

# Tabs

Tabs organize content across different screens, data sets, and other interactions.



## CONTENTS

Using tabs

Fixed tabs

**Scrollable tabs**

Theming tabs

## Using tabs [↗](#)

Tabs organize and allow navigation between groups of content that are related and at the same level of hierarchy.

## Installing

To use Tabs in Flutter you need to import the Material package

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

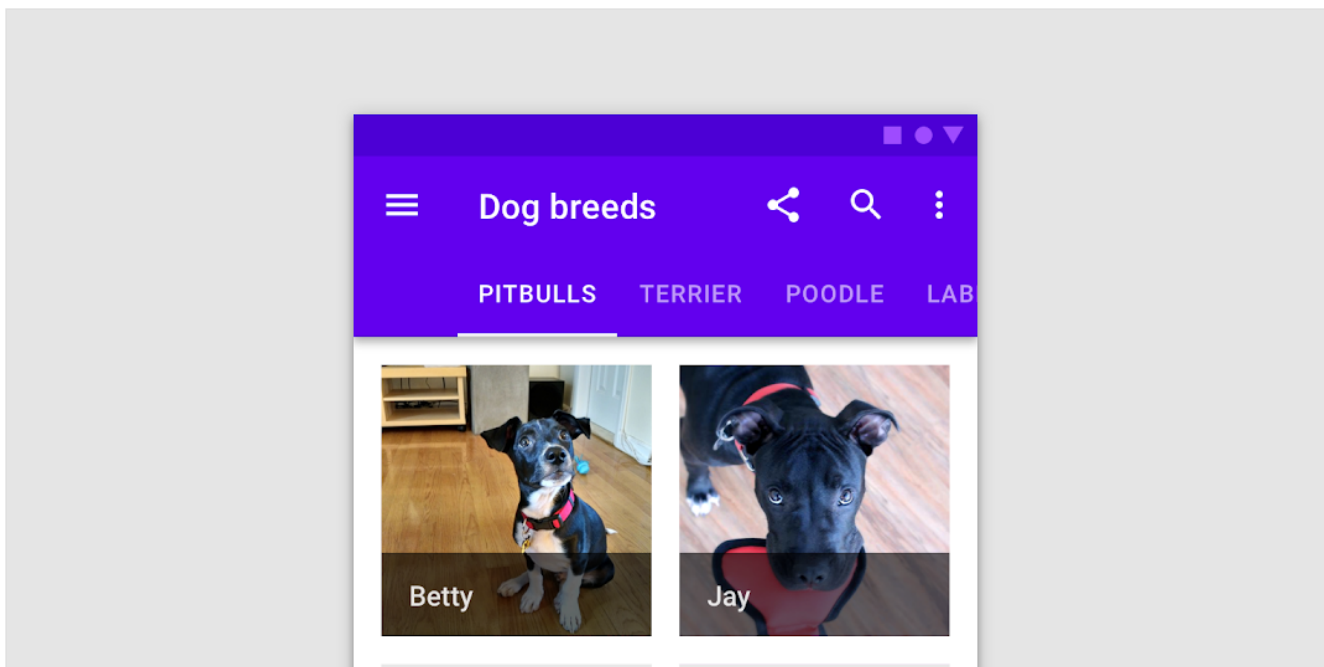
## Making tabs accessible

Flutter's APIs support accessibility setting for large fonts, screen readers, and sufficient contrast. For more information, go to Flutter's [accessibility](#) and [internationalization](#) pages.

