

# 邂逅小程序开发

讲师：王红元  
微博：coderwhy



小程序交流群



加小撩拿资料



老师答疑交流



# 课前提醒

## ■ 1.课程的录播：腾讯课堂只有第一天可以回看，所以我会把上课视频录下来

- 大家可以加 小撩微信 回看每天录制的视频
- 建议尽量来听直播，可以有更多的交流

## ■ 2.不同基础人群的适配

### □ 从零学习如何理解？

- 小程序相关知识的从零学习，甚至你没用过小程序也可以；
- 但是这个从零并不意味着你不需要任何的编程基础，你需要具体一些基础知识；

### □ 每个人的基础不同、知识存储不一样；

### □ 相互学习、共同进步；





# 学习路线和课程大纲





# 阶段一大纲





# 什么是小程序?

- 官方解释：微信小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。
- 我的解释：小程序是微信生态的一部分，它提供了一种更加 **方便和高效** 的用户交互方式。







# 小程序的出现到底带来了什么呢?

## ■ 1. 用户使用的便捷性

- 普通的APP我们需要的使用过程：了解APP - 下载 - 打开 - 注册 - 使用
- 小程序版本的APP的使用过程：了解APP - 打开(扫码/搜索) - 使用

## ■ 我们来设想一个场景：

- - 老爸喜欢下象棋, 如果特意去下载一个象棋App
- - 父母可能自己都找不到这种App在哪里下载
- - 如果这个象棋是单机版的, 那么只能自己玩, 没有什么意思.
- - 如果这个象棋是联网的游戏, 那么父母必然需要注册
- - 登录 (对它们来说又增加了很大的难度)
- - 但是我们父母手机上必然有微信, 微信中包含小程序, 只需要下拉微信, 打开象棋的小程序, 就可以联机玩了.



# 小程序的出现到底带来了什么呢?

## ■ 2. 远好于公众号和H5页面的体验

- 比如寄顺丰快递, 使用公众号和小程序的体验差异非常大;

## ■ 3. 释放手机内存空间

- 对于手机空间不大的人来说, 安装很多App会带来灾难, 经常面临空间不足;
- 小程序不需要安装, 而且有规定的大小(目前不能超过8M, 包括在使用分包的情况下);

## ■ 4. 让手机桌面更加简洁

- 不需要下载App, 就不需要花时间来管理自己的App了;





# 小程序开发的必要性

- 由于小程序各种好处，使用小程序的 **用户会越来越多**
- 那么从公司角度，有自己的小程序就变得非常有必要
- 一方面，为了给用户带来更加便捷的使用体验，需要开发对应一款小程序：
  - 开发出来的小程序，用户体验优于H5的体验，并且底层可以调用原生的各种接口；
  - 而且可以做到：一端开发，多端运行（iOS端、Android端）；
- 另一方面，必须抢占小程序的市场：
  - 你不开发，你竞争对手也会开发，当大量用户为了方便涌入你的竞争对手时，你就来不及了；
- 所以现在公司经常出现，让前端或客户端程序员，甚至是服务端程序员去帮公司做一个小程序。
- 前面我们是从 **市场层面** 探讨小程序的特点
- 作为开发者，我们再从 **技术的角度** 探讨小程序的特点

