# Buttons

## RaisedButton

A raised button is based on a Material widget whose Material.elevation increases when the button is pressed.

flutter create --sample=material.RaisedButton.1 mysample

import 'package:flutter/material.dart';

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

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: MyStatelessWidget(),

),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Center(

child: Column(

mainAxisSize: MainAxisSize.min,

children: <Widget>[

const RaisedButton(

onPressed: null,

child: Text('Disabled Button', style: TextStyle(fontSize: 20)),

),

const SizedBox(height: 30),

RaisedButton(

onPressed: () {},

child: const Text('Enabled Button', style: TextStyle(fontSize: 20)),

),

const SizedBox(height: 30),

RaisedButton(

onPressed: () {},

textColor: Colors.white,

padding: const EdgeInsets.all(0.0),

child: Container(

decoration: const BoxDecoration(

gradient: LinearGradient(

colors: <Color>[

Color(0xFF0D47A1),

Color(0xFF1976D2),

Color(0xFF42A5F5),

],

),

),

padding: const EdgeInsets.all(10.0),

child:

const Text('Gradient Button', style: TextStyle(fontSize: 20)),

),

),

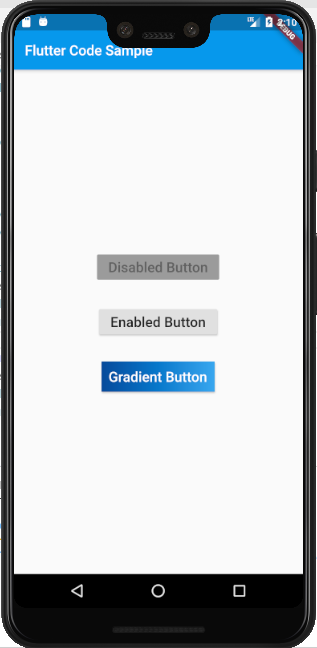
],

),

);

}

}



## FlatButton

A flat button is a text label displayed on a (zero elevation) Material widget that reacts to touches by filling with color.

Use flat buttons on toolbars, in dialogs, or inline with other content but offset from that content with padding so that the button's presence is obvious. Flat buttons intentionally do not have visible borders and must therefore rely on their position relative to other content for context. In dialogs and cards, they should be grouped together in one of the bottom corners. Avoid using flat buttons where they would blend in with other content, for example in the middle of lists.

import 'package:flutter/material.dart';

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

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: MyStatelessWidget(),

),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Center(

child: Column(

mainAxisSize: MainAxisSize.min,

children: <Widget>[

FlatButton(

color: Colors.blue,

textColor: Colors.white,

disabledColor: Colors.grey,

disabledTextColor: Colors.black,

padding: EdgeInsets.all(8.0),

splashColor: Colors.blueAccent,

onPressed: () {

/\*...\*/

},

child: Text(

"Flat Button",

style: TextStyle(fontSize: 20.0),

),

),

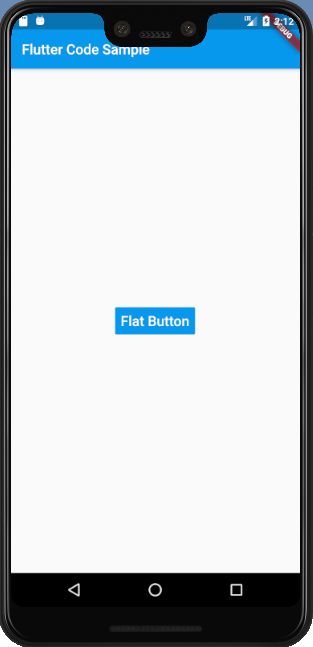
],

),

);

}

}



## IconButton

An icon button is a picture printed on a Material widget that reacts to touches by filling with color (ink).

