

# StatelessWidget

StatelessWidget 是没有 State（状态）的 Widget，当 Widget 在运行时不需要改变时，就用 StatelessWidget。

## StatelessWidget 的实现

下面一段代码是一段 StatelessWidget 的 demo 代码，将下面代码复制到 `main.dart` 里并运行：

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

void main() => runApp(MyApp("Hello World"));

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  final String content;

  MyApp(this.content);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        body: Center(
          child: Text(content),
        ),
      ),
    );
  }
}
```

MyApp 就是一个 StatelessWidget, 它继承自 StatelessWidget, Text 也是 StatelessWidget, 因为 Text 也是继承自 StatelessWidget。

运行的效果为:

8:07

50%



DEBUG

Hello World!

要实现自定的 StatelessWidget，必须要继承 StatelessWidget。

看下面实现的代码：

```
class MyApp extends StatelessWidget {  
  
  @override  
  Widget build(BuildContext context) {  
    return ...  
  };  
}
```

1. 首先继承 StatelessWidget。
2. 必须要实现 build 函数，返回一个 Widget。

## StatelessWidget: immutable (状态不可变)

StatelessWidget 是没有 State (状态) 的，而且 StatelessWidget 只能在加载/构建 Widget 时才绘制一次，这意味着无法基于任何事件或用户操作重绘 StatelessWidget，所以 StatelessWidget 是 immutable 的。

假设在 APP 运行过程中，将 Text 的文本改为 Hello Flutter World，那么 Text 的内容不会更改，APP 的 UI 也不会改变。

## StatelessWidget 的生命周期

StatelessWidget 的生命周期就只有一个，即 build 函数：

- build (build函数)

在代码中加入如下：

```
class MyApp extends StatelessWidget {  
  // This widget is the root of your application.  
  final String content;  
  
  MyApp(this.content);  
  
  @override  
  Widget build(BuildContext context) {  
    print("build"); //StatelessWidget -- build  
    ....  
  }  
}
```

运行后可以看到如下的log：

```
Launching lib/main.dart on iPhone XR in debug  
mode...  
Xcode build done.  
3.9s  
flutter: build
```

## StatelessWidget 使用注意事项

如果你想要为 StatelessWidget 赋值，只能在 StatelessWidget 初始化的时候，通过构造函数传递一些额外的参数。但是请记住，这些参数不会在以后阶段发生变化，即使发生变化，也只能按原值使用。因为 StatelessWidget 只会渲染一次。