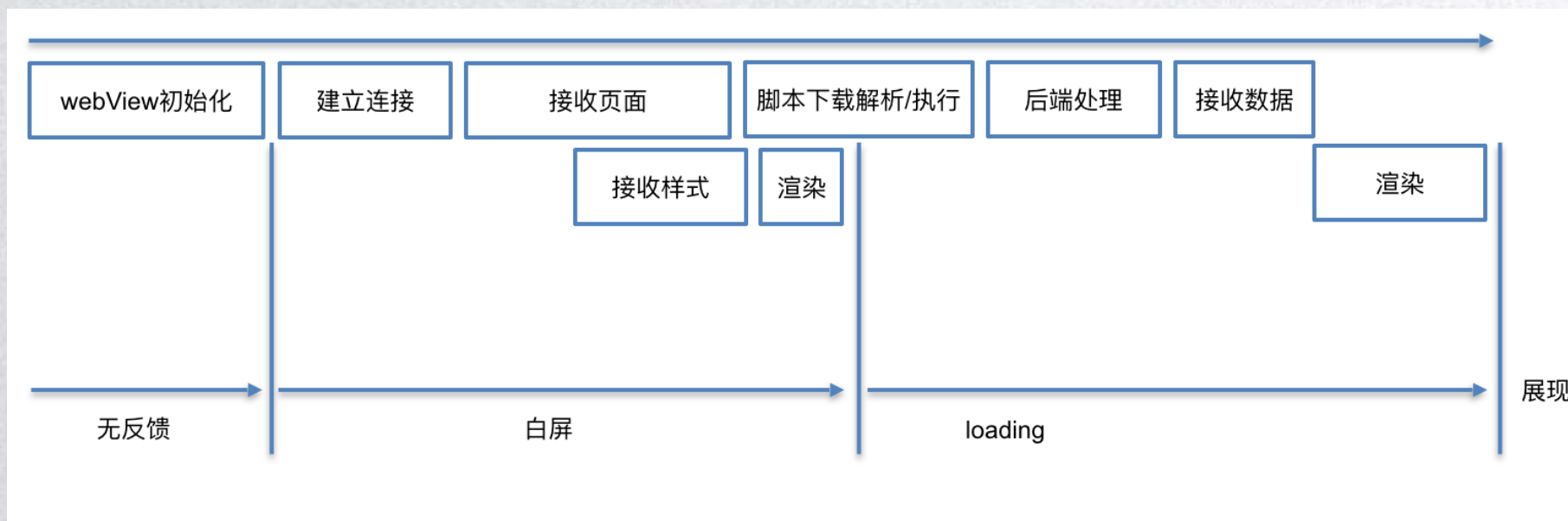




# WebView的发展和不足

## ■ WebView的飞速发展：

- 随着公众号的出现和繁荣，WebView的使用频率越来越高，不少的企业或个人都需要做H5页面。
- H5页面的劣势很明显：**体验太渣**，而且**无法调用很多底层App拥有的功能**。
- 微信团队也因此提供了一些JS-SDK给Web开发者使用，包括拍照、音频、地图、支付、分享等能力。
- 但是依然没法改变体验很渣这一点（比如加载某个页面经常是白屏、交互无反馈、页面切换生硬等）





# Native的优势和劣势

- 很长一段时间，我们认为解决这种白屏、交互体验差、页面切换生硬这种问题，都必须靠原生
  - （或者类似于Weex、React Native这种开发方式，最终呈现出接近原生应用）
- 但是Native也有它很多的弊端：
  - 开发门槛高、多端开发或者适配、版本更新需要审核、版本迭代慢
- 我们希望综合他们各自的优势，出现一种新的解决方案：

-	Native	Web	期望
开发门槛	高	低	低
体验	好	白屏、交互反馈差	接近原生体验
版本更新	需审核，迭代慢	在线更新	在线更新
管控性	平台可管控	难管控	可管控





# 小程序的诞生

- 我们知道，任何新的技术或者模式的产生，都是为了解决行业的痛点：小程序就是在这样的背景下诞生的
- **小程序正是大家期盼已久的产物，它有什么特点呢：**
  - 类似于Web 开发模式，入门的门槛低：基本上是类似于html+css+js；
  - 可直接云端更新：微信审核，无需经过App Store等平台；
  - 提升用户体验：通过提供基础能力、原生组件结合等方式，提升用户体验；
  - 平台管控能力：小程序提供云端更新，通过代码上传、审核等方式，增强对开发者的管控能力；
  - 双线程模型：逻辑层和渲染层分开加载，提供了管控型和安全性（沙盒环境运行JS代码，不允许执行任何和浏览器相关的接口，比如跳转页面、操作DOM等）；
- 目前小程序已经在用户和开发者中普及起来，并且依附于微信平台大量的人群基数，正在快速的发展



# 小程序的历史

- 2016年9月21日发布的小程序内测版
- 2017年1月9日正式上线
- <https://developers.weixin.qq.com/miniprogram/dev/framework/release/>
- 截止目前，小程序的技术和生态都相对非常成熟，完全是学习小程序的好时机。
- 那么，学习小程序需要哪里预备知识呢？





# 小程序预备知识

## ■ 小程序的开发主要分成三部分：

- - 页面布局：WXML，类似HTML
- - 页面样式：WXSS，几乎就是CSS（某些不支持，某些进行了增强）
- - 页面脚本：JavaScript+WXS，（JS，以及WeixinScript后续学习）

## ■ 也就是学习微信小程序还是需要大家具备 基本的前端知识

- 如果使用过一些MVVM架构的框架（Vue、React），那么更有利于对它的学习。

## ■ 前端开发者：直接学习即可。

## ■ 客户端、服务器开发者（iOS、Android、PC、WP？）：先学习HTML + CSS + JS的基础知识会更好一些（并非绝对，学习能力强的可以一边学习，一边补充）。

## ■ 编程零基础：可以学习，但是本课程不是面向 不懂任何编程 的人群，后续可能推出完全零基础课程。





# 开发前准备工作

开发前准备

申请AppID

下载开发工具





# 注册账号 – 申请AppID

■ 登录微信公众平台: <https://mp.weixin.qq.com/>

- 一个账号对应一个小程序;
- 企业、政府、媒体、其他组织主体可以注册50个小程序;
- 个体户和个人类型主体可注册5个小程序;

请选择注册的帐号类型

 <b>订阅号</b> 具有信息发布与传播的能力 适合个人及媒体注册	 <b>服务号</b> 具有用户管理与提供业务服务的能力 适合企业及组织注册
 <b>小程序</b> 具有出色的体验, 可以被便捷地获取与传播 适合有服务内容企业和组织注册	 <b>企业微信</b> 原企业号 具有实现企业内部沟通与协同管理的能力 适合企业客户注册