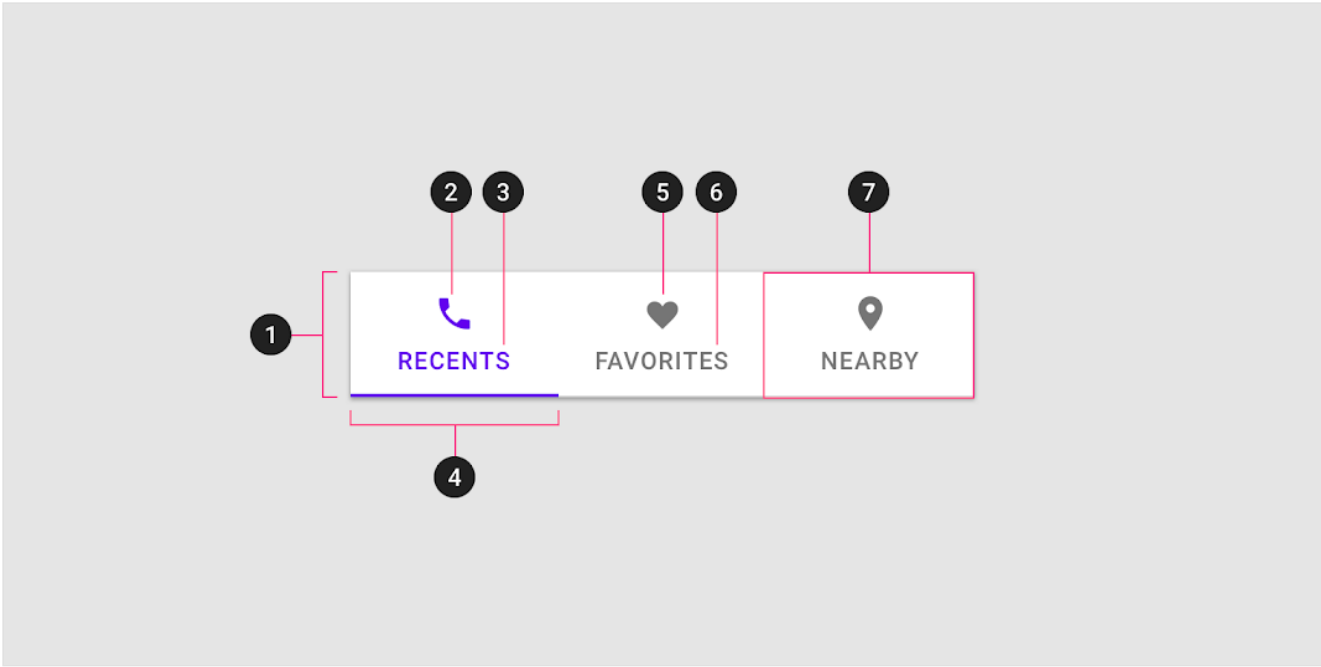
For more guidance on writing labels, go to [our page on how to write a good accessibility label](#).

## Types

There are two types of tabs: 1. [fixed tabs](#) 2. [scrollable tabs](#)



## Anatomy and key properties



- 1. Container
- 2. Active icon (Optional if there's a label)
- 3. Active text label (Optional if there's an icon)
- 4. Active tab indicator
- 5. Inactive icon (Optional if there's a label)
- 6. Inactive text label (Optional if there's an icon)
- 7. Tab item

Container attributes

	Property
Color	In the parent widget, <code>AppBar</code> assign the <code>backgroundColor</code> property
Height	Wrap your <code>TabBar</code> widget with a <code>PreferredSize</code> widget and assign <code>preferredSize</code>

Active icon attributes

	Property
<b>MaterialToolbar icon</b>	In widget <code>Tab</code> assign the <code>icon</code> parameter
<b>MaterialToolbar icon color</b>	In <code>TabBar</code> assign the <code>labelColor</code> parameter

## Active text label attributes

	Property
<b>Text label</b>	In widget <code>Tab</code> assign the <code>text</code> parameter.
<b>Color</b>	In <code>TabBar</code> assign the <code>labelColors</code> and <code>unselectedLabelColor</code> parameter
<b>Typography</b>	In <code>TabBar</code> assign the <code>labelStyle</code> and <code>unselectedLabelStyle</code> parameter

## Active tab indicator attributes

	Property
<b>Indicator</b>	Indicator exists by default for custom indicator go to <code>TabBar</code> widget and fill out the
<b>Indicator Padding</b>	In widget <code>TabBar</code> assign the <code>indicatorPadding</code> parameter.
<b>Indicator Size</b>	In <code>TabBar</code> assign the <code>indicatorSize</code> parameter
<b>Indicator Weight</b>	In <code>TabBar</code> assign the <code>indicatorWeight</code> parameter

