

```

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Using Common Widgets',
      theme: ThemeData(
        primarySwatch: Colors.lightGreen, // Set primary theme color
      ),
      home: Home(),
    );
  }
}

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      // ----- APPBAR SECTION -----
      appBar: AppBar(
        // Leading widget (menu button at start of AppBar)
        leading: IconButton(
          icon: Icon(Icons.menu),
          onPressed: () {},
        ),
        // Title of AppBar
        title: Text("Home"),
        // Action buttons on right side
        actions: <Widget>[
          IconButton(
            icon: Icon(Icons.search),
            onPressed: () {},
          ),
          IconButton(
            icon: Icon(Icons.more_vert),
            onPressed: () {},
          ),
        ],
        // Background widget behind the title
        flexibleSpace: SafeArea(
          child: Icon(
            Icons.photo_camera,
            size: 75.0,
            color: Colors.white70,
          ),
        ),
      ),
    );
  }
}

```

```

    ),
  ),
  // Custom bottom bar inside AppBar
  bottom: PreferredSize(
    child: Container(
      color: Colors.lightGreen.shade100,
      height: 75.0,
      child: Center(
        child: Text('Bottom Section'),
      ),
    ),
  ),
  preferredSize: Size.fromHeight(75.0),
),
),

// ----- BODY SECTION -----
body: Padding(
  padding: EdgeInsets.all(16.0),
  child: SafeArea(
    child: SingleChildScrollView(
      child: Column(
        children: <Widget>[
          const ContainerWithBoxDecorationWidget(),
          Divider(),
          const ColumnWidget(),
          Divider(),
          const RowWidget(),
          Divider(),
          const ColumnAndRowNestingWidget(),
          Divider(),
          const ButtonsWidget(),
          Divider(),
          const ButtonBarWidget(),
          Divider(),
          const ImagesAndIconWidget(),
          Divider(),
          const BoxDecorationWidget(),
          Divider(),
          const InputDecoratorsWidget(),
          Divider(),
          const OrientationLayoutIconsWidget(),
          Divider(),
          const OrientationLayoutWidget(),
          Divider(),
          const GridViewWidget(),
          Divider(),
          const OrientationBuilderWidget(),
        ],
      ),
    ),
  ),
),

```

```

    ),
    ),
    ),
  ),

  // ----- FLOATING ACTION BUTTON -----
  floatingActionButtonLocation: FloatingActionButtonLocation.endDocked,
  floatingActionButton: FloatingActionButton(
    onPressed: () {},
    child: Icon(Icons.play_arrow),
    backgroundColor: Colors.lightGreen.shade100,
  ),
  bottomNavigationBar: BottomAppBar(
    notchMargin: 6.0,
    color: Colors.lightGreen.shade100,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Icon(Icons.pause),
        Icon(Icons.stop),
        Icon(Icons.access_time),
        Padding(padding: EdgeInsets.all(32.0)),
      ],
    ),
  ),
);
}
}

```

```

// ----- WIDGET CLASSES -----

```

```

// 1. Container with BoxDecoration
class ContainerWithBoxDecorationWidget extends StatelessWidget {
  const ContainerWithBoxDecorationWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      height: 100.0,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.only(
          bottomLeft: Radius.circular(100.0),
          bottomRight: Radius.circular(10.0),
        ),
        gradient: LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [Colors.white, Colors.lightGreen.shade500],

```

```

    ),
    boxShadow: [
      BoxShadow(
        color: Colors.grey,
        blurRadius: 10.0,
        offset: Offset(0.0, 10.0),
      ),
    ],
  ),
  child: Center(
    child: RichText(
      text: TextSpan(
        text: 'Flutter World',
        style: TextStyle(
          fontSize: 24.0,
          color: Colors.deepPurple,
          decoration: TextDecoration.underline,
          fontStyle: FontStyle.italic,
        ),
      ),
      children: <TextSpan>[
        TextSpan(text: ' for'),
        TextSpan(
          text: ' Mobile',
          style: TextStyle(
            color: Colors.deepOrange,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  ),
);
}
}

```

// 2. Column Example

```

class ColumnWidget extends StatelessWidget {
  const ColumnWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Text('Column 1'),
        Divider(),
        Text('Column 2'),
        Divider(),
      ],
    );
  }
}

```

```

        Text('Column 3'),
      ],
    );
  }
}

```

// 3. Row Example

```

class RowWidget extends StatelessWidget {
  const RowWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Text('Row 1'),
        Text('Row 2'),
        Text('Row 3'),
      ],
    );
  }
}

```

// 4. Column with Row Nesting

```

class ColumnAndRowNestingWidget extends StatelessWidget {
  const ColumnAndRowNestingWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Text('Columns and Row Nesting 1'),
        Text('Columns and Row Nesting 2'),
        Text('Columns and Row Nesting 3'),
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: <Widget>[
            Text('Row Nesting 1'),
            Text('Row Nesting 2'),
            Text('Row Nesting 3'),
          ],
        ),
      ],
    );
  }
}