立即注册 | 简体中文 ▾

每个邮箱仅能申请一个小程序

邮箱   
作为登录帐号, 请填写未被微信公众平台注册, 未被微信开放平台注册, 未被个人微信号绑定的邮箱

密码   
字母、数字或者英文符号, 最短8位, 区分大小写

确认密码   
请再次输入密码

验证码   换一张

☐ 你已阅读并同意《微信公众平台服务协议》及《微信小程序平台服务条款》

注册



# 注册成功 – 查看AppID – 查看文档

微信公众平台 | 小程序

文档社区工具🔔👤

查看文档

🏠 首页

📁 管理

版本管理

成员管理

用户反馈

📊 统计

🛠 功能

客服

模板消息

小程序评测

</> 开发

开发

运维中心开发设置开发者工具接口设置

开发者ID

开发者ID

AppID(小程序ID)wx44a8878d2bb8c7d8

AppSecret(小程序密钥)

生成

服务器域名





# 小程序开发工具

■ 开发工具的选择：官方微信Web开发者工具、VSCode

■ 下载开发工具：

□ <https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>



**稳定版 Stable Build (1.02.1904090)**

测试版缺陷收敛后转为稳定版；

Windows 64、Windows 32、macOS



# 创建第一个小程序应用

小程序项目

小程序

小游戏

代码片段

公众号网页项目

公众号网页

 注销 >

新建项目 导入项目

项目名称

MiniProgram2

目录

C:\Users\home\Desktop\微信小程序\代码\MiniProgram2

AppID

wx44a8878d2bb8c7d8

若无 AppID 可 [注册](#)  
或使用 [测试号](#)

开发模式

小程序

后端服务

☒ 不使用云服务

☐ 小程序·云开发

小程序·云开发为开发者提供数据库、存储和云函数等完整的云端支持。无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现小程序快速上线和迭代。 [了解详情](#)

