when?

Three categories of basic widgets

- Structure and navigation.
- Displaying information.
- Positioning widgets.

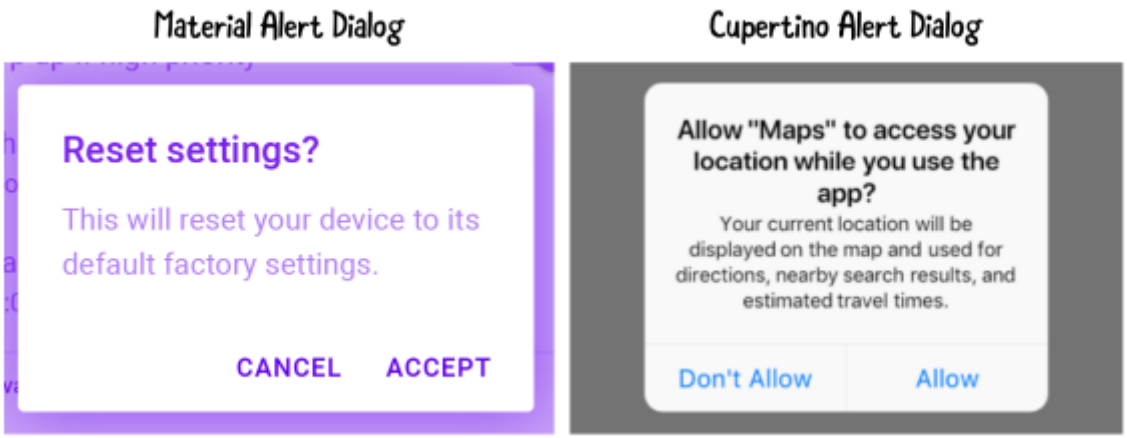
First app in flutter:

Let's discuss the demo app.

- **main**() is the entry point for the code when the app launches.
- Everything in Flutter starts with a widget.
- **runApp**() tells Flutter which is the top-level widget for the app and takes in the root widget.
- Every stateless widget must override the **build**() method.
- The **build**() function return a MaterialApp widget.

MaterialApp Widget:

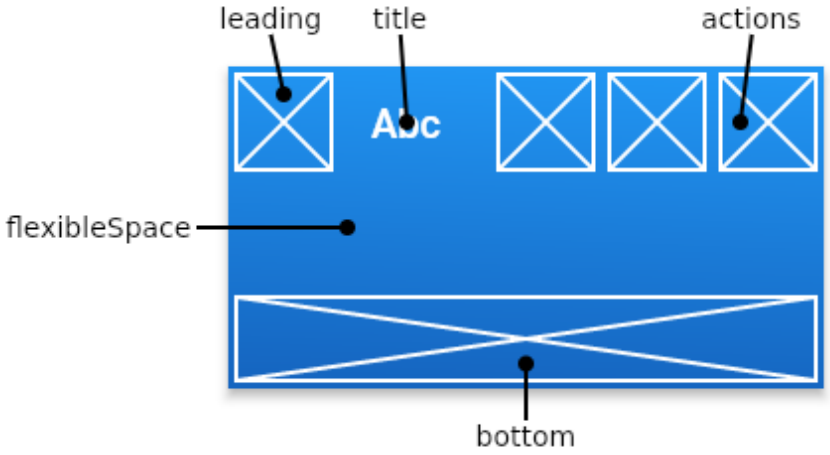
- The MaterialApp widget contains a Scaffold widget, which **defines** the **layout** and **structure** of the app.
- Android uses the **Material Design** system.
- iOS uses the **Cupertino** system.



Scaffold widget properties:

Scaffold has the following properties:

- **appBar**: It displays a horizontal bar which mainly placed at the top of the *Scaffold*.
- **appBar** uses the widget *AppBar* which has its own properties like elevation, title, brightness, etc.



Scaffold widget properties:

- **body**: It will display the main or primary content in the Scaffold. It is below the *appBar* and under the *floatingActionButton*.
- **floatingActionButton**: is a button that is placed at the right bottom corner by default.
- **FloatingActionButton** is an icon button that floats over the content of the screen at a fixed place.

Scaffold widget properties: cont.

- **drawer**: *drawer* is a slider menu or a panel which is displayed at the side of the Scaffold.
- The user has to swipe left to right or right to left according to the action defined to access the drawer menu.
- **bottomNavigationBar**: is like a menu at the bottom of the Scaffold. We can add multiple icons or texts or both in the bar as items.

BottomNavigationBar class:

- The bottom navigation bar consists of multiple items in the form of text labels , icons, or both, laid out on top of a piece of material.
- It provides quick navigation between the top-level views of an app.
- A bottom navigation bar is usually used in conjunction with a Scaffold, where it is provided as the Scaffold.bottomNavigationBar argument.


BottomNavigationBar Demo
Calls
Camera
Chats

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('BottomNavigationBar Demo'),
    ),  // AppBar
    body: Center(
      child: _pages.elementAt(_selectedIndex),
    ),  // Center
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.call),
          label: 'Calls',
        ),  // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.camera),
          label: 'Camera',
        ),  // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.chat),
          label: 'Chats',
        ),  // BottomNavigationBarItem
      ],  // <BottomNavigationBarItem>[]
      currentIndex: _selectedIndex,
      onTap: _onItemTapped,
    ),  // BottomNavigationBar
  );  // Scaffold
}
}
```

Types of widgets:

- **Display widgets:** handle what the user sees onscreen.
- Examples of display widgets include: Text, Image Button.
- **Layout widgets:** help with the arrangement of widgets.
- Examples of layout widgets include: Container, Padding, Stack, Column, SizedBox, Row.

Display widgets: Text

- The Text widget displays a string of text with single style.
- The string might break across multiple lines or might all be displayed on the same line depending on the layout constraints.