Icon buttons are commonly used in the AppBar.actions field, but they can be used in many other places as well.

import 'package:flutter/material.dart';

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

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: Center(

child: MyStatefulWidget(),

),

),

);

}

}

double \_volume = 0.0;

class MyStatefulWidget extends StatefulWidget {

MyStatefulWidget({Key key}) : super(key: key);

@override

\_MyStatefulWidgetState createState() => \_MyStatefulWidgetState();

}

class \_MyStatefulWidgetState extends State<MyStatefulWidget> {

Widget build(BuildContext context) {

return Column(

mainAxisSize: MainAxisSize.min,

children: <Widget>[

IconButton(

icon: Icon(Icons.volume\_up),

tooltip: 'Increase volume by 10',

onPressed: () {

setState(() {

\_volume += 10;

});

},

),

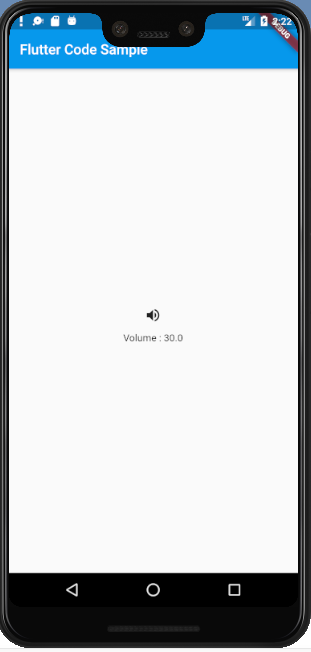
Text('Volume : $\_volume')

],

);

}

}



### Adding a filled background

Icon buttons don't support specifying a background color or other background decoration because typically the icon is just displayed on top of the parent widget's background. Icon buttons that appear in AppBar.actions are an example of this.

It's easy enough to create an icon button with a filled background using the Ink widget. The Ink widget renders a decoration on the underlying Material along with the splash and highlight InkResponse contributed by descendant widgets.

import 'package:flutter/material.dart';

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

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: MyStatelessWidget(),

),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Material(

color: Colors.white,

child: Center(

child: Ink(

decoration: const ShapeDecoration(

color: Colors.lightBlue,

shape: CircleBorder(),

),

child: IconButton(

icon: Icon(Icons.android),

color: Colors.white,

onPressed: () {},

),

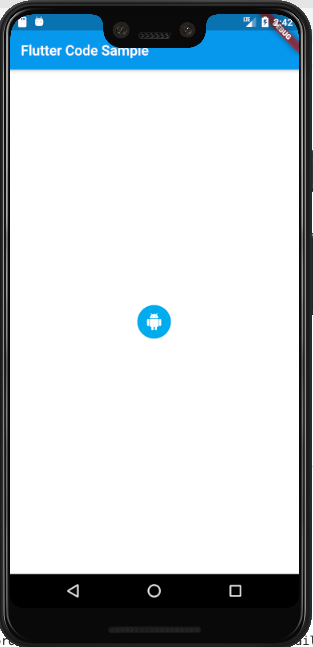
),

),

);

}

}



## InkWell

## GestureDetector

A widget that detects gestures.

import 'package:flutter/material.dart';

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

class MyApp extends StatefulWidget {

MyApp({Key key}) : super(key: key);

@override

\_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

static const String \_title = 'Flutter Code Sample';

bool \_lights = true;

String \_st = "OFF";

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: Material(

color: Colors.white,

child: Center(

child:

/\*Ink(

decoration: const ShapeDecoration(

color: Colors.lightBlue,

shape: CircleBorder(),

),

child: IconButton(

icon: Icon(Icons.android),

color: Colors.white,

onPressed: () {},

),

),\*/

Container(

alignment: FractionalOffset.center,

color: Colors.white,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Padding(

padding: const EdgeInsets.all(8.0),

child: Icon(

Icons.lightbulb\_outline,

color: \_lights ? Colors.yellow.shade600 : Colors.black,

size: 60,

),

),

GestureDetector(

onTap: () {

setState(() {

\_lights = !\_lights;

\_st = \_lights ? "OFF":"ON";

});

},

child: Container(

color: Colors.yellow.shade600,

padding: const EdgeInsets.all(8),

child: Text('TURN LIGHTS $\_st'),

),

),

],

),

),

),

),

),

);

}

}