☐ 腾讯云

语言

JavaScript

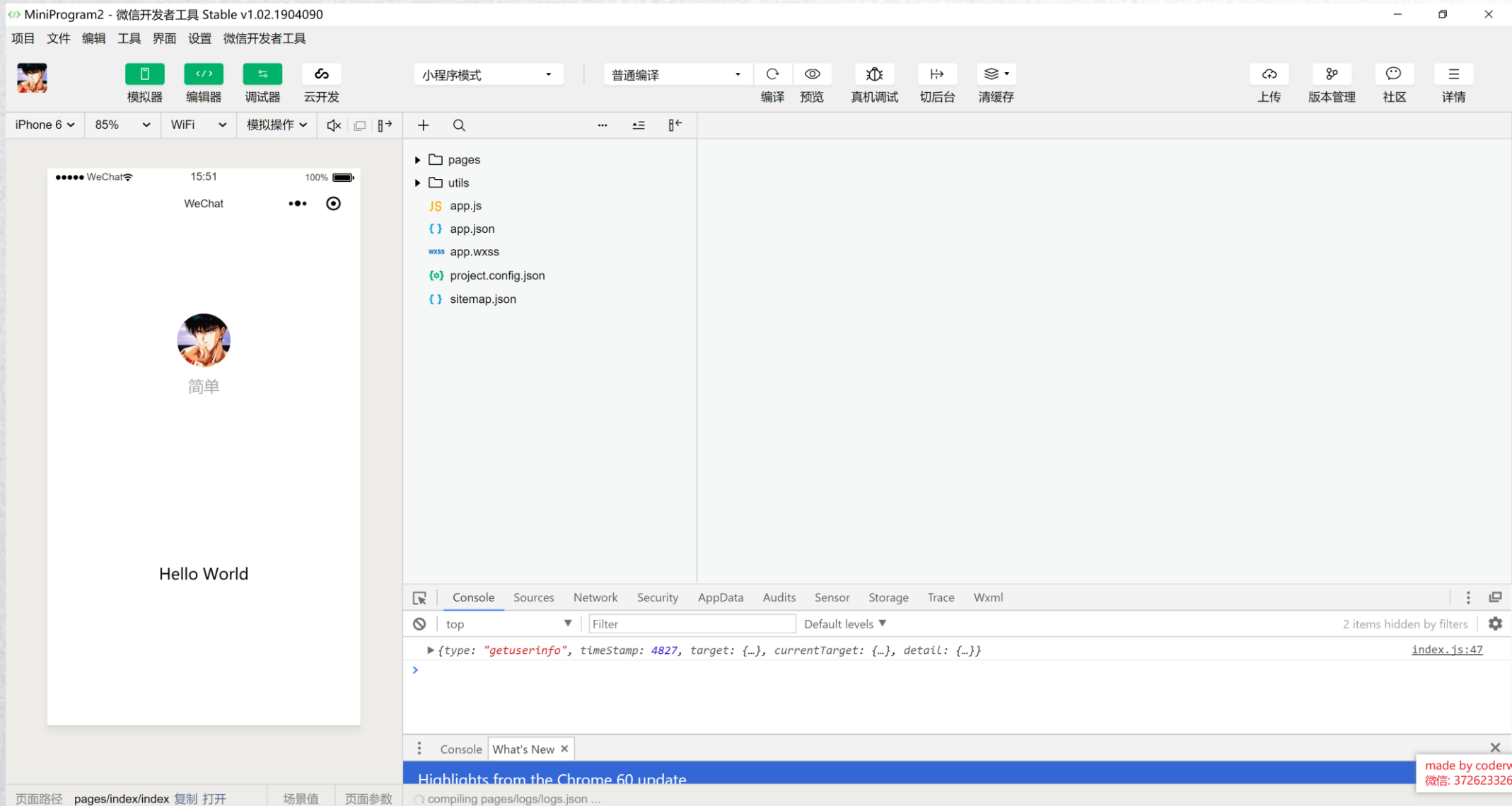
取消

新建





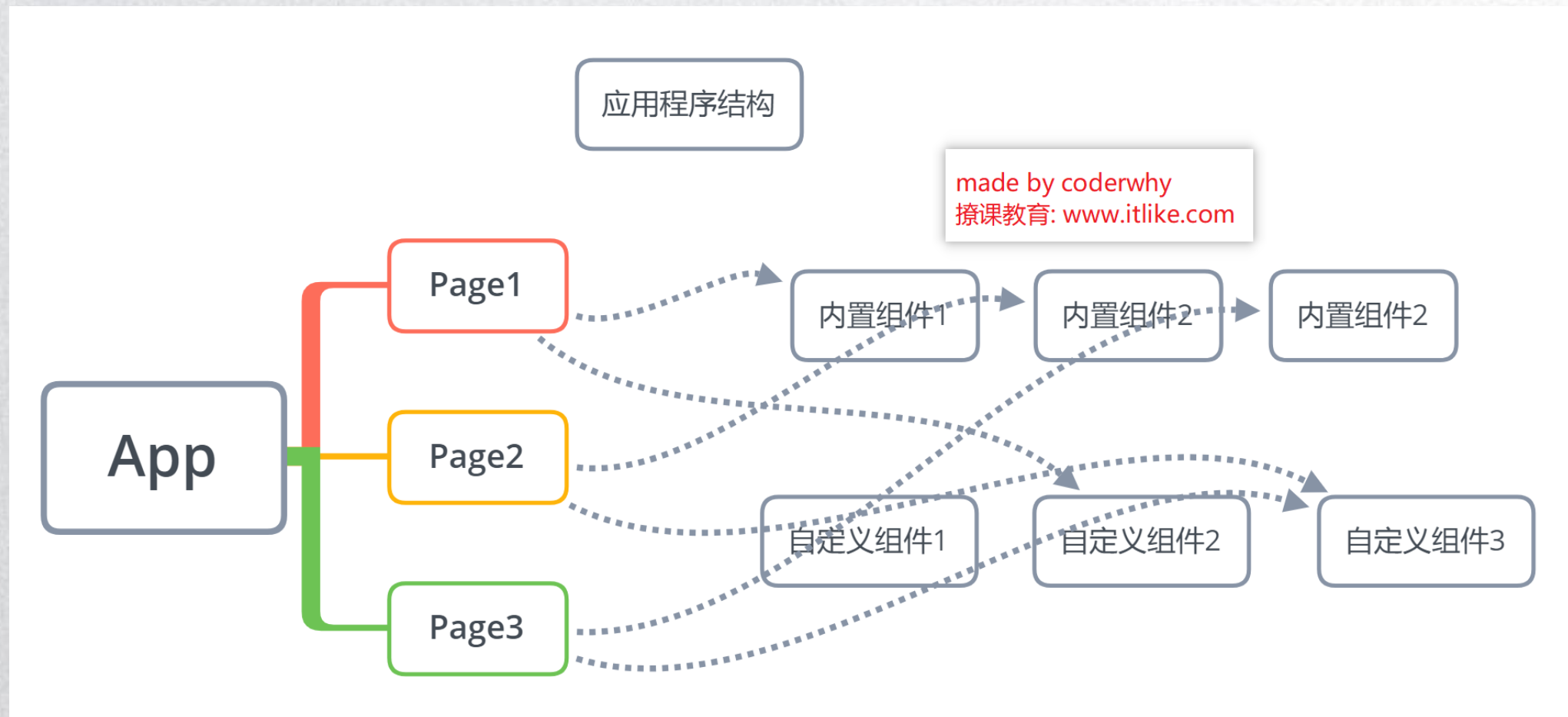
# 开发工具介绍





# 应用程序的结构

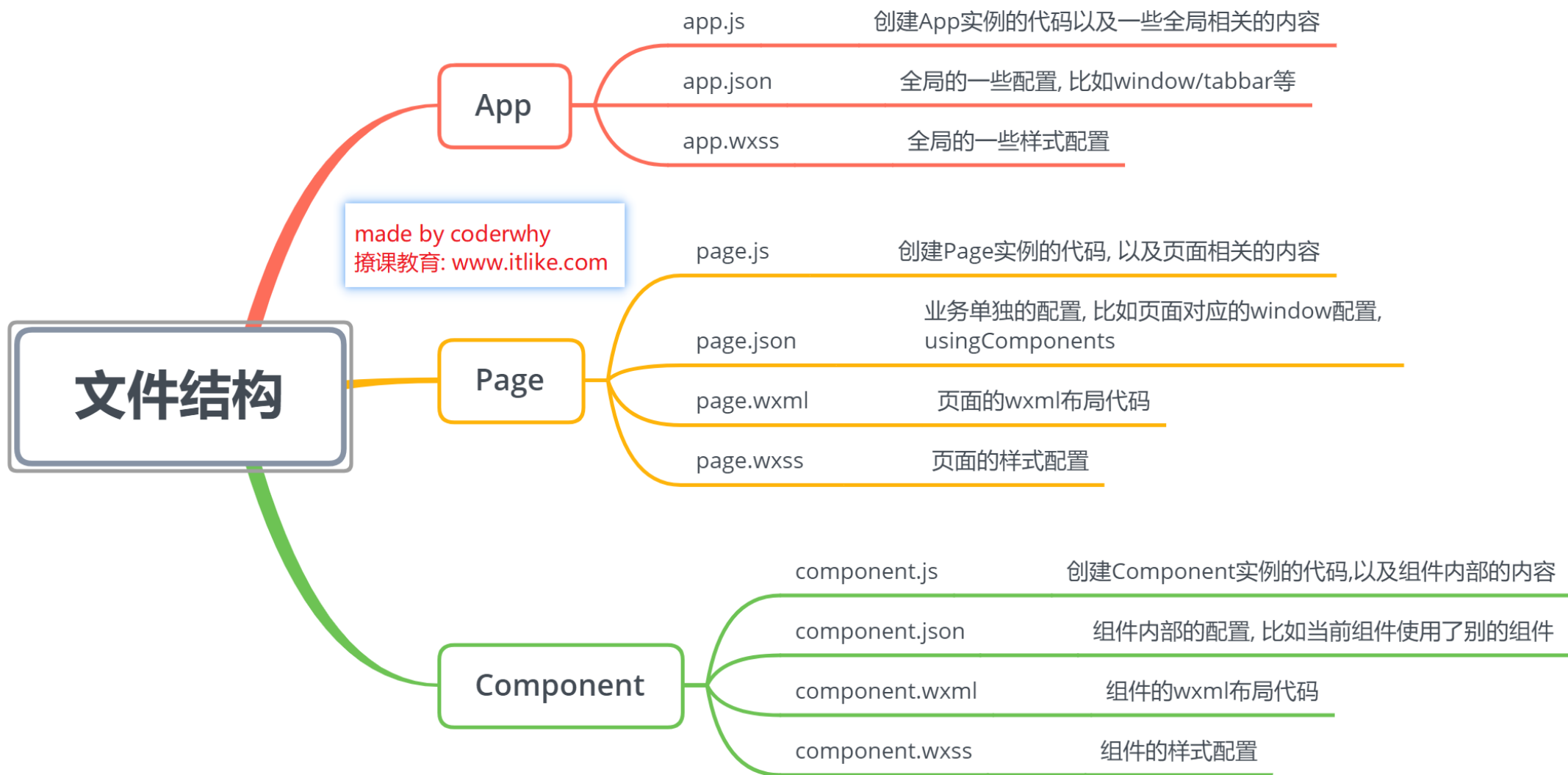
- 小程序结构划分：最上层App -> 多个Page -> 多个组件







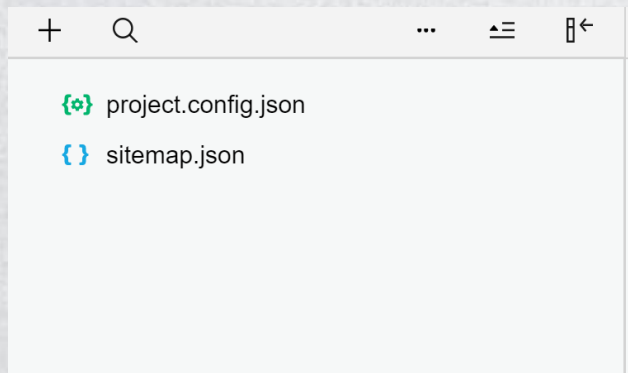
# 应用目录的结构





# 从零搭建项目(一)

- 删除项目中原有的文件(保留project.config.js – sitemap.json)



- 找不到app.json, 所以创建一个app.json文件

▼ Sun May 05 2019 16:06:00 GMT+0800 (中国标准时间) app.json 文件读取错误

✖ ▶ 未找到入口 app.json 文件, 或者文件读取失败, 请检查后重新编译。

- 不能正确的解析:

▼ Sun May 05 2019 16:08:23 GMT+0800 (中国标准时间) app.json 文件解析错误

✖ ▶ undefined  
Expecting 'STRING', 'NUMBER', 'NULL', 'TRUE', 'FALSE', '{', '[', got EOF  
1 |





