一、Flutter写UI方式

Flutter采用了一种新的方式来写UI，就是通过代码声明式。

1、Android的UI布局

（1）XML声明式

通过XML来布局，但是在代码了无法直接修改UI，只能通过findViewById之后再修改UI

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center_horizontal"
 android:orientation="vertical">

 <ImageView
  android:layout_width="@dimen/dp_72"
  android:layout_height="@dimen/dp_72"
  android:layout_marginTop="@dimen/dp_40"
  android:src="@drawable/global_nearby_user_location" />

  <TextView
  android:id="@+id/tv_nearby_user_location"
```

```
  style="@style/BaseWrapParent"
  android:layout_marginTop="@dimen/dp_12"
  android:textSize="@dimen/textsize_15"
  tools:text="开启 定位功能 后你就可以查看附近的书友啦~" />

  <TextView
  android:id="@+id/tv_open_location"
  android:layout_width="@dimen/dp_255"
  android:layout_height="@dimen/dp_43"
  android:layout_marginTop="@dimen/dp_41"
  android:background="@drawable/selector_yellow_ff6a33_22dp_yellow_ff6a3
3"
  android:gravity="center"
  android:text="@string/tv_nearby_bookshelf_fragment_open_location"
  android:textColor="@color/white_FFFFFF"
  android:textSize="@dimen/textsize_15" />
</LinearLayout>
```

（2）代码命令式

代码命令式可以直接来修改UI，但是代码中无法看到布局的UI元素的关系和嵌套层次

```
// 分别实现各个view
ViewA a = new ViewA(...)
ViewB b = new ViewB(...)
ViewC c1 = new ViewC(...)
ViewC c2 = New ViewC(...)
// 然后将子view 添加到容器中
a.add(b)
b.add(c1)
b.add(c2)
```

如果这里有很多UI，我们无法直接看UI元素的关系和布局层次

2、IOS

（1）Storyboard

通过Storyboard来组织视图和设置约束

（2）代码命令式

可以直接操作组件，但是无法直接看出UI元素的关系和布局层次

3、Flutter的UI写法

Flutter使用的是代码声明式，只能通过代码来书写布局，主要优点是两方面

- 可以在代码中直接操作UI
- 可以在代码中看出UI组件之间的关系，同时也可以看出布局层次
- 减少了XML解析的步骤，构建效率更高

```
class TestStatelessWidget extends StatelessWidget {
 final String content;

 TestStatelessWidget(this.content);

 @override
 Widget build(BuildContext context) {
 // TODO: implement build
 return MaterialApp(
  title: 'StatelessfulWidget Demo',
  color: Colors.black,
  home: Scaffold(
  appBar: AppBar(
  title: Text('Flutter Demo'),
  ),
  body: Center(
  child: Text(content),
  ),
  ),
  );
  }
```