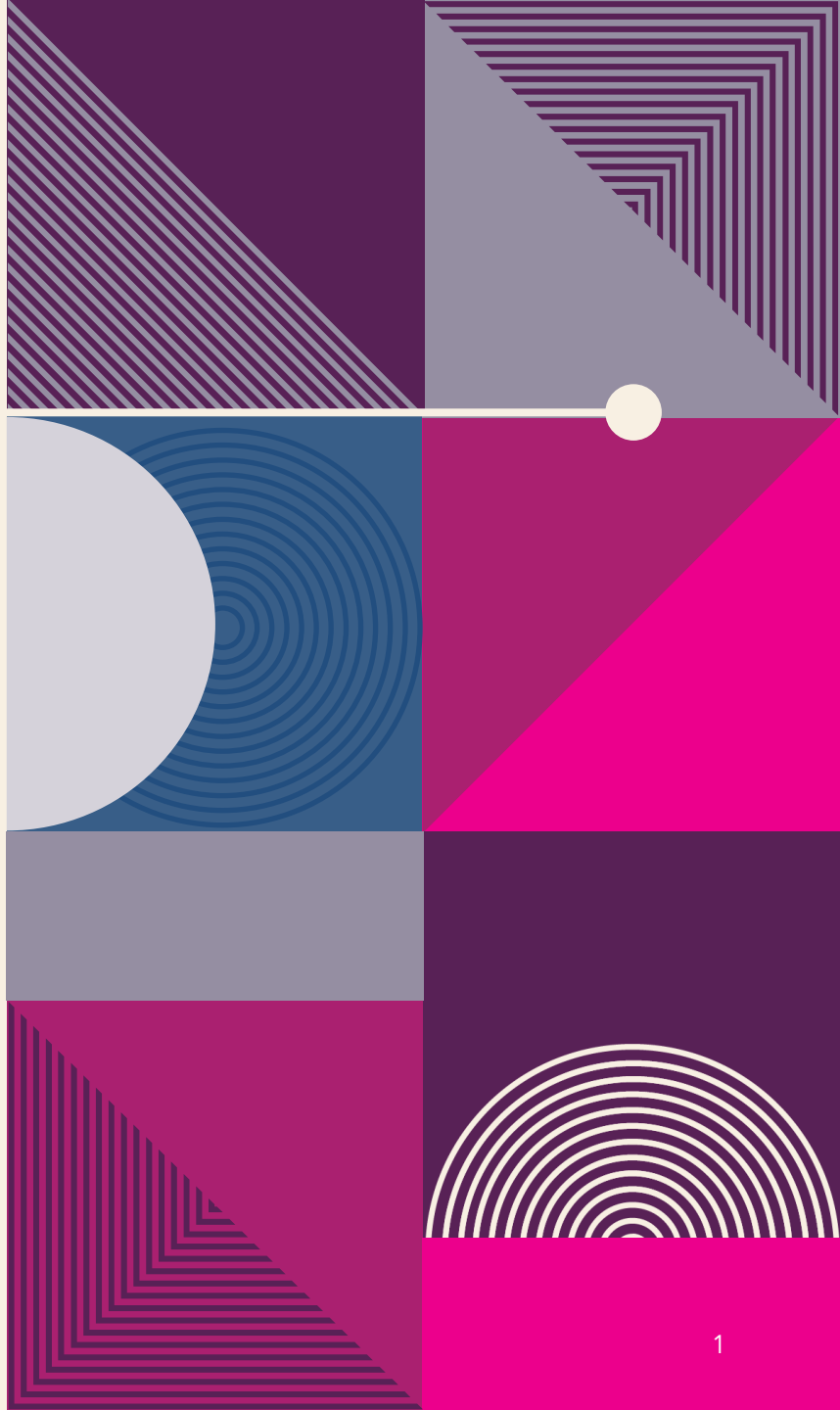
CHECKING ORIENTATION



CHECKING ORIENTATION

There are two ways to figure out orientation,

1. MediaQuery.of(context).orientation.
2. OrientationBuilder.

Creating the Orientation App

Create a new Flutter project and name it **flutter_orientation**. .

