# 1 基础篇

## 1.1 请谈谈微信小程序主要目录和文件的作用

* project.config.json 项目配置文件，用得最多的就是配置是否开启https校验；
* App.js 设置一些全局的基础数据等；
* App.json 底部tab, 标题栏和路由等设置；分包预加载
* App.wxss 公共样式，引入iconfont等；引入外部组件库
* pages 里面包含一个个具体的页面；
* index.json (配置当前页面标题和引入组件等)；是否支持下拉刷新，返回按钮
* index.wxml (页面结构)；
* index.wxss (页面样式表)；
* index.js (页面的逻辑，请求和数据处理等)；

## 1.2 请谈谈wxml与标准的html的异同

* 都是用来描述页面的结构；
* 都由标签、属性等构成；
* 标签名字不一样，且小程序标签更少，单一标签更多；
* 多了一些 wx:if 这样的属性以及 {{ }} 这样的表达式
* WXML仅能在微信小程序开发者工具中预览，而HTML可以在浏览器内预览
* 组件封装不同， WXML对组件进行了重新封装，
* 小程序运行在JS Core内，没有DOM树和window对象，小程序中无法使用window对象和document对象。

## 1.3请谈谈WXSS和CSS的异同

* 都是用来描述页面的样子；
* WXSS 具有 CSS 大部分的特性，也做了一些扩充和修改；
* WXSS新增了尺寸单位，WXSS 在底层支持新的尺寸单位 rpx；
* WXSS 仅支持部分 CSS 选择器；
* WXSS 提供全局样式与局部样式

## 1.4 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
* 页面跳转或重定向时，使用url带参数传递数据
* 使用组件模板 template传递参数
* 使用缓存传递参数
* 使用数据库传递数据

## 1.5 请谈谈小程序的双向绑定和vue的异同

大体相同，但小程序直接this.data的属性是不可以同步到视图的，必须调用this.setData()方法！

## 1.6 小程序的生命周期函数

* onLoad() 页面加载时触发，只会调用一次，可获取当前页面路径中的参数。
* onShow() 页面显示/切入前台时触发，一般用来发送数据请求；
* onReady() 页面初次渲染完成时触发, 只会调用一次，代表页面已可和视图层进行交互。
* onHide() 页面隐藏/切入后台时触发, 如底部 tab 切换到其他页面或小程序切入后台等。
* onUnload() 页面卸载时触发，如redirectTo或navigateBack到其他页面时。

## 1.7 微信小程序原理

* 小程序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；
* 它的架构，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；
* 它从技术讲和现有的前端开发差不多，采用JavaScript、WXML、WXSS三种技术进行开发；
* 功能可分为webview和appService两个部分；
* webview用来展现UI，appService有来处理业务逻辑、数据及接口调用；
* 两个部分在两个进程中运行，通过系统层JSBridge实现通信，实现UI的渲染、事件的处理等

JavaScript：首先JavaScript的代码是运行在微信App中的，并不是运行在浏览器中，因此一些H5技术的应用，需要微信App提供对应的API支持，而这限制住了H5技术的应用，且其不能称为严格的H5，可以称其为伪H5，同理，微信提供的独有的某些API，H5也不支持或支持的不是特别好。  
WXML：WXML微信自己基于XML语法开发的，因此开发时，只能使用微信提供的现有标签，HTML的标签是无法使用的。  
  
WXSS：WXSS具有CSS的大部分特性，但并不是所有的都支持，而且支持哪些，不支持哪些并没有详细的文档。  
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。

## 1.8 页面跳转函数的区别

* 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

## 1.9 数据请求的封装

* 将所有的接口放在统一的js文件中并导出（或者将请求地址、头、方法在一个js文件里统一定义为一个常量并导出）
* 在app.js创建封装请求数据的方法
* 在子页面中调用封装的方法请求数据

## 1.10 小程序的优缺点

##### ****小程序的优点****

* 无需下载
* 打开速度快
* 开发成本低
* 为用户提供良好的安全保障。发布有一套严格的审查流程，不能通过审查的程序无法发布上线
* 服务请求快

##### ****小程序的缺点****

* 依托微信，不能开发后台管理功能
* 大小限制不能超过2M，不能打开超过5个层级的页面