

白球手札：微信小程序学习摘要

框架

微信小程序是一个框架，小程序开发框架。

目标是通过尽可能**简单、高效**的方式让开发者可以在微信中开发具有原生 APP 体验的服务。

框架结构

自己的**视图层描述语言（View）** WXML 和 WXSS

基于 JavaScript 的**逻辑层（App Service）**框架

视图层与逻辑层间提供了**数据传输和事件系统**

特点

响应的数据绑定

整个系统分为两块视图层（View）和逻辑层（App Service）

只需要在逻辑层修改数据，视图层就会做相应的更新。

省心的页面管理

框架 管理了整个小程序的页面路由。

可以做到页面间的**无缝切换**，并给以页面**完整的生命周期**。

只要将页面的数据，方法，生命周期函数注册进 框架 中，其他的一切复杂的操作都交由 框架 处理。

更加微信“原生”

与在微信上浏览 webAPP 相比，会减少脱节感。

文件结构

小程序包含一个描述整体程序的 **app** 和多个描述各自页面的 **page**。

关键文件三个

最关键也是必不可少的，是 `app.js`、`app.json`、`app.wxss` 这三个。

`.js` 后缀的是脚本文件

监听并处理小程序的生命周期函数、

声明全局变量。

调用框架提供的丰富的 API

`.json` 后缀的文件是配置文件

`app.json` 是对整个小程序的全局配置。

配置小程序是由**哪些页面组成**

配置小程序的**窗口 背景色**，配置**导航条样式**，

配置**默认标题**

注意该文件不可添加任何注释。

`.wxss` 后缀的是样式表文件

就是 css

关键文件夹 pages

每一个页面的【路径+页面名】都需要写在 `app.json` 的 `pages` 中

且 `pages` 中的第一个页面是小程序的首页。

小程序页面

组成

描述页面的这四个文件必须具有相同的路径与文件名

`.js` 后缀的文件是脚本文件，

监听并处理页面的生命周期函数、

获取小程序实例，

声明并处理数据，

响应页面交互事件

`.json` 后缀的文件是配置文件，

`.wxss` 后缀的是样式表文件，

`.wxml` 后缀的文件是页面结构文件。

`.json`

非必要

将会在 `app.json` 的基础上，覆盖相同的配置项。

`.wxml`

```
<!--index.wxml-->
<view class="container">
  <view bindtap="bindViewTap" class="userinfo">
    <image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"></image>
    <text class="userinfo-nickname">{{userInfo.nickName}}</text>
  </view>
  <view class="usermotto">
```

```
<text class="user-motto">{{motto}}</text>
</view>
</view>
```

具有新的标签

支持自刷新的数据传递方式

MVVC

.WXSS

当有页面样式表时，页面的样式表中的样式规则会**层叠覆盖** **app.wxss** 中的样式规则。

如果不指定页面的样式表，也可以在页面的结构文件中**直接使用** **app.wxss** 中指定的样式规则。