

# Flutter CircularProgressIndicator Tutorial

## Flutter CircularProgressIndicator Widget Tutorial

---

Flutter CircularProgressIndicator is a material widget which indicates that the application is busy.

You can use CircularProgressIndicator for both determinate and indeterminate progresses.

Determinate progress bar can be used when you can show the progress of ongoing process like percentage of conversion happened when saving a file, etc. To show the progress, you have to provide a value between 0.0 and 1.0.

Indeterminate progress bar can be used when you cannot know the percentage of progress for the ongoing task. By default, CircularProgressIndicator works as an indeterminate progress bar.

### Example for Determinate CircularProgressIndicator

---

**main.dart**

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

void main() {
  runApp(MaterialApp(
    home: MyApp(),
  ));
}

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

class _MyAppState extends State<MyApp> {
  Icon fab = Icon(
    Icons.refresh,
  );

  bool showProgress = false;
  double progress = 0.2;

  void toggleSubmitState() {
    setState(() {
      showProgress = !showProgress;
    });
  }
}
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: new Text("Flutter - tutorialkart.com"),
    ),
    body: Center(
      child: showProgress
        ? CircularProgressIndicator(value:progress)
        : Text('Click on Refreseh button below', style: TextStyle(fontSize: 20,))
    floatingActionButton: FloatingActionButton(
      child: fab,
      onPressed: () => setState(() {
        showProgress = !showProgress;
        if (showProgress) {
          Future.delayed(const Duration(milliseconds: 1000), () {
            setState(() {
              progress = 0.6;
            });
          });
        }
        fab = Icon(
          Icons.stop,
        );
      } else {
        fab = Icon(Icons.refresh);
      }
    })),
  ),
);
}

```

## Output

As per the code, initially the status is at 0.2, (20% progress), then after a delay of 1000ms (1sec), the progress is updated to 0.6 (60%).

Usually, when you are using determinate circular progress indicator in your application, you would get the progress from the task you are performing in the background. And once that is 100%, or 1.0, you would hide the circular progress indicator and continue with the functionality of the application.

## [Example for Indeterminate CircularProgressIndicator](#)

In this example, we are using CircularProgressIndicator in indeterminate mode. It does not show any progress, but circles continuously, indicating the user that something is being worked out, and he may have to wait.

To use CircularProgressIndicator in indeterminate mode, you should not provide any value to the constructor CircularProgressIndicator().

```

import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(
    home: MyApp(),
  ));
}

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

class _MyAppState extends State<MyApp> {
  Icon fab = Icon(
    Icons.refresh,
  );

  bool showProgress = true;

  void toggleSubmitState() {
    setState(() {
      showProgress = !showProgress;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: new Text("Flutter - tutorialkart.com"),
      ),
      body: Center(
        child: showProgress
          ? CircularProgressIndicator()
          : Text('Click on Refreseh button below', style: TextStyle(fontSize: 20,)),
      ),
      floatingActionButton: FloatingActionButton(
        child: fab,
        onPressed: () => setState(() {
          showProgress = !showProgress;
          if (showProgress) {
            fab = Icon(
              Icons.stop,
            );
          } else {
            fab = Icon(Icons.refresh);
          }
        }
      ),
    );
  }
}

```

## Output





You can get the complete code of the Flutter Application used in the above examples at the following link.

Determinate mode: [https://github.com/tutorialkart/flutter/tree/master/flutter\\_circularprogress\\_d](https://github.com/tutorialkart/flutter/tree/master/flutter_circularprogress_d).

Indeterminate mode: [https://github.com/tutorialkart/flutter/tree/master/flutter\\_circularprogress\\_i](https://github.com/tutorialkart/flutter/tree/master/flutter_circularprogress_i).

## Conclusion

In this [Flutter Tutorial](#), we learned about `CircularProgressIndicator` widget in its determinate and indeterminate modes with examples.

### 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](#)