# 从零搭建项目(二)

- 在app.json最外层添加对象:

```
app.json  ×
1  {
2
3  }
```

- 缺少一个字段pages, 并且是一个array

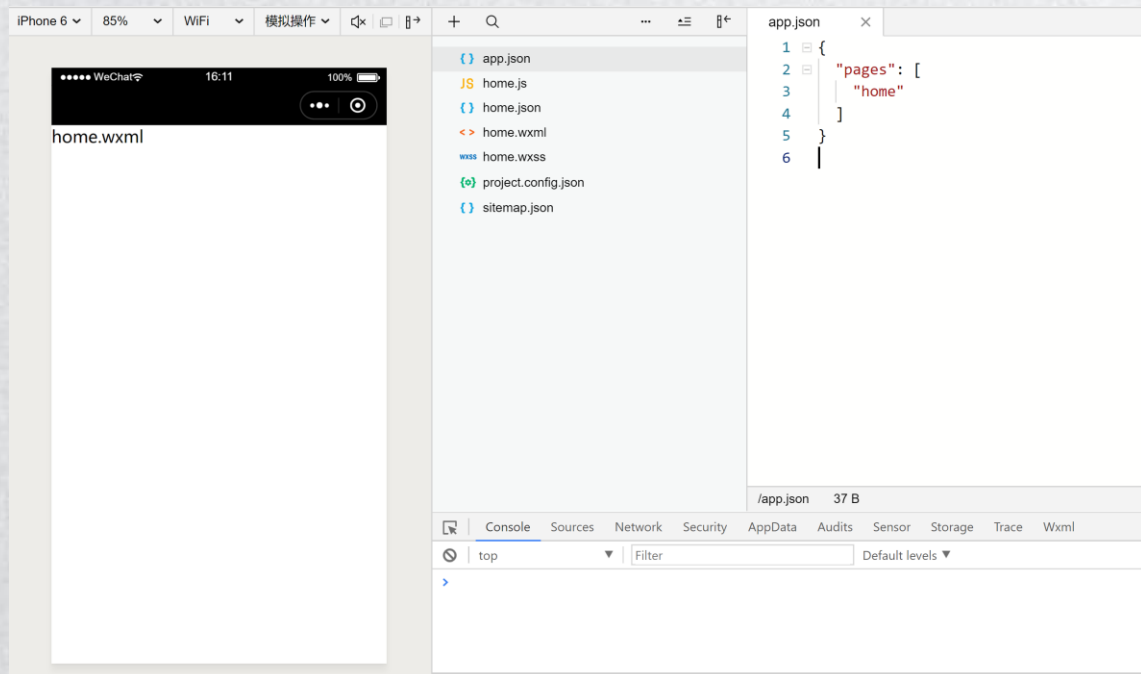
```
▼ Sun May 05 2019 16:09:19 GMT+0800 (中国标准时间) app.json 文件内容错误
✖ ▶ appJSON["pages"] 字段需为 array
>
```

```
app.json  ●
1  {
2    "pages": [
3      "home"
4    ]
5  }
```



# 从零搭建项目(三)

- 保存上面的操作, 就会看到如下页面:



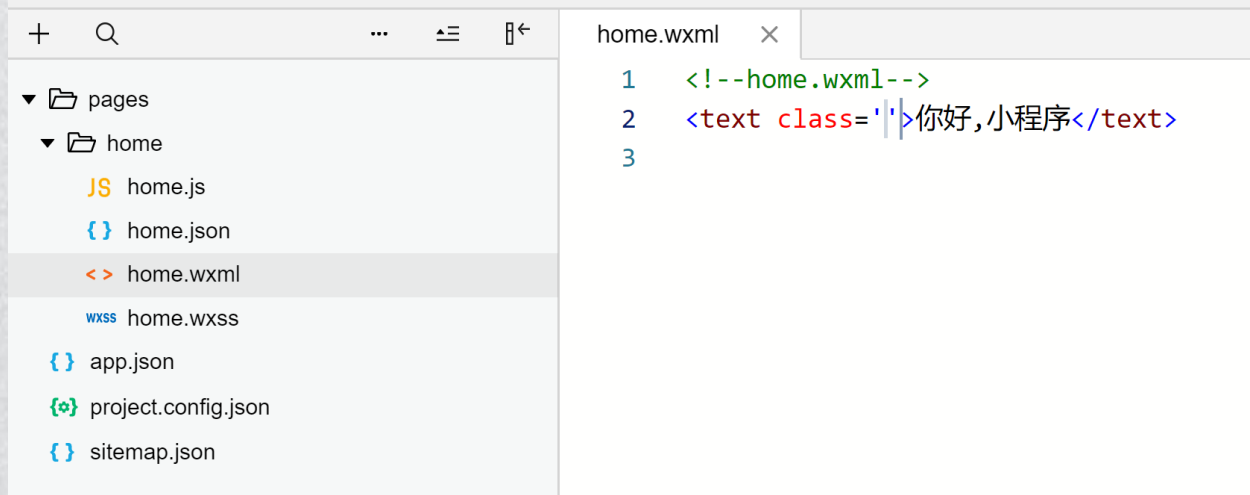
- 创建pages文件夹, 里面创建home文件夹, 将home相关的内容放在home文件夹中