## Inactive icon attributes

	Property
<b>MaterialToolbar icon</b>	In widget <code>Tab</code> assign the <code>icon</code> parameter
<b>MaterialToolbar icon color</b>	In <code>TabBar</code> assign the <code>unselectedLabelColor</code> parameter

## Inactive text label attributes

	Property
<b>Text label</b>	In widget <code>Tab</code> assign the <code>text</code> parameter.
<b>Color</b>	In <code>TabBar</code> assign the <code>unselectedLabelColor</code> parameter
<b>Typography</b>	In <code>TabBar</code> assign the <code>unselectedLabelStyle</code> parameter

## Tab item attributes

	Property
<b>child</b>	In widget <code>Tab</code> assign the <code>child</code> parameter. This is usually a <code>Text</code> widget, possibly wrapped in a <code>Text</code> widget.

## Fixed tabs [↗](#)

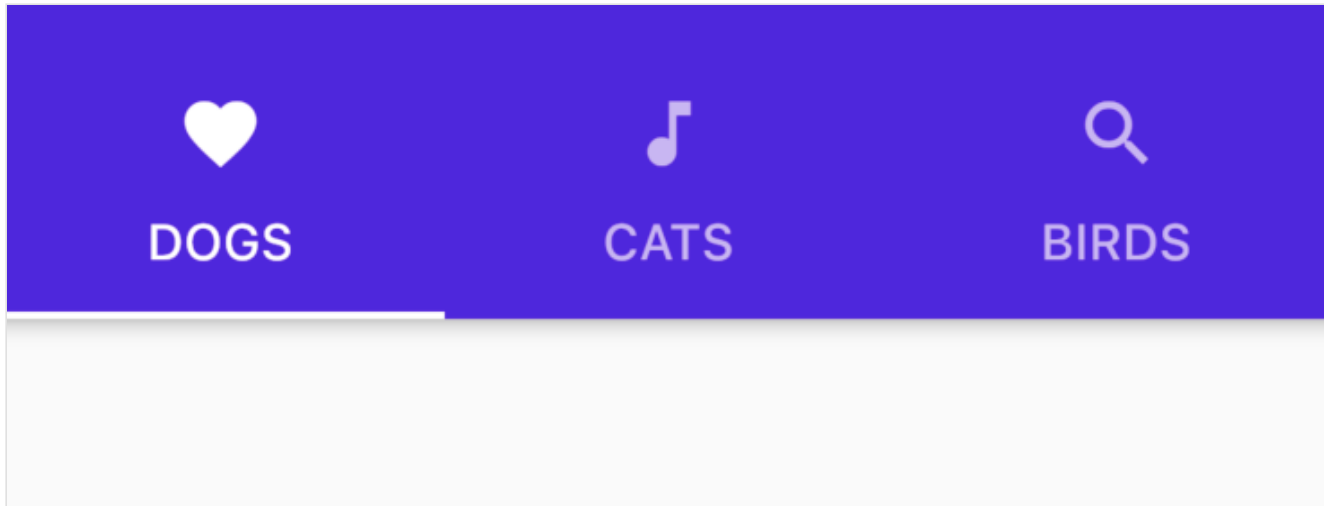
Fixed tabs display all tabs on one screen, with each tab at a fixed width. The width of each tab is determined by dividing the number of tabs by the screen width. They don't scroll to reveal more tabs; the visible tab set represents the only tabs available.

## Fixed tab example

## Tabs

- [Class definition](#)
- [GitHub source](#)
- [Dartpad demo](#)
- [YouTube video](#)

The following example shows a row of of fixed tabs.



```
MaterialApp(  
  title: 'Flutter Demo',  
  home: DefaultTabController(  
    length: 3,  
    child: Scaffold(  
      appBar: AppBar(  
        title: Text('Fixed Tabs'),  
        automaticallyImplyLeading: false,  
        backgroundColor: Color(0xff5808e5),  
        bottom: TabBar(  
          indicatorColor: Colors.white,  
          tabs: [  
            Tab(text: 'DOGS', icon: Icon(Icons.favorite)),  
            Tab(text: 'CATS', icon: Icon(Icons.music_note)),  
            Tab(text: 'BIRDS', icon: Icon(Icons.search)),  
          ],  
        ),  
      ),  
      body: TabBarView(  
        children: [  

