

# 一、MaterialApp

MaterialApp表示当前的风格是Material Design，Android中使用的UI风格是Material Design，而ios则是cupertino Design，这里我们构建的是Android应用，因此使用的MaterialApp。

而为了使用已经封装好的Material Design风格的Widget，因此必须使用MaterialApp，而它也作为Widget树中的Root Widget

## 二、Scaffold

Scaffold是实现了Material Design的基本布局结构，例如AppBar、Drawer、SnackBar等，为了使用这些基本布局，必须使用Scaffold。

因此我们将MaterialApp作为Root Widget，而Root Widget的子Widget是Scaffold，我们添加的Widget就作为Scaffold的子Widget

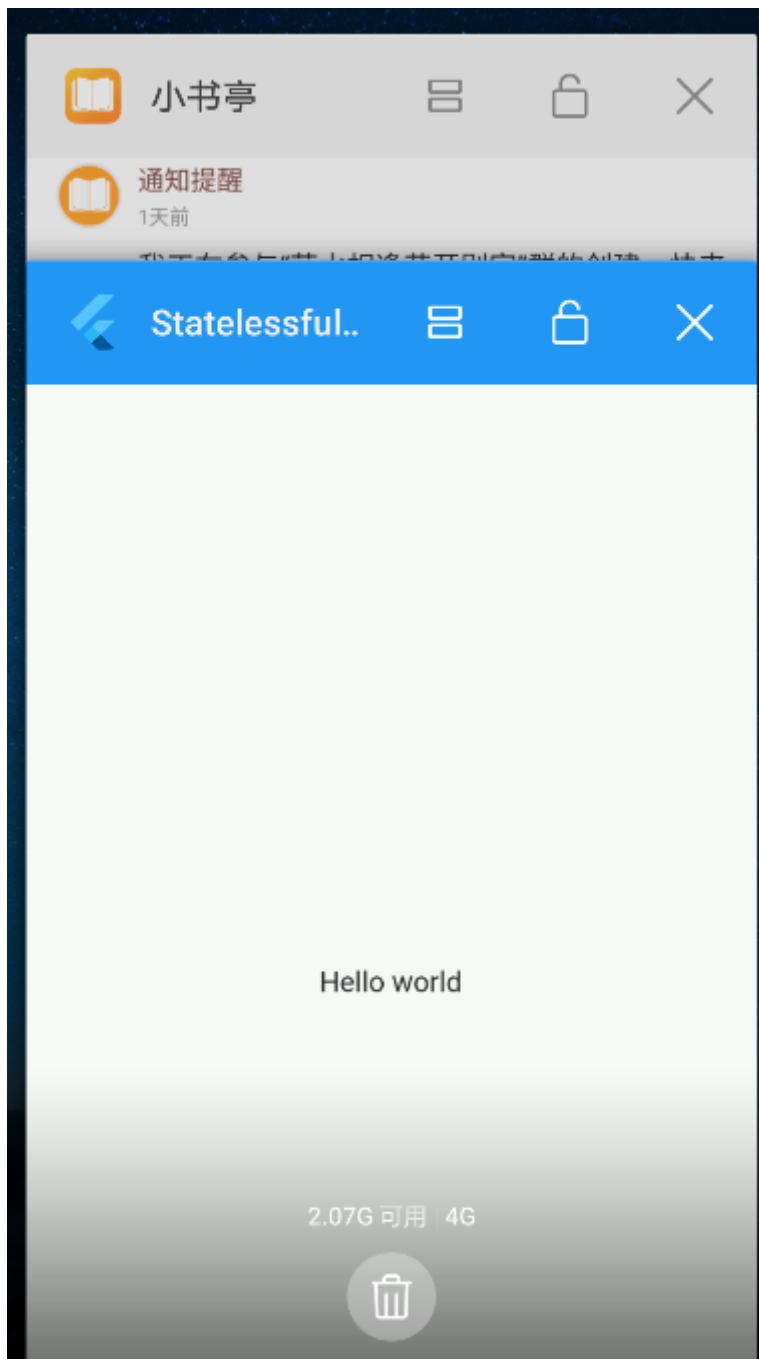
## 三、MaterialApp的使用

MaterialApp一共有25个属性，常用的主要有以下几个：

### 1、title

主要是在Android任务管理器或者ios程序切换时显示的标题

```
1 return MaterialApp(  
