**1. 简单描述下微信小程序的相关文件类型**

微信小程序项目结构主要有四个文件类型

* WXML（WeiXin Markup Language）是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。内部主要是微信自己定义的一套组件
* WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式
* js 逻辑处理，网络请求
* json 小程序设置，如页面注册，页面标题及tabBar

主要文件

* app.json 必须要有这个文件，如果没有这个文件，项目无法运行，因为微信框架把这个作为配置文件入口，整个小程序的全局配置。包括页面注册，网络设置，以及小程序的 window 背景色，配置导航条样式，配置默认标题
* app.js 必须要有这个文件，没有也是会报错！但是这个文件创建一下就行 什么都不需要写以后我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量
* app.wxss 可选

**2. 简述微信小程序原理**

微信小程序采用 JavaScript、WXML、WXSS 三种技术进行开发,本质就是一个单页面应用，所有的页面渲染和事件处理，都在一个页面内进行，但又可以通过微信客户端调用原生的各种接口

微信的架构，是数据驱动的架构模式，它的 UI 和数据是分离的，所有的页面更新，都需要通过对数据的更改来实现

小程序分为两个部分 webview 和 appService 。其中 webview 主要用来展现 UI ，appService 有来处理业务逻辑、数据及接口调用。它们在两个进程中运行，通过系统层 JSBridge 实现通信，实现 UI 的渲染、事件的处理

**3. 小程序的双向绑定和vue哪里不一样**

小程序直接 this.data 的属性是不可以同步到视图的，必须调用：

this.setData({

// 这里设置

})

**4. 小程序的wxss和css有哪些不一样的地方**

WXSS 和 CSS 类似，不过在 CSS 的基础上做了一些补充和修改

* 尺寸单位 rpx

rpx 是响应式像素,可以根据屏幕宽度进行自适应。规定屏幕宽为 750rpx。如在 iPhone6 上，屏幕宽度为 375px，共有 750 个物理像素，则 750rpx = 375px = 750 物理像素

* 使用 @import 标识符来导入外联样式。@import 后跟需要导入的外联样式表的相对路径，用;表示语句结束

/\*\* index.wxss \*\*/

@import './base.wxss';

.container{

color: red;

}

**5. 小程序页面间有哪些传递数据的方法**

* 使用全局变量实现数据传递 在 app.js 文件中定义全局变量 globalData， 将需要存储的信息存放在里面

// app.js

App({

// 全局变量

globalData: {

userInfo: null

}

})

使用的时候，直接使用 getApp() 拿到存储的信息

* 使用 wx.navigateTo 与 wx.redirectTo 的时候，可以将部分数据放在 url 里面，并在新页面 onLoad 的时候初始化

//pageA.js

// Navigate

wx.navigateTo({

url: '../pageD/pageD?name=raymond&gender=male',

})

// Redirect

wx.redirectTo({

url: '../pageD/pageD?name=raymond&gender=male',

})

// pageB.js

...

Page({

onLoad: function(option){

console.log(option.name + 'is' + option.gender)

this.setData({

option: option

})

}

})

需要注意的问题：

wx.navigateTo 和 wx.redirectTo 不允许跳转到 tab 所包含的页面

onLoad 只执行一次

* 使用本地缓存 Storage 相关

**6. 小程序的生命周期函数（详见**[**生命周期回调函数**](https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/page.html#%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F%E5%9B%9E%E8%B0%83%E5%87%BD%E6%95%B0)**）**

* onLoad 页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数
* onShow() 页面显示/切入前台时触发
* onReady() 页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互
* onHide() 页面隐藏/切入后台时触发。 如 navigateTo 或底部 tab 切换到其他页面，小程序切入后台等
* onUnload() 页面卸载时触发。如 redirectTo 或 navigateBack 到其他页面时

**7. 怎么封装微信小程序的数据请求（参考**[**这里**](https://segmentfault.com/a/1190000014789969)**）**

可以用Promise封装

**8. 哪些方法可以用来提高微信小程序的应用速度**

1、提高页面加载速度

2、用户行为预测

3、减少默认 data 的大小

4、组件化方案

**9. 微信小程序的优劣势**

优势

* 即用即走，不用安装，省流量，省安装时间，不占用桌面
* 依托微信流量，天生推广传播优势
* 开发成本比 App 低

缺点

* 用户留存，即用即走是优势，也存在一些问题
* 入口相对传统 App 要深很多
* 限制较多,页面大小不能超过1M。不能打开超过5个层级的页面

**10. 怎么解决小程序的异步请求问题**

小程序支持 ES6 语法

* 在返回成功的回调里面处理逻辑
* Promise 异步
* async/await

**11. 小程序关联微信公众号如何确定用户的唯一性**

如果开发者拥有多个移动应用、网站应用、和公众帐号（包括小程序），可通过 unionid 来区分用户的唯一性，因为只要是同一个微信开放平台帐号下的移动应用、网站应用和公众帐号（包括小程序），用户的 unionid 是唯一的。换句话说，同一用户，对同一个微信开放平台下的不同应用，unionid 是相同的

**12. 如何实现下拉刷新（参考**[**这里**](https://juejin.im/post/5a781c756fb9a063606eb742)**）**

* 首先在全局 config 中的 window 配置 enablePullDownRefresh
* 在 Page 中定义 onPullDownRefresh 钩子函数,到达下拉刷新条件后，该钩子函数执行，发起请求方法
* 请求返回后，调用 wx.stopPullDownRefresh 停止下拉刷新

**13. bindtap和catchtap的区别是什么**

相同点：首先他们都是作为点击事件函数，就是点击时触发。在这个作用上他们是一样的，可以不做区分

不同点：他们的不同点主要是bindtap是不会阻止冒泡事件的，catchtap是阻值冒泡的

**14. 简述下 `wx.navigateTo()`, `wx.redirectTo()`, `wx.switchTab()`, `wx.navigateBack()`, `wx.reLaunch()`的区别**

* wx.navigateTo()：保留当前页面，跳转到应用内的某个页面。但是不能跳到 tabbar 页面
* wx.redirectTo()：关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabbar 页面
* wx.switchTab()：跳转到 abBar 页面，并关闭其他所有非 tabBar 页面
* wx.navigateBack()关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages() 获取当前的页面栈，决定需要返回几层
* wx.reLaunch()：关闭所有页面，打开到应用内的某个页面