```

// 5. Different Buttons

```

class ButtonsWidget extends StatelessWidget {
  const ButtonsWidget({Key? key}) : super(key: key);

```

```

@override
Widget build(BuildContext context) {
  return Column(
    children: <Widget>[
      // ----- TEXTBUTTONS (formerly FlatButton) -----
      Row(
        children: <Widget>[
          FlatButton(
            onPressed: () {},
            child: const Text('Flag'),
          ),
          FlatButton(
            onPressed: () {},
            style: FlatButton.styleFrom(
              backgroundColor: Colors.lightGreen,
              foregroundColor: Colors.white, // replaces textColor
            ),
            child: const Icon(Icons.flag),
          ),
        ],
      ),

      const Divider(),

      // ----- ELEVATEDBUTTONS (formerly RaisedButton) -----
      Row(
        children: <Widget>[
          ElevatedButton(
            onPressed: () {},
            child: const Text('Save'),
          ),
          ElevatedButton(
            onPressed: () {},
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.lightGreen,
              foregroundColor: Colors.white,
            ),
            child: const Icon(Icons.save),
          ),
        ],
      ),

      const Divider(),

      // ----- ICONBUTTONS (unchanged) -----
      Row(
        children: <Widget>[

```

```

        IconButton(onPressed: () {}, icon: const Icon(Icons.flight)),
        IconButton(
          onPressed: () {},
          icon: const Icon(Icons.flight),
          iconSize: 42.0,
          color: Colors.lightGreen,
          tooltip: 'Flight',
        ),
      ],
    ),
  ],
);
}
}

```

// 6. ButtonBar

```

class ButtonBarWidget extends StatelessWidget {
  const ButtonBarWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white70,
      child: ButtonBar(
        alignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          IconButton(icon: Icon(Icons.map), onPressed: () {}),
          IconButton(icon: Icon(Icons.airport_shuttle), onPressed: () {}),
          IconButton(icon: Icon(Icons.brush), onPressed: () {}),
        ],
      ),
    );
  }
}

```

// 7. Images and Icons

```

class ImagesAndIconWidget extends StatelessWidget {
  const ImagesAndIconWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Image.asset("assets/images/img.png",
          fit: BoxFit.cover, width: MediaQuery.of(context).size.width / 3),
        Image.network(
          'https://flutter.io/images/catalog-widget-placeholder.png',
          width: MediaQuery.of(context).size.width / 3,

```

```

    ),
    Icon(Icons.brush, color: Colors.lightBlue, size: 48.0),
  ],
);
}
}

```

// 8. BoxDecoration Example

```

class BoxDecorationWidget extends StatelessWidget {
  const BoxDecorationWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 100.0,
      width: 100.0,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20.0),
        color: Colors.orange,
        boxShadow: [
          BoxShadow(
            color: Colors.grey, blurRadius: 10.0, offset: Offset(0.0, 10.0))
        ],
      ),
    );
  }
}

```

// 9. InputDecoration Example

```

class InputDecorationWidget extends StatelessWidget {
  const InputDecorationWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Column(children: <Widget>[
      TextField(
        decoration: InputDecoration(
          labelText: "Notes",
          border: OutlineInputBorder(),
        ),
      ),
      Divider(),
      TextFormField(
        decoration: InputDecoration(labelText: "Enter your notes"),
      ),
    ]);
  }
}

```

// 10. Orientation Example with MediaQuery


```

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: [Icon(Icons.school, size: 48.0)],
      )
      : Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Icon(Icons.school, size: 48.0),
          Icon(Icons.brush, size: 48.0),
        ],
      );
  }
}

```

// 11. Orientation Example with Container

```

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(
        color: Colors.yellow,
        height: 100.0,
        width: 100.0,
        child: Center(child: Text('Portrait')),
      )
      : Container(
        color: Colors.green,
        height: 100.0,
        width: 200.0,
        child: Center(child: Text('Landscape')),
      );
  }
}

```

// 12. GridView Example

```

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, (index) {
    return Text("Grid $index", textAlign: TextAlign.center);
  }
),
);
}
}

```

// 13. OrientationBuilder Example

```

class OrientationBuilderWidget extends StatelessWidget {
  const OrientationBuilderWidget({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return OrientationBuilder(
      builder: (context, orientation) {
        return orientation == Orientation.portrait
          ? Container(
              color: Colors.yellow,
              height: 100.0,
              width: 100.0,
              child: Center(child: Text('Portrait')),
            )
          : Container(
              color: Colors.green,
              height: 100.0,
              width: 200.0,
              child: Center(child: Text('Landscape')),
            );
      },
    );
  }
}

```