

# flutter 学习

## 1. 课前复习

- 常用组件回顾
- 布局方式
- 网络加载

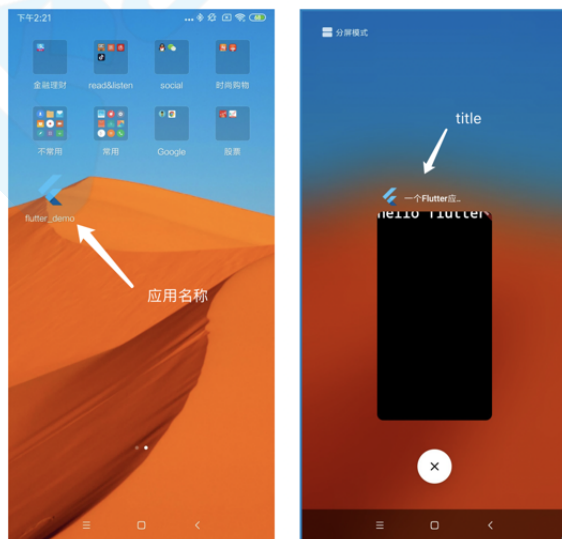
## 2. 课堂目标

- MaterialApp、Scaffold学习
- 项目开发

## 3. 知识点

### 1. 组件学习

1. **MaterialApp**: 封装了应用程序实现Material Design所需要的一些Widget, 实际是一种设计风格, 里面会有已有的一些组件 (eg: theme)
  - title: 该属性会在 Android 应用管理器的App上方显示, 对于 ios 设备是没有效果的



- home: Widget 类型, 这是在应用程序正常启动时首先显示的Widget, 除非指定了 initialRoute。如果 initialRoute 显示失败, 也该显示该Widget。
- routes: Map<String, WidgetBuilder> 类型, 是应用的顶级路由表, 当使用 Navigator.pushNamed 进行命名路由的跳转时, 会在此路表中进行查找并跳转

```
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(
      primarySwatch: Colors.blue,
    ),
    home: HomePage(),
    routes: {
      '/home': (BuildContext context) => HomePage(),
      '/detail': (BuildContext context) => DetailPage()
    },
  );
}
```

```
Widget build(BuildContext context) {
  return Center(
    child: GestureDetector(
      onTap: (){
        Navigator.pushNamed(context, 'routeName');
      },
      child: Text(
        '首页点击跳转详情页',
      ),
    ),
  );
}
```

## 2. Scaffold：实现了基本的 Material Design 布局结构

- 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
- bottomNavigationBar：显示在页面底部的导航栏，`items` 必须大于2个

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('首页'),
          // centerTitle: false,
        ),
      ),
    );
  }
}
```

```

drawer: Drawer(
  child: Column(
    children: <Widget>[
      DrawerItem(1, '选项1'),
      DrawerItem(2, '选项2'),
      DrawerItem(3, '选项3'),
      DrawerItem(4, '选项4'),
      DrawerItem(5, '选项5')
    ],
  ),
),
bottomNavigationBar: BottomNavigationBar(
  type: BottomNavigationBarType.fixed,
  currentIndex: 1,
  items: [
    new BottomNavigationBarItem(
      icon: Icon(Icons.account_balance), title: Text('标题一')),
    new BottomNavigationBarItem(
      icon: Icon(Icons.contacts), title: Text('标题二')),
    new BottomNavigationBarItem(
      icon: Icon(Icons.library_music), title: Text('标题三'))
  ],
),
body: Center(
  child: Text('THIS IS BODY'),
),
),
);
}
}

```

## 2.项目开发

### 1. 配置文件设置

项目中的pubspec.yaml文件

```

#name很重要，如果修改了name所有的dart的文件的import前引用的本地的文件啊的包名都需要修改
name: flutterdemo
description: A new Flutter application.

#发布环境
dependencies:
  flutter:
    sdk: flutter

#添加依赖packages ^表示适配和当前大版本一致的版本，~表示适配和当前小版本一致的版本

```

#eg: ^表示适配和当前大版本一致的版本, 即2.x.x都可, ~表示适配和当前小版本一致的版本, 即2.1.x都可

```
dio: ^2.1.0
cupertino_icons: ^0.1.2
english_words: ^3.1.0
# image_picker: ^0.4.8
```

#本地环境开发

```
dev_dependencies:
  flutter_test:
    sdk: flutter
```

#启用国际化

```
flutter_localizations:
  sdk: flutter
```

#数组

```
server:
  - aaaaaaa
  - bbbbbbb
  - ddddddd
```

#常量

```
age: 22          # int
boolitem: true   #定义一个boolean值
name: 'hello'    #定义一个string
```

flutter:

```
# The following line ensures that the Material Icons font is
# included with your application, so that you can use the icons in
# the material Icons class.
uses-material-design: true
```

# To add assets to your application, add an assets section, like this:

#添加资源, 不单单是图片, images是个和pubspec.yaml配置文件同级的目录, 如果不同级, 需要添加..

```
assets:
  - images/park.jpg
  - images/lake.jpg
  - images/touxiang.jpg
# - images/a_dot_burr.jpeg
# - images/a_dot_ham.jpeg
```

#字体样式

```
fonts:
  #family与fonts是一个整体, 列表的一项
  - family: Schyler
    fonts:
      - asset: fonts/Schyler-Regular.ttf
      - asset: fonts/Schyler-Italic.ttf
```

```
        style: italic
- family: Trajan Pro
  fonts:
    - asset: fonts/TrajanPro.ttf
    - asset: fonts/TrajanPro_Bold.ttf
    weight: 700
```

注意点：

- 大小写敏感
- 缩进代表层级，使用空格，默认2个空格（flutter工具做了处理，tab也可以）
- "#" 表示注释内容 ":" 表示键值对，注意后面要空格 "{}" 表示键值表 "-" 表示列表，注意后面要空格 "[]" 表示数组，注意每项之间有空格 "?" 表示复杂的键

## 2. 主题theme

在 Flutter 中使用 ThemeData 来在应用中共享颜色和字体样式，Theme有两种：全局Theme和局部Theme。全局Theme是由应用程序根 MaterialApp 创建的Theme。

- primaryIconTheme: IconThemeData 类型，一个与主色对比的图片主题
- primaryColor: Color 类型，App主要部分的背景色（ToolBar,Tabbar等）
- accentColor: Color 类型，前景色（文本、按钮等）
- primaryTextTheme: TextThemeData 类型，一个与主色对比的文本主题
- errorColor: Color 类型，用于输入验证错误的颜色，例如在 TextField 中
- disabledColor: Color 类型，用于 Widget 无效的颜色，无论任何状态。eg禁用复选框

本节课重点是完成如下界面



## 4. 总结

---

- 如何创建TabController
- 如何配置相关设置

## 5. 作业 && 答疑

---

完成今天相关代码内容

## 6. 下节课内容

---

- 请提前熟悉MVC相关思想

