Buttons and their features

List of replaced classes:

- FlatButton → TextButton
- RaisedButton → ElevatedButton
- OutlineButton → OutlinedButton
- ButtonTheme → TextButtonTheme, ElevatedButtonTheme, OutlineButtonTheme

Constructors:

Syntax:

```
RaisedButton({Key key,
@required VoidCallback onPressed,
VoidCallback onLongPress,
ValueChanged<bool> onHighlightChanged,
MouseCursor mouseCursor,
ButtonTextTheme textTheme,
Color textColor,
Color disabledTextColor,
Color color,
Color disabledColor,
Color focusColor,
Color hoverColor,
Color highlightColor,
Color splashColor,
Brightness colorBrightness,
double elevation,
double focusElevation,
double hoverElevation,
double highlightElevation,
double disabledElevation,
EdgeInsetsGeometry padding,
VisualDensity visualDensity,
ShapeBorder shape,
Clip clipBehavior: Clip.none,
FocusNode focusNode,
bool autofocus: false,
MaterialTapTargetSize materialTapTargetSize,
```

```
Duration animationDuration,
Widget child})
```

Properties:

- **animationDuration:** This parameter takes in *Duration Class* as the object to determine for how long the animation will play.
- **autofocus:** This property determines whether the button will be selected on the initial focus or not by taking in a *boolean* as the parameter.
- **child:** the widget to be displayed.
- **clipBehavior:** This parameter decides whether the content inside the button will be clipped or not.
- **color:** the color of the button.
- **colorBrightness:** This property decides the theme brightness to be used for this the RaisedButton by taking in Brightness class as the object.
- **disabledColor:** the color of the button when disabled.
- **disabledElevation:** This property sets the elevation height of the button when it is disabled. It takes a *double* value as the object.
- **disabledTextColor:** This property sets the color of the text inside the button when the RaisedButton is disabled. It takes *Color* class as the object.
- **elevation:** This property sets the elevated location of the button on the z-axis by taking in a *Double* value as the object.
- **enabled:** This parameter determined whether the button is enabled or disabled, by taking in a *boolean* value as the object.
- **enableFeedback:** This property also holds a *boolean* value as the object. It controls whether there will be and sound or vibration when the button is clicked.
- **focusColor:** This property controls the color of the button at the time of input focus, but holding the *Color* class as the object.
- **focusElevation:** It controls the location of RaisedButton at the z-axis at the time of input focus. It takes in a *double* value as the object.
- **focusNode:** This property takes in *FocusNode* class as the object. It provides an additional focus node to the RaisedButton.

- **height:** This determines the height of the button by taking in a *double* value.
- **highlightColor:** This parameter sets the highlight color to the button by taking is a *Color* class as the object.
- **highlightElevation:** This property controls the elevation height of the RaisedButton when it is enabled and pressed. It also takes a double value.
- **hoverColor:** This determines the color of the RaisedButtom at the time of hover, by employing the *Color* class.
- **hoverElevation:** This parameter sets the elevation height on the z-axis for the button went it is in hover state.
- **materialTapTargetSize**the color of the button.
- **minWidth:** This determines the minimum width the RaisedButton can take. It holds a *double* value as the object.
- **mouseCursor:** This property controls the type on the cursor when the mouse hovers over the button. It employs the *MouseCursor* class as the object.
- **onLongPress:** The callback function when the button is long pressed.
- **onPressed:** The callback function when the button is tapped.
- **padding:** padding inside the button.
- **shape:** The shape of the raised button.
- **splashColor:** splash color of the button.
- **textColor:** The color of the text.
- **textTheme:** This parameter defines the default theme for the RaisedButton. It holds *ButtonTextTheme enum* to do so.
- **visualDensity:** This defines the layout compactness of the button by taking in the *VisualDensity* class as the object.

Implementation:

The *main.dart* file:

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


void main() {
```

```
runApp(const MyApp());
}

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

// This widget is the root of your application.
@override
Widget build(BuildContext context) {
        return MaterialApp(
        title: 'Raised Button',
        theme: ThemeData(
                primarySwatch: Colors.blue,
        ),
        home: const MyHomePage(),
        debugShowCheckedModeBanner: false,
        );
}
}

class MyHomePage extends StatefulWidget {
const MyHomePage({Key? key}) : super(key: key);