```

```
Center(child: Text('DOGS')),  
Center(child: Text('CATS')),  
Center(child: Text('BIRDS')),  
],  
),  
),  
),  
);
```



DESIGN



IMPLEMENTATION

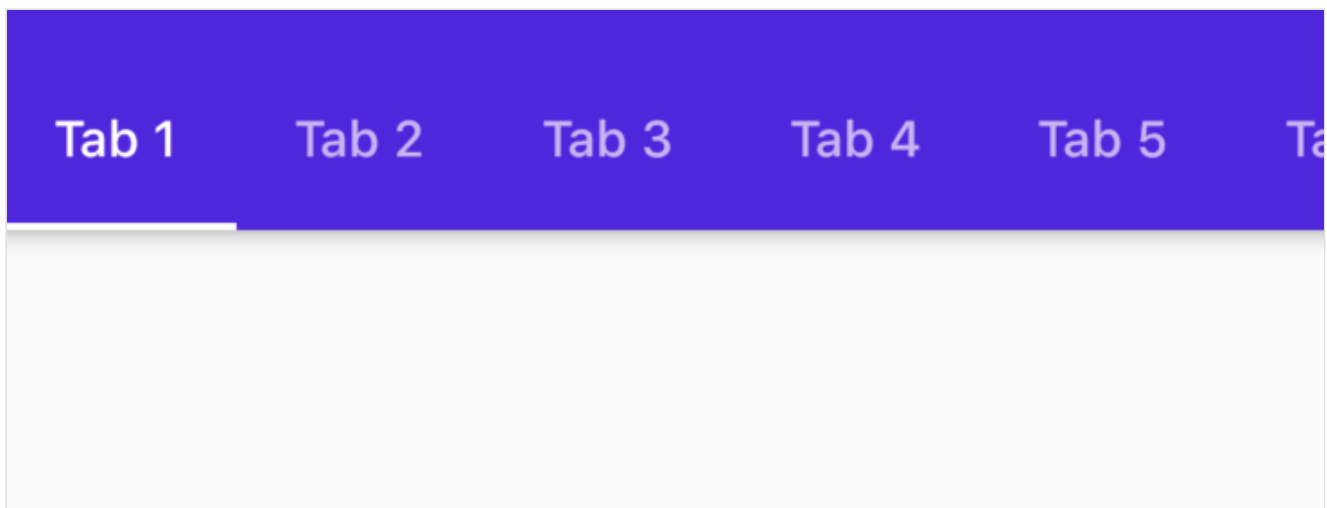
## Scrollable tabs [↗](#)

Scrollable tabs are displayed without fixed widths. They are scrollable, such that some tabs will remain off-screen until scrolled.### Fixed tab example

### Tabs

- [Class definition](#)
- [GitHub source](#)
- [Dartpad demo](#)
- [Demo site](#)

The following example shows a row of of scrollable tabs.



```
final tabs = [  
  'Tab 1',
```

```
'Tab 2',  
'Tab 3',  
'Tab 4',  
'Tab 5',  
'Tab 6',  
'Tab 7',  
'Tab 8',  
'Tab 9',  
];  
  
return MaterialApp(  
  title: 'Flutter Demo',  
  home: DefaultTabController(  
    length: tabs.length,  
    child: Scaffold(  
      appBar: AppBar(  
        title: Text('Scrollable Tabs'),  
        automaticallyImplyLeading: false,  
        backgroundColor: Color(0xff5808e5),  
        bottom: TabBar(  
          indicatorColor: Colors.white,  
          isScrollable: true,  
          tabs: [  
            for (final tab in tabs) Tab(text: tab),  
          ],  
        ),  
      ),  
      body: TabBarView(  
        children: [  
          for (final tab in tabs)  
            Center(  
              child: Text(tab),  
            ),  
        ],  
      ),  
    ),  
  ),  
);
```