2   title: 'StatelessfulWidget Demo',  
3   home: Scaffold(  
4     appBar: AppBar(  
5       title: Text('Flutter Demo'),  
6     ),  
7     body: Center(  
8       child: Text(content),  
9     ),  
10  ),  
11 );
```

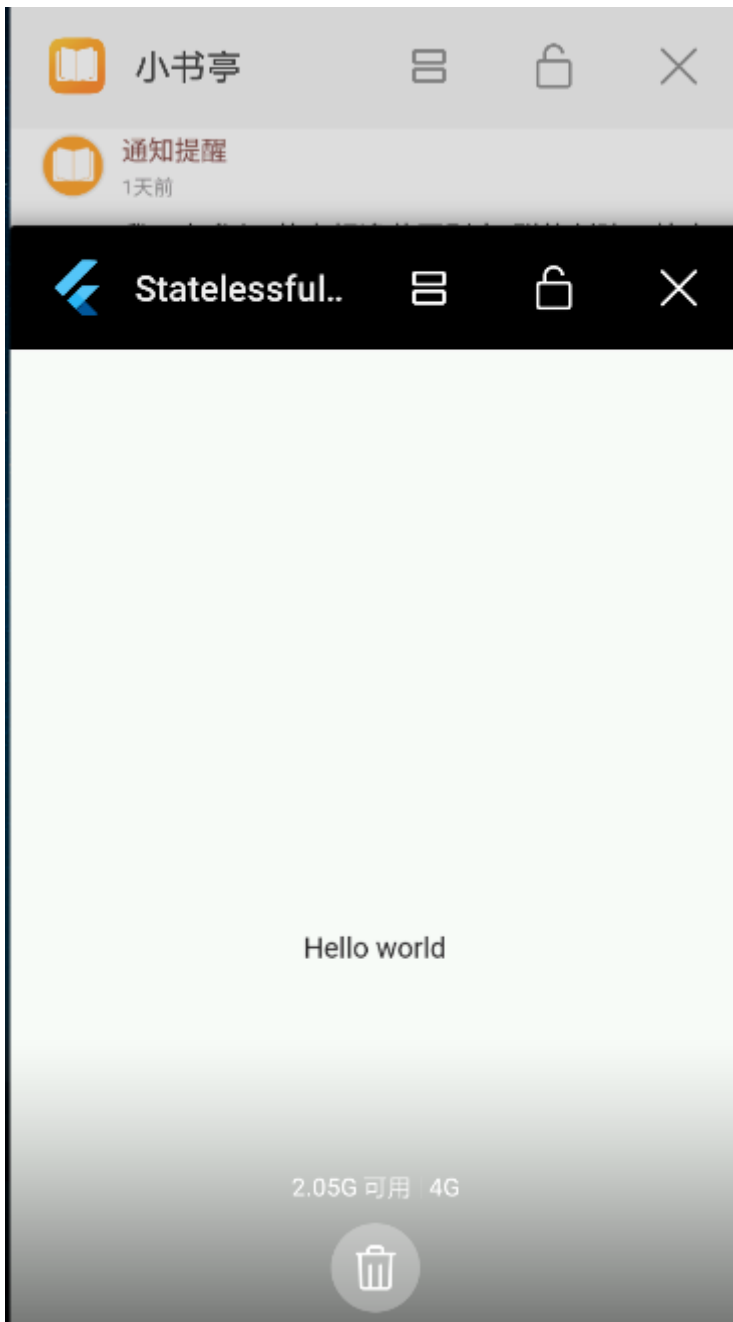


## 2、color

该有颜色是任务管理器切换时的背景颜色

```
1 return MaterialApp(  
2   title: 'StatelessfulWidget Demo',  
3   color: Colors.black,  
4   home: Scaffold(  
5     appBar: AppBar(  
6       title: Text('Flutter Demo'),  
7     ),  
8     body: Center(  
9       child: Text(content),  
10    ),  
11  ),  
12 );
```

```
11  ),  
12  );
```