1. **Open** the **home.dart** file and

- **Add** to the **body** a **SafeArea** with **SingleChildScrollView** as a child.
- **Add Padding** as a child of the **SingleChildScrollView**.
- **Add** a **Column** as a child of the **Padding**.
- In the **Column** children property, **add** the widget class called **OrientationLayoutIconsWidget**(), which you will create next.
- Make sure you add the **const** keyword before the widget class name to take advantage of caching to improve performance.

```
body: SafeArea(
  child: SingleChildScrollView(
    child: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: <Widget>[
          const OrientationLayoutIconsWidget(),
],  ),
    ),
  ),
),
```

2. Add the **OrientationLayoutIconsWidget**() widget **class** after **class Home extends StatelessWidget {...}**.

```
class OrientationLayoutIconsWidget extends StatelessWidget {
  const OrientationLayoutIconsWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Orientation _orientation =
MediaQuery.of(context).orientation;
    return Container();
  }
}
```

3. Based on the current **Orientation**, you return a different layout of **Icon** widgets.

- **Use** a **ternary operator** to check whether **Orientation** is **portrait**, and if so, return a single **Row** icon.
- If **Orientation** is **landscape**, return a **Row** of **two Icon** widgets. **Replace** the current return **Container**() with the following code:

```
class OrientationLayoutIconsWidget extends StatelessWidget {
  const OrientationLayoutIconsWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Orientation _orientation = MediaQuery.of(context).orientation;
        return _orientation == Orientation.portrait
        ? Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Icon(
          Icons.school,
          size: 48.0,
        ),
      ],
    )
        : Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Icon(
          Icons.school,
          size: 48.0,
        ),
        Icon(
          Icons.brush,
          size: 48.0,     ),        ],
    );
  }
}
```

4. After **OrientationLayoutIconsWidget**(), add a **Divider** widget and the **OrientationLayoutWidget**() widget **class** to **create**.

```
body: SafeArea(
  child: SingleChildScrollView(
    child: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: <Widget>[
          const
OrientationLayoutIconsWidget(),
          Divider(),
          const OrientationLayoutWidget(),
        ],
      ),
    ),
  ),
),
```

5. Create OrientationLayoutWidget() widget class.

```
class OrientationLayoutWidget extends StatelessWidget {
  const OrientationLayoutWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Orientation _orientation = MediaQuery.of(context).orientation;

    return _orientation == Orientation.portrait
        ? Container(
      alignment: Alignment.center,
      color: Colors.yellow,
      height: 100.0,
      width: 100.0,
      child: Text('Portrait'),
    )
        : Container(
      alignment: Alignment.center,
      color: Colors.lightGreen,
      height: 100.0,
      width: 200.0,
      child: Text('Landscape'),    );
  }
}
```

6. After **OrientationLayoutWidget**(), add a **Divider** widget and the **GridViewWidget**() widget **class** that you will **create**.

```
body: SafeArea(
  child: SingleChildScrollView(
    child: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: <Widget>[
          const OrientationLayoutIconsWidget(),
          Divider(),
          const OrientationLayoutWidget(),
          Divider(),
          const GridViewWidget(),
        ],
      ),
    ),
  ),
),
```

7. Create GridViewWidget() widget class.

```
class GridViewWidget extends StatelessWidget {
  const GridViewWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Orientation _orientation = MediaQuery.of(context).orientation;

    return GridView.count(
      shrinkWrap: true,
      physics: NeverScrollableScrollPhysics(),
      crossAxisCount: _orientation == Orientation.portrait ? 2 : 4,
      childAspectRatio: 5.0,
      children: List.generate(8, (int index) {
        return Text("Grid $index", textAlign: TextAlign.center,);
      }),
    );
  }
}
```

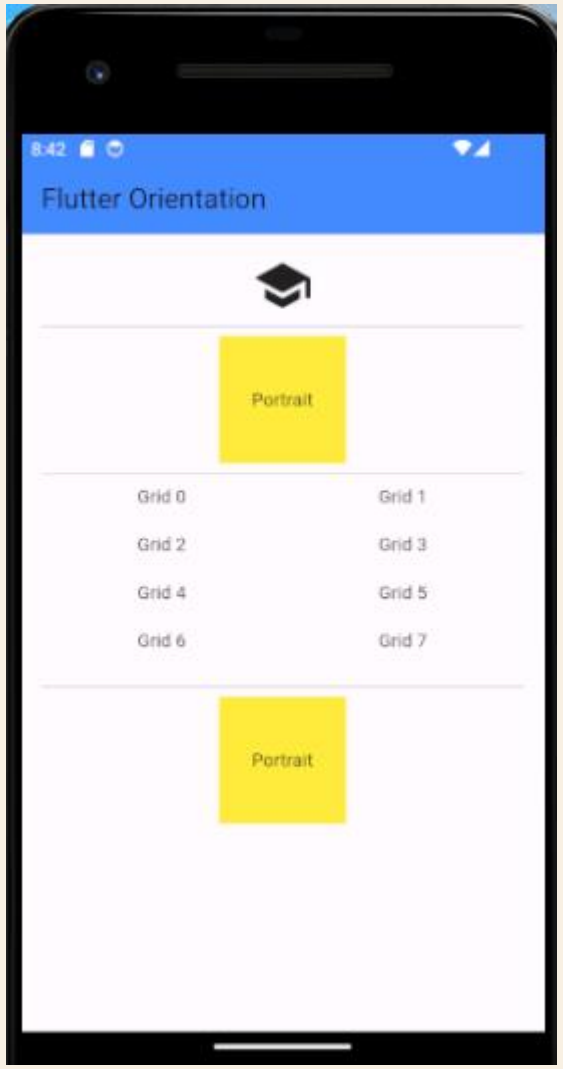
8. After **GridViewWidget**(), add a **Divider** widget and the **OrientationBuilderWidget**() widget **class** that you will create.