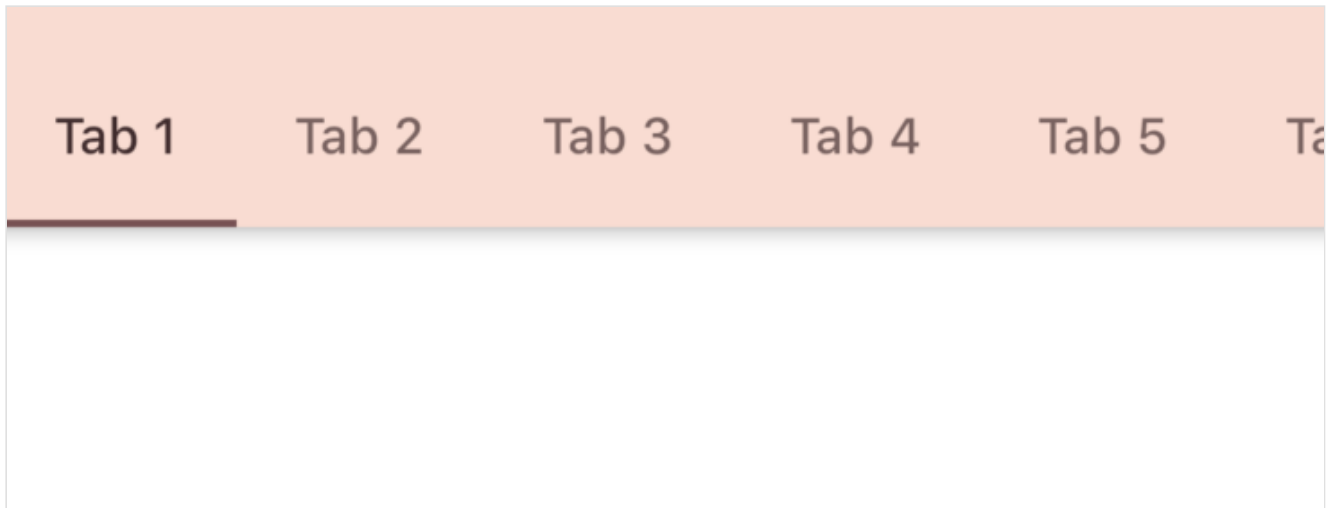
# Theming tabs

## Tabs

The following is an example data table using the [Shrine theme](#).

- [Class definition](#)
- [GitHub source](#)
- [Dartpad demo for Scrollable Theme](#)
- [Dartpad demo for Fixed Theme](#)

The following example shows a row of of scrollable tabs.



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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final tabs = [
      'Tab 1',
      'Tab 2',
      'Tab 3',
      'Tab 4',
      'Tab 5',
      'Tab 6',
      'Tab 7',
    ];
```

```
        'Tab 8',
        'Tab 9',
    ];

    return MaterialApp(
      title: 'Flutter Demo',
      theme: _buildShrineTheme(),
      home: DefaultTabController(
        length: tabs.length,
        child: Scaffold(
          appBar: AppBar(
            title: Text('Scrollable Tabs Theme'),
            automaticallyImplyLeading: false,
            bottom: TabBar(
              isScrollable: true,
              tabs: [
                for (final tab in tabs) Tab(text: tab),
              ],
            ),
          ),
          body: TabBarView(
            children: [
              for (final tab in tabs)
                Center(
                  child: Text(tab),
                ),
            ],
          ),
        ),
      );
  }
}

ThemeData _buildShrineTheme() {
  final ThemeData base = ThemeData.light();
  return base.copyWith(
    colorScheme: _shrineColorScheme,
```

```
    accentColor: shrineBrown900,
    primaryColor: shrinePink100,
    buttonColor: shrinePink100,
    scaffoldBackgroundColor: shrineBackgroundWhite,
    cardColor: shrineBackgroundWhite,
    textSelectionColor: shrinePink100,
    errorColor: shrineErrorRed,
    buttonTheme: const ButtonThemeData(
      colorScheme: _shrineColorScheme,
      textTheme: ButtonTextTheme.normal,
    ),
    primaryIconTheme: _customIconTheme(base.iconTheme),
    textTheme: _buildShrineTextTheme(base.textTheme),
    primaryTextTheme: _buildShrineTextTheme(base.primaryTextTheme),
    accentTextTheme: _buildShrineTextTheme(base.accentTextTheme),
    iconTheme: _customIconTheme(base.iconTheme),
    indicatorColor: shrineBrown600,
  );
}