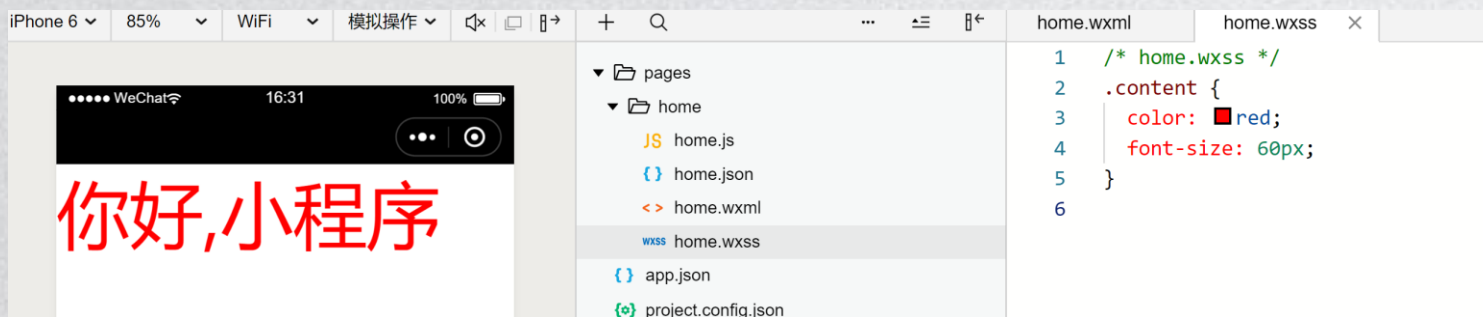


# 从零搭建项目(四)

## ■ 修改代码,显示 你好 小程序



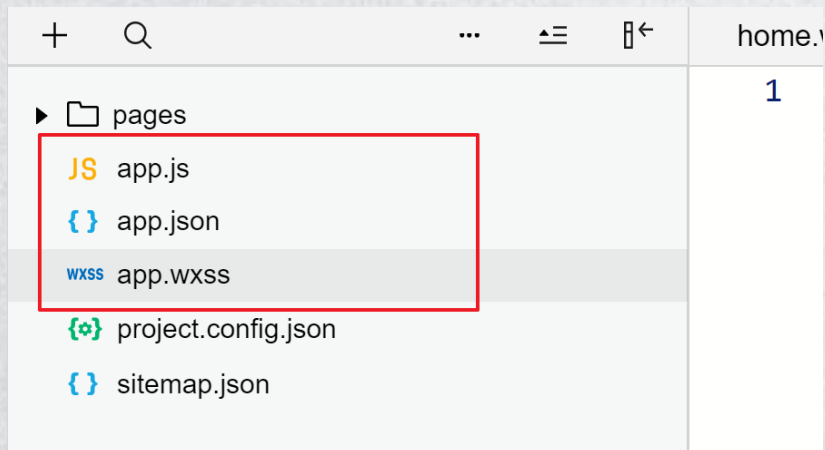
## ■ 修改样式:





# 从零搭建项目(五)

## ■ 补齐app.js和app.wxss:





# 多个项目管理方式

项目1文件夹

项目2文件夹

项目3文件夹

## 分支管理

git checkout -b 分支1

分支1(知识点演练)

git checkout -b 分支2

分支2(知识点演练)

git checkout -b 分支3

分支3(知识点演练)

## Tag管理

git tag 01\_知识点1

知识点演练

git tag 02\_知识点2

知识点演练

git tag 03\_知识点3

知识点演练





# 将项目加入git管理

## ■ 在GitHub上创建远程仓库

## ■ 本地初始化空仓库

- git init

- git add .

可以在这个步骤git pull

- git commit -m '初始化项目'

## ■ 本地和远程仓库相互关联:

- git remote add origin 远程仓库地址

- git push origin master

## ■ 创建和管理tag代码:

- git tag tag名称

- git push --tags

## ■ 额外补充:

- 1.补充一: 如果远程仓库为为仓库, 那么需要在push前进行如下命令:

- git pull origin master --allow-unrelated-histories

- 注意: 这里如果不加--allow-unrelated-histories, 会报错: fatal: refusing to merge unrelated histories

