08. Useful Widget

- Locators
    - Align
    - Center
    - Expanded
    - Padding margin: EdgeInsets
    - SizedBox
- Events
    - Gesture 1 (GestureDetector and InkWell)
    - gesture 2 (Example)
- Other Widgets
    - Card
    - CircleAvatar
    - Drawer
    - Progress
    - SnackBar

- In this section, we discuss useful widgets that we can use to build Flutter applications.
- There are many more widgets to use, but these widgets are a good starting point for projects.

Locators

- Align
- Center
- Expanded
- Padding margin
- Sizedbox

Align

```
Align(
  // The options are:
  //  bottomLeft/Center/Right,
  //  centerLeft/Right, center,
  //  topLeft/Right, top.
  alignment: Alignment.bottomRight,
  child: Container(
    alignment: Alignment.center,
  ),
);
```

- The `Align` widget aligns its child.

Center

```
body: Center( // Centering the container
  child: Container(
    alignment: Alignment.center,
    child: const Text(
      textAlign: TextAlign.center,
      style: TextStyle(
```

- The `Center` widget centers its child within its parent widget or the available space on the screen.

Other ways to align

```
child: Container( // Centering the text in the container
  alignment: Alignment.center,
  child: const Text(  // In the Text, the string is in the center
    textAlign: TextAlign.center,
```

- The `alignment` property of the `Container` determines where its child widget (here, the `Text`) is placed.
- The `textAlign` property of the `Text` widget specifies how the text should be aligned horizontally within the `Text` widget’s own space.

Expanded

```
children: <Widget>[
  Expanded(flex: 2, child: Container(...),),
  Expanded(flex: 1, child: Container(...),),
  Expanded(flex: 1, child: Container(...),),
],
```

- `Expanded` widget controls the expanded ratio: 2:1:1 in this example.

Padding margin: EdgeInsets

```
child: Container(
  padding: const EdgeInsets.all(8.0),
  margin: const EdgeInsets.all(8.0),
```

- We can use padding and margin to give some space: we use `EdgeInsets` to make the space.

SizedBox

```
body: SizedBox(
  width: 100, height: 100,
  child: Container(
    color: Colors.red,
  ),
),
```

- The `Sizedbox` widget can give size properties to **any** widgets without a size-related property.

Events

- gesture 1 (GestureDetector and InkWell)
- gesture 2 (Example)

Gesture 1 (GestureDetector and InkWell)

```
GestureDetector(
  onTap: () { ... }
  child: Text('Click Me!!'),
),
InkWell(
  onTap: () { ... }
  child: Text('Click Me!!'),
),
```

- We can add a gesture action to the widget that does not have the onTap

gesture 2 (Example)

- We can use a variable (`_lightIsOn`) and the `setState` function to make a Text into a button to on/off the light on or off.

```
child: Icon(
  Icons.lightbulb_outline,
  color: _lightIsOn ? Colors.yellow.shade600 : Colors.black,
),
GestureDetector(onTap: () {setState(() {_lightIsOn = !_lightIsOn;});},
child: Container(
  child: Text(_lightIsOn ? 'TURN LIGHT OFF' : 'TURN LIGHT ON'),
),
```

Other Widgets

- Card
- CircleAvatar
- Drawer
- Progress
- SnackBar

Card

```
child: Card(
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(16.0),
  ),
  elevation: 4.0, // shadow depth
  child: Container(
    width: 200, height: 200,
    child: Center(child: Text("Hello")),
  ),
```

- The card widget is used to display information in an organized way.

CircleAvatar

```
child: GestureDetector(
  onTap: () {setState(() {...});},
  child:  CircleAvatar(
    backgroundImage: NetworkImage('...'),
```

- The CircleAvatar displays information in a circle.
- In this example, we use the `NetworkImage` widget to get an image from the Internet.

Drawer

```
drawer: Drawer(
  child: ListView(
    children: [
      ListTile(),
      ListTile(),
    ],
  ),
),
```

- The Drawer widget makes a drawer on the top left side.

Progress

```
CircularProgressIndicator(),
LinearProgressIndicator(),
```

- The Progress widget shows a circular or linear bar to display that the work is in progress.

SnackBar

```
ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(content: Text('Processing Data')),
);
```

- The SnackBar widget uses the current `context` and `showSnackBar` functions to display information at the bottom.