IconThemeData _customIconTheme(IconThemeData original) {
  return original.copyWith(color: shrineBrown900);
}

TextTheme _buildShrineTextTheme(TextTheme base) {
  return base
    .copyWith(
      caption: base.caption.copyWith(
        fontWeight: FontWeight.w400,
        fontSize: 14,
        letterSpacing: defaultLetterSpacing,
      ),
      button: base.button.copyWith(
        fontWeight: FontWeight.w500,
        fontSize: 14,
        letterSpacing: defaultLetterSpacing,
      ),
    )
}
```

```
.apply(  
  fontFamily: 'Rubik',  
  displayColor: shrineBrown900,  
  bodyColor: shrineBrown900,  
);  
}  
  
const ColorScheme _shrineColorScheme = ColorScheme(  
  primary: shrinePink100,  
  primaryVariant: shrineBrown900,  
  secondary: shrinePink50,  
  secondaryVariant: shrineBrown900,  
  surface: shrineSurfaceWhite,  
  background: shrineBackgroundWhite,  
  error: shrineErrorRed,  
  onPrimary: shrineBrown900,  
  onSecondary: shrineBrown900,  
  onSurface: shrineBrown900,  
  onBackground: shrineBrown900,  
  onError: shrineSurfaceWhite,  
  brightness: Brightness.light,  
);  
  
const Color shrinePink50 = Color(0xFFEEAE6);  
const Color shrinePink100 = Color(0xFFEDBD0);  
const Color shrinePink300 = Color(0xFFBB8AC);  
const Color shrinePink400 = Color(0xFFEA4A4);  
  
const Color shrineBrown900 = Color(0xFF442B2D);  
const Color shrineBrown600 = Color(0xFF7D4F52);  
  
const Color shrineErrorRed = Color(0xFFC5032B);  
  
const Color shrineSurfaceWhite = Color(0xFFFFFBFA);  
const Color shrineBackgroundWhite = Colors.white;  
  