- 2.补充二: 从其他分支切换回来,文件夹没有被删除

- git clean -df



# 小程序初体验

## ■ 小程序初体验

- ❑ 1.数据绑定
- ❑ 2.列表渲染
- ❑ 3.事件监听



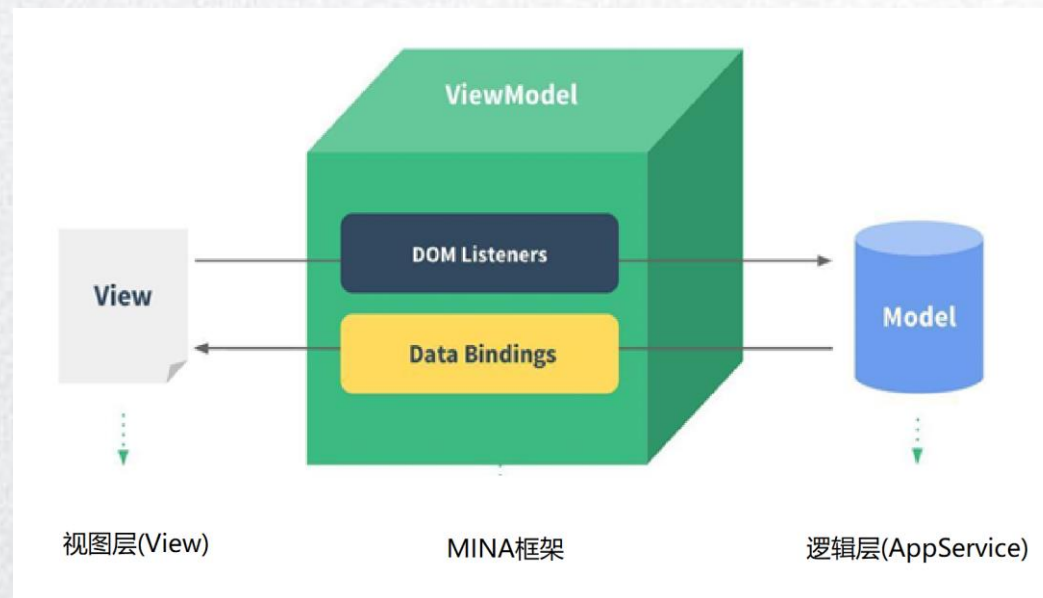
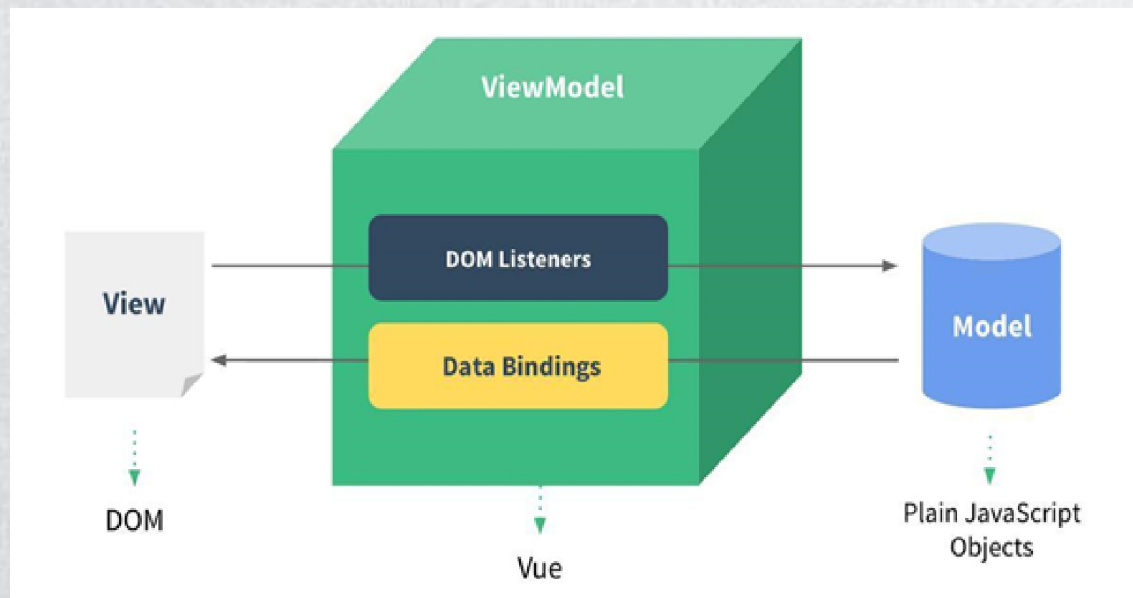
```
home.wxml  ×  home.js
1  <!-- 1.数据展示 -->
2  <view>Hello {{content}}</view>
3
4  <!-- 2.列表展示 -->
5  <view wx:for="{{movies}}" wx:key="index">{{item}}</view>
6
7  <!-- 3.修改数据 -->
8  <view>当前计数:{{counter}}</view>
9  <button size='mini' bind:tap="increment">+</button>
10 <button size='mini' bind:tap="decrement">-</button>
11
```

```
wxml  home.js  ×
Page({
  data: {
    content: "coderwhy",
    movies: ["大话西游", "盗梦空间", "星际穿越", "少年派"],
    counter: 0
  },
  onChangeTap() {
    this.setData({
      content: "kobe"
    })
  },
  increment() {
    this.setData({
      counter: ++this.data.counter
    })
  },
  decrement() {
    this.setData({
      counter: --this.data.counter
    })
  }
})
```



# 小程序的MVVM架构

## ■ Vue的MVVM和小程序MVVM对比



## ■ MVVM为什么好用呢?

- DOM Listeners: ViewModel层可以将DOM的监听绑定到Model层
- Data Bindings: ViewModel层可以将数据的变量, 响应式的反应到View层

## ■ MVVM架构将我们从 命令式编程 转移到 声明式编程





# 命令式编程 – 声明式编程

```
<body>
  <h2 class="title"></h2>
  <button class="btn">按钮</button>
  <input type="date">

  <script>
    // 1. 定义变量
    let name = "哈哈"

    // 2. 通过class获取元素
    const titleEl = document.querySelector('.title')

    // 3. 设置内容的显示
    titleEl.textContent = name;

    // 4. 监听按钮的点击
    const btnEl = document.querySelector('.btn')

    // 5. 监听按钮的点击
    btnEl.addEventListener('click', () => {
      name = '呵呵'
      titleEl.textContent = name;
    })
  </script>
</body>
```

```
<body>
  <div id="app">
    <h2>{{title}}</h2>
    <button @click="btnClick">按钮</button>
  </div>

  <script src="vue.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: {
        title: '哈哈'
      },
      methods: {
        btnClick() {
          this.title = '呵呵'
        }
      }
    })
  </script>
</body>
```