

Flutter RaisedButton – onPressed

Flutter RaisedButton – onPressed

Flutter RaisedButton's onPressed property lets you assign it with a callback function. The application executes this callback function when user presses on the RaisedButton.

In this tutorial, we will learn how to execute a set of statements using callback function for onPressed property of RaisedButton.

Following is a code snippet of how you write the callback for onPressed property.

```
RaisedButton(  
  child: Text(  
    'Login',  
  ),  
  onPressed: () {  
    //statement(s)  
  },  
),
```

You can write any statements in the onPressed callback, from print statements to updating UI components or navigating to a new screen, etc.

Example – Perform Action when User presses Flutter RaisedButton

In this example, we have a RaisedButton that displays a button with `Login` text. So, it appears like a Login button. Also, we have initialized onPressed property of this button with a callback function.

When user presses this button, this callback function is called. The application executes all the statements inside this callback function. In this example application, we have written a print statement, for demonstrating the working of onPressed property. You may replace that with as many statements that emulate a required functionality as per the requirement of your application.

main.dart

```
import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}
```

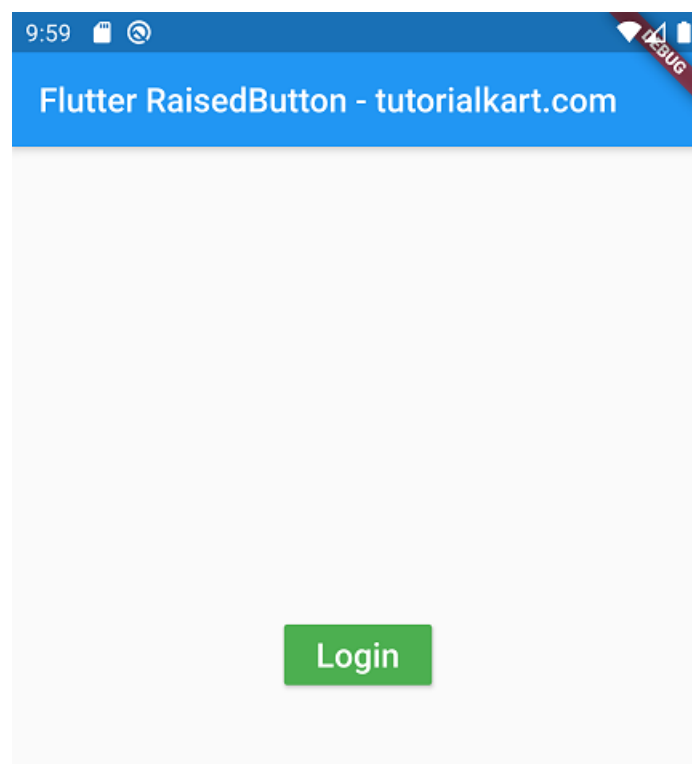
```

}

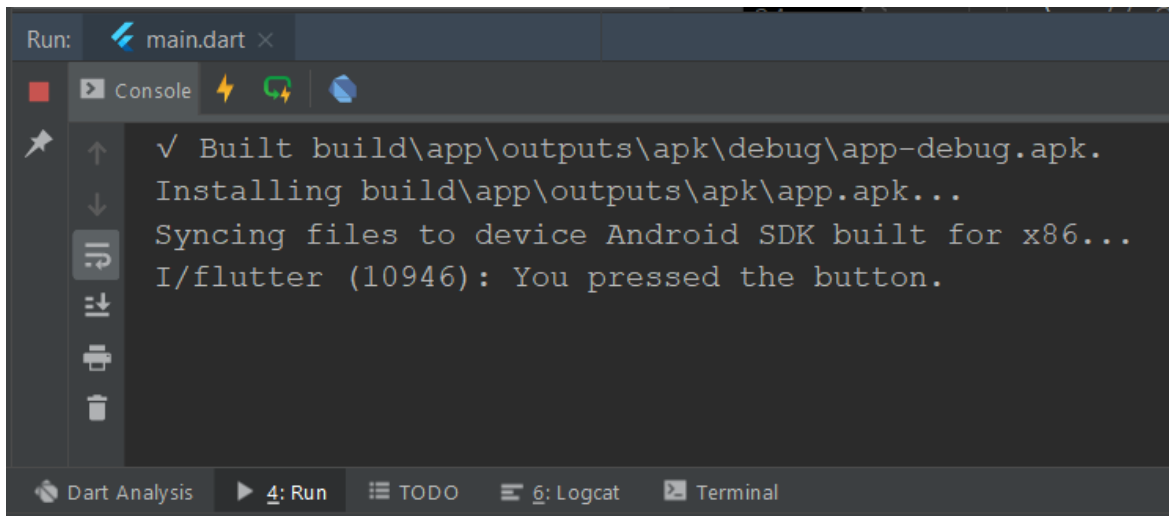
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Flutter RaisedButton - tutorialkart.com'),
        ),
        body: Center(
          child: Container(
            child: RaisedButton(
              child: Text(
                'Login',
                style: TextStyle(fontSize: 20),
              ),
              color: Colors.green,
              textColor: Colors.white,
              onPressed: () {
                print('You pressed the button.');
```

Run this application. You can run it on an Android Emulator, or a Real Android Device. The result shall be more or less the same as shown in the following screenshot.



When user presses on the button, the callback executes the statements and prints a string to the console as shown in the console window below.



```
Run: main.dart x
Console
✓ Built build\app\outputs\apk\debug\app-debug.apk.
Installing build\app\outputs\apk\app.apk...
Syncing files to device Android SDK built for x86...
I/flutter (10946): You pressed the button.
Dart Analysis 4: Run TODO 6: Logcat Terminal
```

Conclusion

In this [Flutter Tutorial](#), we learned how to perform an action or execute a set of statements when user presses on a button in your Flutter application.

Flutter Tutorial

- ◆ [Flutter Tutorial](#)
- ◆ [Flutter - Install on Linux - Ubuntu](#)
- ◆ [Flutter - Basic Application Example](#)

Flutter Widgets

- ◆ [Flutter Text](#)
- ◆ [Flutter TextField](#)
- ◆ [Flutter FlatButton](#)
- ◆ [Flutter RaisedButton](#)
- ◆ [Flutter SnackBar](#)
- ◆ [Flutter Switch](#)
- ◆ [Flutter ToggleButtons](#)
- ◆ [Flutter Table](#)
- ◆ [Flutter DataTable](#)

◆ [Flutter Tooltip](#)

◆ [Flutter TabBar & TabBarView](#)

Flutter Animation

◆ [Flutter Animation Basic Example](#)

◆ [Flutter Animate Color](#)

Flutter Packages

◆ [Flutter sqflite - SQLite Tutorial](#)

Flutter Examples

◆ [Flutter Login Screen Sample](#)