const defaultLetterSpacing = 0.03;
```



## Up next

Name

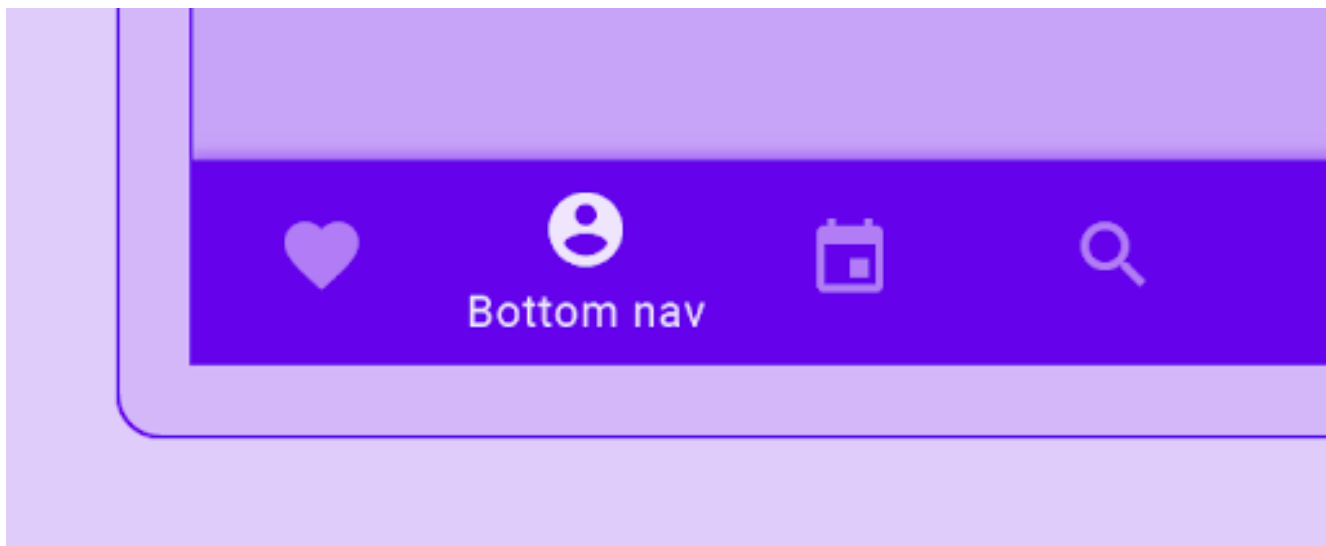
Phone

Area ▼

### COMPONENTS

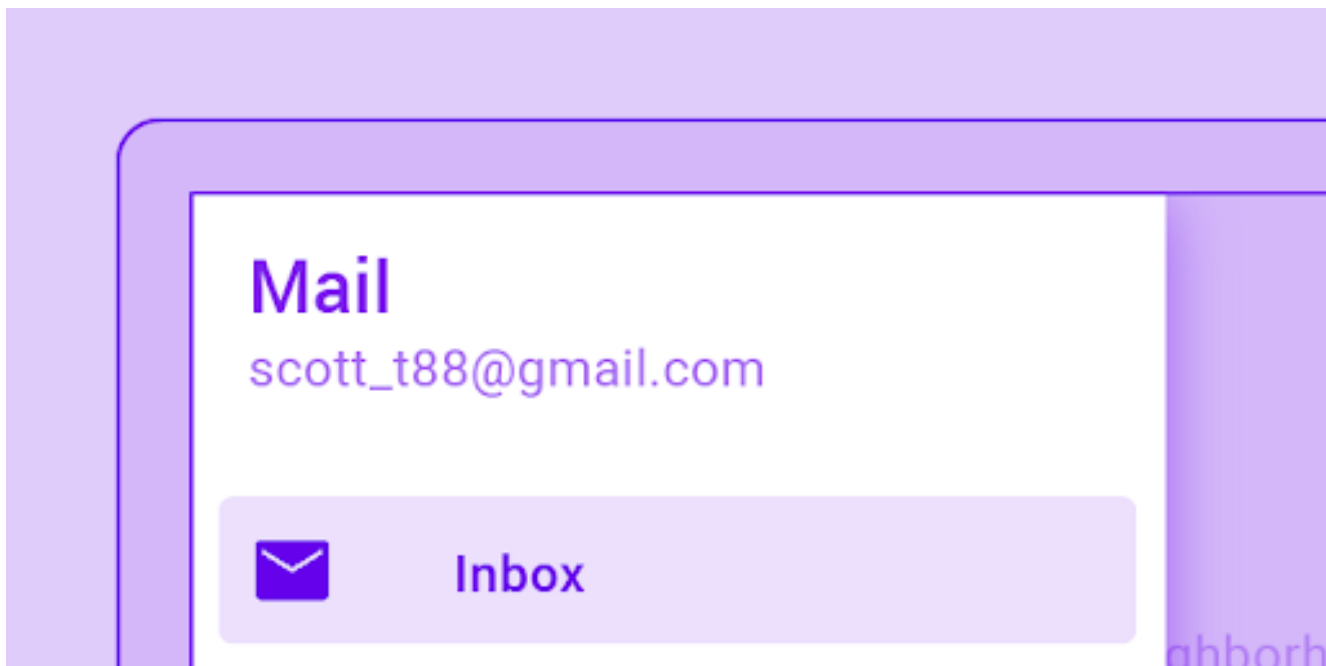
## Text fields





## COMPONENTS

## Bottom navigation



## COMPONENTS

## Navigation drawer



Material is an adaptable system of guidelines, components, and tools that support the best practices of user interface design. Backed by open-source code, Material streamlines collaboration between designers and developers, and helps teams quickly build beautiful products.

[GitHub](#) / [Twitter](#) / [YouTube](#) / [Blog RSS](#) / [Subscribe for updates](#)

---

Google

[Privacy Policy](#)

[Terms of Service](#)

[Feedback](#)