@override
// ignore: library_private_types_in_public_api
```

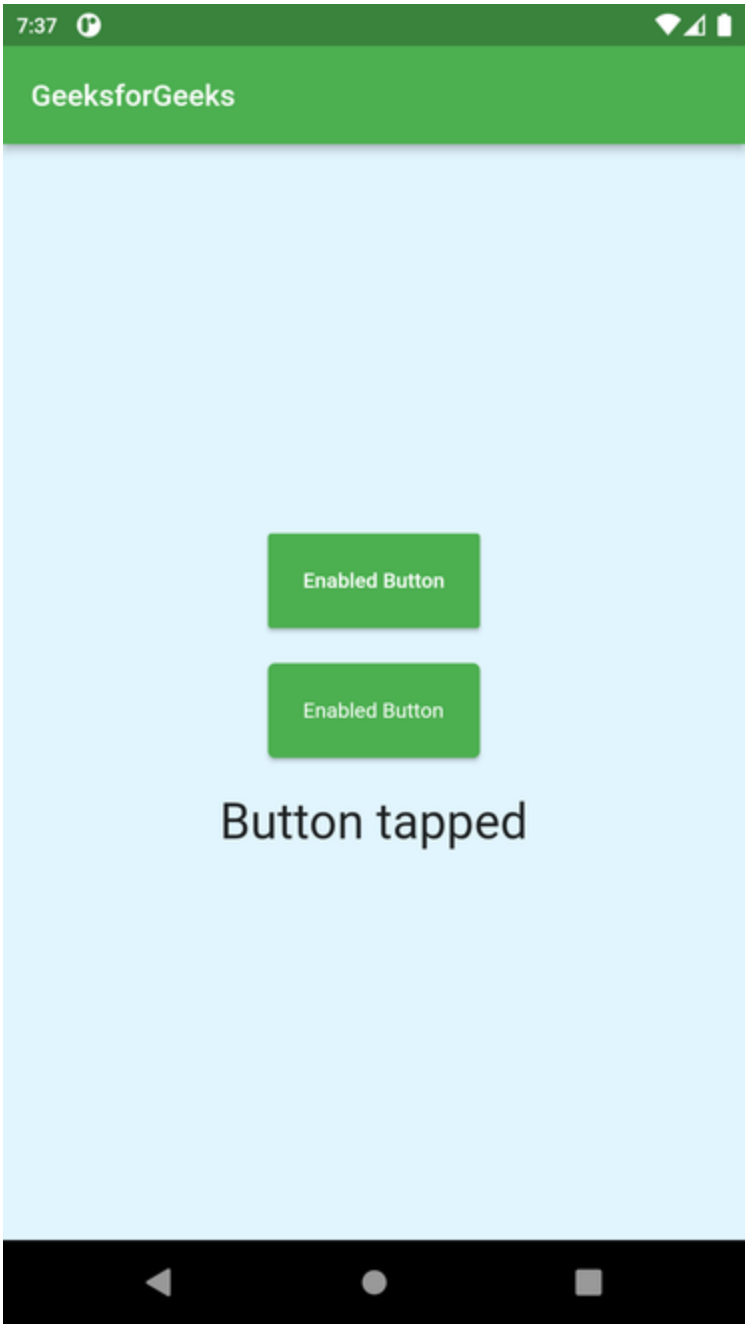
```
_MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
String istapped = '';
@override
Widget build(BuildContext context) {
        return Scaffold(
        appBar: AppBar(
                title: const Text('GeeksforGeeks'),
                backgroundColor: Colors.green,
        ),
        body: Center(
                child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                        // RaisedButton has been deprecated
                        // We can use Elevated button achieve the same results
                        RaisedButton(
                        //         disabledColor: Colors.red,
                        // disabledTextColor: Colors.black,
                        padding: const EdgeInsets.all(20),
                        textColor: Colors.white,
                        color: Colors.green,
                        onPressed: () {
```

```
                    setState(() {
                    istapped = 'Button tapped';
                    });
                },
                child: const Text('Enabled Button'),
                ),
                const SizedBox(
                height: 20,
                ),
                // ElevatedButton
                ElevatedButton(
                    style: ButtonStyle(
                        backgroundColor:
MaterialStateProperty.all(Colors.green),
                        padding:
                            MaterialStateProperty.all(const
EdgeInsets.all(20)),
                        textStyle: MaterialStateProperty.all(
                            const TextStyle(fontSize: 14, color:
Colors.white))),
                    onPressed: () {
                    setState(() {
                        istapped = 'Button tapped';
                    });
                    },
                    child: const Text('Enabled Button')),
```