### 3、theme

ThemeData类型

设置App的主题，比如颜色、字体

### 4、home

Widget类型

进入程序后显示的第一个类型，必须是Scaffold

```
1  return MaterialApp(  
2    title: 'StatelessfulWidget Demo',
```

```
3   color: Colors.black,  
4   home: Scaffold(  
5     appBar: AppBar(  
6       title: Text('Flutter Demo'),  
7     ),  
8     body: Center(  
9       child: Text(content),  
10    ),  
11  ),  
12 );
```

## 四、Scaffold的使用

Scaffold总共有16个属性，下面介绍我们常用的属性

### 1、appBar

AppBar类型

设置页面顶部的标题栏，不设置的话默认不显示

### 2、backgroundColor

Color类型

背景颜色，类似于Android页面根布局中的background属性

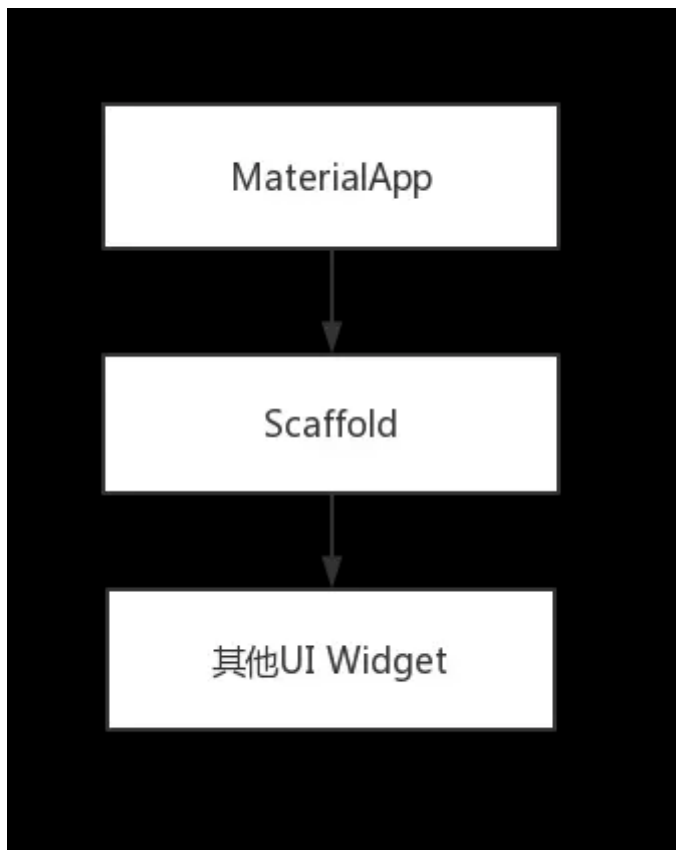
### 3、body

Widget类型

body中就是我们要添加的组件

## 五、Flutter App的Widget树

因为大部分情况下我们都需要使用MaterialApp和Scaffold，因此Flutter的App中的Widget树经常是这样的



## 六、关于Android和IOS不同主题的疑问

1、大部分情况下推荐使用Material Design，因为目前Flutter对Material Design的Widget支持

的更多，而Cupertino的组件相对较少

2、可以对一份设计图设计两份界面，一份Material Design，一份Cupertino，然后在Widget中通过if-else返回不同的Widget