

Flutter 目录结构介绍、入口、自定义 Widget、Center 组件、Text 组件、MaterialApp 组件、Scaffold 组件

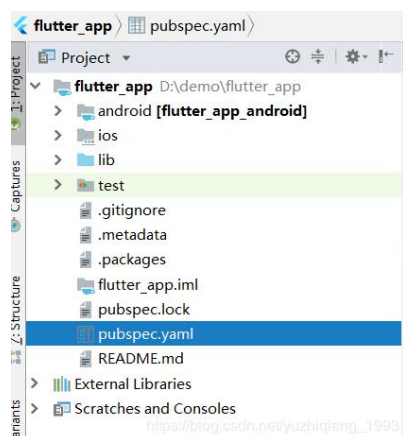
主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html> （IT 营）

一、Flutter 目录结构介绍.....	1
二、Flutter 入口文件、入口方法.....	2
三、Flutter 第一个 Demo Center 组件的使用.....	2
四、Flutter 把内容单独抽离成一个组件.....	3
五、给 Text 组件增加一些装饰.....	4
六、件用 MaterialApp 和 Scaffold 两个组件装饰 App.....	4
1、MaterialApp.....	4
2、Scaffold.....	5
七、Vscode 调试 Flutter 项目.....	6

一、Flutter 目录结构介绍



我们着重需要注意一下几个文件夹, 其他的暂时不用理会

文件夹	作用
android	android 平台相关代码
ios	ios 平台相关代码
lib	flutter 相关代码, 我们主要编写的代码就在这个文件夹
test	用于存放测试代码
pubspec.yaml	配置文件, 一般存放一些第三方库的依赖。

二、Flutter 入口文件、入口方法

每一个 flutter 项目的 lib 目录里面都有一个 main.dart 这个文件就是 flutter 的入口文件

main.dart 里面的

```
void main(){  
  runApp(MyApp());  
}  
  
也可以简写  
void main()=>runApp(MyApp());
```

其中的 main 方法是 dart 的入口方法。runApp 方法是 flutter 的入口方法。
MyApp 是自定义的一个组件

三、Flutter 第一个 Demo Center 组件的使用

```
import 'package:flutter/material.dart';  
void main(){  
  runApp(Center(  
    child: Text(  
      "我是一个文本内容",  
      textDirection: TextDirection.ltr,  
    ),  
  ));  
}
```

四、Flutter 把内容单独抽离成一个组件

在 Flutter 中自定义组件其实就是一个类，这个类需要继承 StatelessWidget/StatefulWidget 前期我们都继承 StatelessWidget。后期给大家讲 StatefulWidget 的使用。

StatelessWidget 是无状态组件，状态不可变的 widget

StatefulWidget 是有状态组件，持有的状态可能在 widget 生命周期改变

```
import 'package:flutter/material.dart';  
  
void main(){  
  runApp(MyApp());  
}  
class MyApp extends StatelessWidget{  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
    return Center(  
      child: Text(  
        "我是一个文本内容",  
        textDirection: TextDirection.ltr,  
      ),  
    );  
  }  
}
```

五、给 Text 组件增加一些装饰

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

void main(){
  runApp(MyApp());
}

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return Center(
      child: Text(
        "我是 Dart 一个文本内容",
        textDirection: TextDirection.ltr,
        style: TextStyle(
          fontSize: 40.0,
          fontWeight: FontWeight.bold,
          // color: Colors.yellow
          color: Color.fromRGBO(255, 222, 222, 0.5)

        ),
      ),
    );
  }
}
```

六、件用 MaterialApp 和 Scaffold 两个组件装饰 App

1、MaterialApp

MaterialApp 是一个方便的 Widget，它封装了应用程序实现 Material Design 所需要的一些 Widget。一般作为顶层 widget 使用。

常用的属性：

home（主页）

title（标题）

color（颜色）

theme（主题）

routes（路由）

...

2、 Scaffold

Scaffold 是 Material Design 布局结构的基本实现。此类提供了用于显示 drawer、snackbar 和底部 sheet 的 API。

Scaffold 有下面几个主要属性：

appBar - 显示在界面顶部的一个 AppBar。

body - 当前界面所显示的主要内容 Widget。

drawer - 抽屉菜单控件。

...

```
import 'package:flutter/material.dart';
void main(){
  runApp(MyApp());
}
class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      title:"我是一个标题",
      home:Scaffold(
        appBar: AppBar(
          title:Text('IT 营'),
          elevation: 30.0, //设置标题阴影 不需要的话值设置成 0.0
        ),
```

```
        body: MyHome(),
      ),
      theme: ThemeData( //设置主题颜色
        primarySwatch: Colors.yellow
      ),
    );
  }
}

class MyHome extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return Center(
      child: Text(
        "我是 Dart 一个文本内容",
        textDirection: TextDirection.ltr,
        style: TextStyle(
          fontSize: 40.0,
          fontWeight: FontWeight.bold,
          color: Colors.black38
          // color: Color.fromRGBO(255, 222, 222, 0.5)

        ),
      ),
    );
  }
}
```

七、Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键：点击后热加载，也就算是重新加载吧。

p 键：显示网格，这个可以很好的掌握布局情况，工作中很有用。

o 键：切换 android 和 ios 的预览模式。

q 键：退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```