```
                const SizedBox(height: 20),
                Text(
                istapped,
                textScaleFactor: 2,
                )
            ],
            ),
        ),
        backgroundColor: Colors.lightBlue[50],
        );
}
}
```

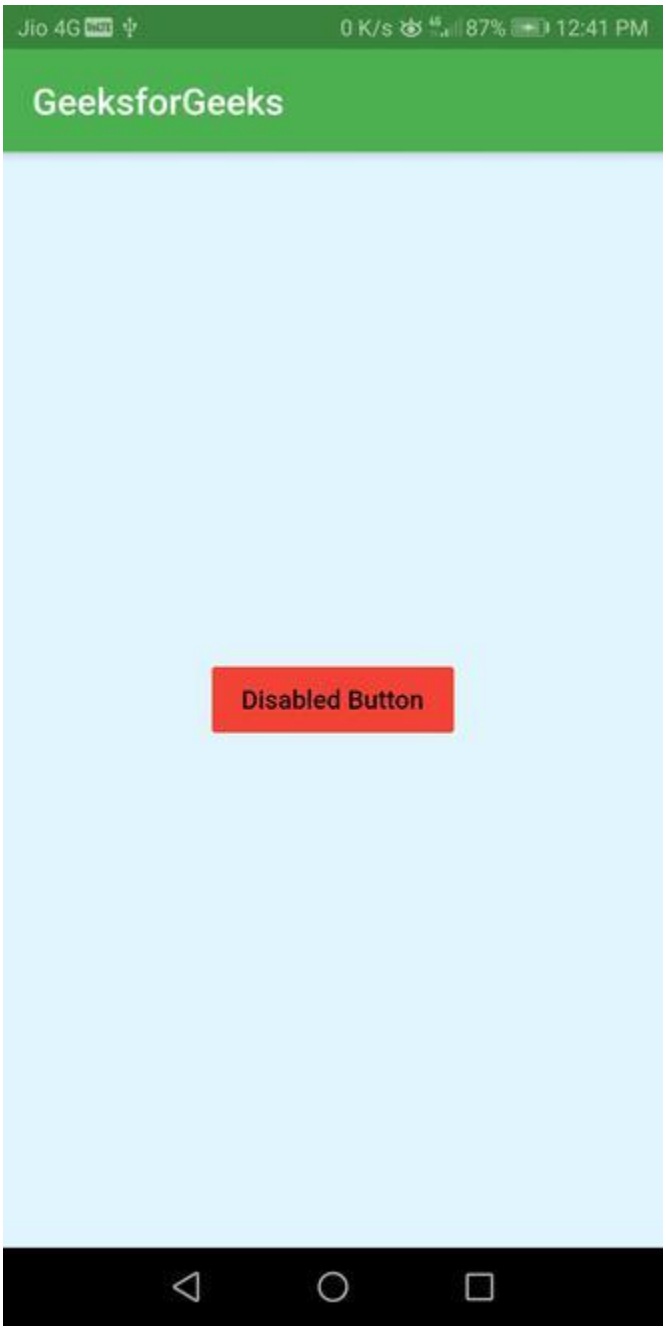
Dart**Output:**



If the properties are defined as below:

```
RaisedButton(
          disabledColor: Colors.red,
          disabledTextColor: Colors.black,
          onPressed: null,
          child: Text('Disabled Button'),
        ),
```

The following design changes can be observed:

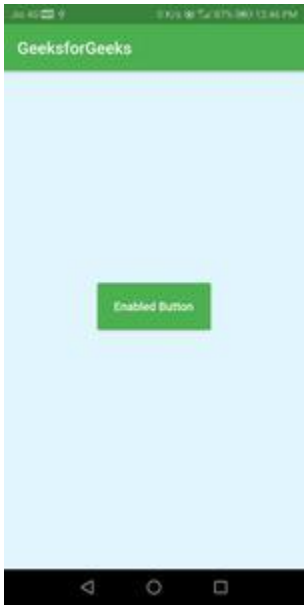


If the properties are defined as below:

```
RaisedButton(
          padding: const EdgeInsets.all(20),
          textColor: Colors.white,
          color: Colors.green,
          onPressed: ()=>null,
          child: Text('Enabled Button'),
```

```
        ),
```

The following design changes can be observed:



If the properties are defined as below:

```
RaisedButton(
              padding: const EdgeInsets.all(20),
              textColor: Colors.white,
              color: Colors.green,
              onPressed: (){
                setState(() {
                  istapped='Button tapped';
                });
              },
              child: Text('Enabled Button'),
            ),
            Text(istapped,textScaleFactor: 2,)
          ),
```

The following design changes can be observed:

Output explanation:

- Create **RaisedButton** and wrap it with the Center widget.
- Give the child of *RaisedButton* as a *Text* widget.
- Perform **onPressed** function when the button is *tapped*.
- Perform optional **onLongPress** function when the button is *long pressed*.