```
Text(
  'Hello, $_name! How are you?',
  textAlign: TextAlign.center,
  overflow: TextOverflow.ellipsis,
  style: const TextStyle(fontWeight: FontWeight.bold),
)
```

Display widgets: Image

- A widget that displays an image.

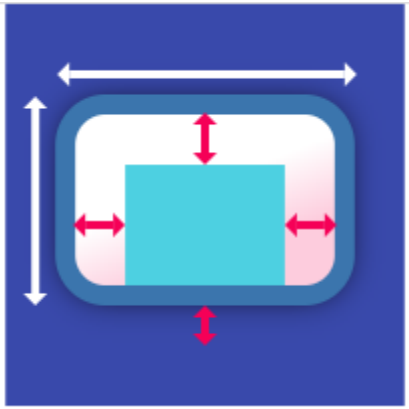
Several constructors are provided for the various ways that an image can be specified:

1. new Image, for obtaining an image from an ImageProvider.
2. new Image.asset, for obtaining an image from an AssetBundle using a key.
3. new Image.network, for obtaining an image from a URL.
4. new Image.file, for obtaining an image from a File.
5. new Image.memory, for obtaining an image from a Uint8List.

Layout widgets: Container

- A convenience widget that combines common painting, positioning, and sizing widgets.
- A container first surrounds the child with padding and then applies additional constraints to the padded extent.

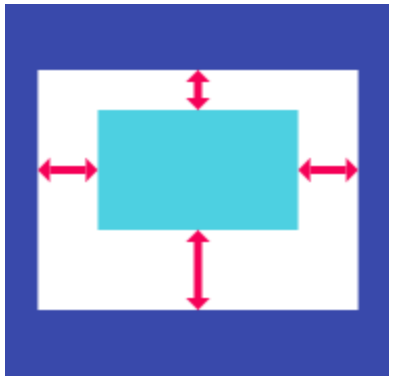
```
Center(
  child: Container(
    margin: const EdgeInsets.all(10.0),
    color: Colors.amber[600],
    width: 48.0,
    height: 48.0,
  ),
)
```



Layout widgets: Padding

- A widget that insets its child by the given padding.

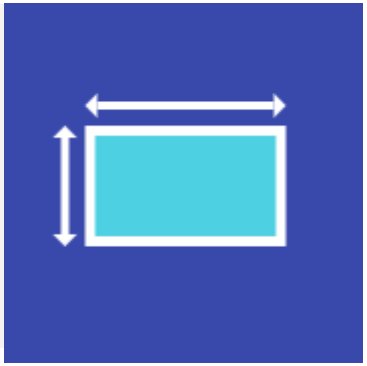
```
const Card(
  child: Padding(
    padding: EdgeInsets.all(16.0),
    child: Text('Hello World!'),
  ),
)
```



Layout widgets: SizedBox

- A box with a specified size.
- If given a child, this widget forces it to have a specific width and/or height.
- These values will be ignored if this widget's parent does not permit them.

```
const SizedBox(
  width: 200.0,
  height: 300.0,
  child: Card(child: Text('Hello World!')),
)
```



Layout widgets: Column

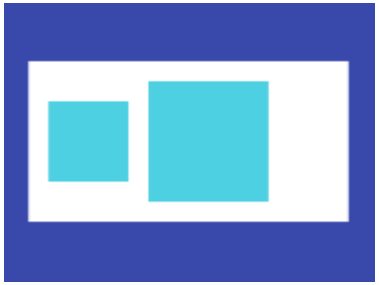
- A widget that displays its children in a vertical array.
- The Column widget does not scroll .

```
Column(
  children: const <Widget>[
    Text('Deliver features faster'),
    Text('Craft beautiful UIs'),
    Expanded(
      child: FittedBox(
        fit: BoxFit.contain, // otherwise the logo will be tiny
        child: FlutterLogo(),
      ),
    ),
  ],
)
```

Layout widgets: Row

- A widget that displays its children in a horizontal array.
- The Row widget does not scroll .

```
Row(
  children: const <Widget>[
    Expanded(
      child: Text('Deliver features faster', textAlign: TextAlign.center),
    ),
    Expanded(
      child: Text('Craft beautiful UIs', textAlign: TextAlign.center),
    ),
    Expanded(
      child: FittedBox(
        fit: BoxFit.contain, // otherwise the logo will be tiny
        child: FlutterLogo(),
      ),
    ),
  ],
)
```



Information displays: Card

- A Material Design card. A card has slightly rounded corners and a shadow.
- A card is a sheet of Material used to represent some related information, for example an album, a geographical location, a meal, contact details, etc.

Information displays: Card

```
@override
Widget build(BuildContext context) {
  return Center(
    child: Card(
      child: InkWell(
        splashColor: Colors.blue.withAlpha(30),
        onTap: () {
          debugPrint('Card tapped.');
        },
        child: const SizedBox(
          width: 300,
          height: 100,
          child: Text('A card that can be tapped'),
        ),
      ),
    ),
  );
}
```