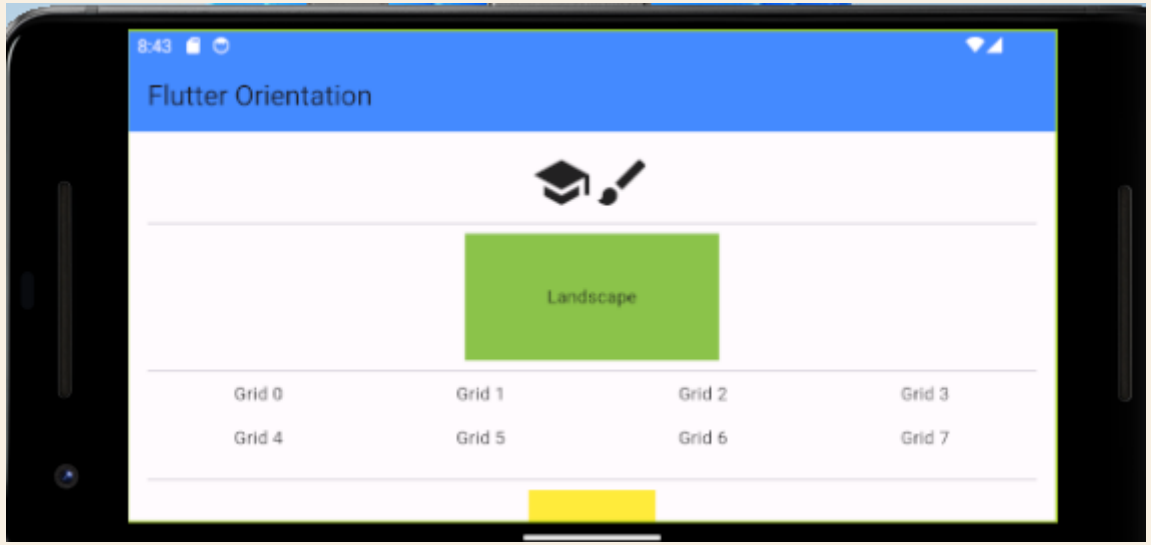
```
body: SafeArea(
  child: SingleChildScrollView(
    child: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: <Widget>[
          const OrientationLayoutIconsWidget(),
          Divider(),
          const OrientationLayoutWidget(),
          Divider(),
          const GridViewWidget(),
          Divider(),
          const OrientationBuilderWidget(),
        ],
      ),
    ),
  ),
),
```

9. Create OrientationBuilderWidget() widget class.

```
// OrientationBuilder as a child does not give correct Orientation. i.e Child of Column...
// OrientationBuilder as a parent gives correct Orientation
class OrientationBuilderWidget extends StatelessWidget {
  const OrientationBuilderWidget({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return OrientationBuilder(
      builder: (BuildContext context, Orientation orientation) {
        return orientation == Orientation.portrait
            ? Container(
          alignment: Alignment.center,
          color: Colors.yellow,
          height: 100.0,
          width: 100.0,
          child: Text('Portrait'),
        )
            : Container(
          alignment: Alignment.center,
          color: Colors.lightGreen,
          height: 100.0,
          width: 200.0,
          child: Text('Landscape'),
        );
      },
    );
  }
}
```


8:42
Flutter Orientation
Portrait
Grid 0
Grid 1
Grid 2
Grid 3
Grid 4
Grid 5
Grid 6
Grid 7
Portrait


8:43
Flutter Orientation
Landscape
Grid 0
Grid 1
Grid 2
Grid 3
Grid 4
Grid 5
Grid 6
Grid 7