

In Flutter, widgets are the building blocks of the user interface, and they come in two types: **Stateless Widgets** and **Stateful Widgets**.

## Stateless Widgets

- **Definition:** A widget that does not require mutable state. This means that once the widget is built, it cannot change. It will remain the same until it is destroyed.
- **Use Case:** You use a Stateless Widget when your UI does not depend on any changes during runtime. For example, displaying static text, images, or buttons that don't need to change after being built.
- **Example:**

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

class MyStatelessWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Stateless Widget Example'),
      ),
      body: Center(
        child: Text('This is a stateless widget'),
      ),
    );
  }
}
```

## Stateful Widgets

- **Definition:** A widget that has a mutable state, meaning it can change during the runtime of the app. When the state changes, the widget is rebuilt to reflect the changes.
- **Use Case:** You use a Stateful Widget when your UI depends on changes that occur during runtime. For example, a counter app where the number changes when a button is pressed, forms where input fields need to update dynamically, or UI elements that react to user interactions.
- **Example:**

```

import 'package:flutter/material.dart';

class MyStatefulWidget extends StatefulWidget {
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Stateful Widget Example'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text('You have pressed the button this many times:'),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headline4,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: Icon(Icons.add),
      ),
    );
  }
}

```

# Key Differences

- **State Management:**
  - Stateless Widgets don't have internal state management.
  - Stateful Widgets can manage their own state, which can change over time.
- **Rebuilds:**
  - Stateless Widgets are rebuilt only when the parent widget changes.
  - Stateful Widgets can be rebuilt multiple times due to changes in the state.

Understanding when to use each type of widget is essential for effective Flutter development. If your widget doesn't need to change, use a Stateless Widget for simplicity. If it needs to react to user input or other dynamic changes, use a Stateful Widget.