#### **1、微信的小程序的主要文件**

* WXML——模板文件
* JSON——配置/设置文件，如标题,tabbar,页面注册
* WXSS——样式文件，样式可直接用import导入
* JS——脚本逻辑文件，逻辑处理，网络请求
* app.json——配置文件入口，整个小程序的全局配置，网络超时时间、底部tab、页面路径，window字段是小程序所有页面的顶部背景颜色、文字颜色
* app.js——可以没有内容，可以在里边监听生命周期函数、声明全局变量
* app.wxss——全局配置样式文件

#### **2、数据请求怎么封装**

* 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

#### **3、参数传递**

* 给HTML元素中添加data-\*属性来传递需要的值，之后通过e.currentTarget.dataset或onload的param参数获取。注意不能有大写字母，不可以存放对象
* 跳转页面时通过navigator传递需要的参数值
* 设置id的方法标识，通过e.currentTarget.id获取设置的id值，然后通过设置全局变量的方法来传递数值

#### ****4、生命周期函数****

* onLoad——页面加载，调一次
* onShow——页面显示，每次打开页面都调用
* onReady——初次渲染完成，调一次
* onHide——页面隐藏，当navigateTo或底部tab切换时调用
* onUnload——页面卸载，当redirectTo或navigateBack时调用

#### ****5、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({})

#### ****6、1px = 2rpx****

#### ****7、如何自定义组件**（弹窗）**

* 先创建一个components文件夹，用来存放所有自定义组件的，目录结构依然是js,wxml,json,wxss

基本配置：

* .json——进行自定义组件声明

{

"component": true

}

使用组件：

* 假如在index.wxml中使用这个自定义的组件，首先在index.json中进行声明

{

"usingComponents": {

"toastdemo": "/components/toastdemo/toastdemo"

}

}

* 接着在index.wxml中引用
* 然后在index.js进行配置
* 使用时直接执行this.toastdemo.showToast('弹框组件调用成功',2000)就可以了

#### ****8、小程序内的页面跳转****

* 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（参数必须为字符串）
* 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 tabBar 页面，并关闭其他所有非 tabBar 页面，路径后不能带参数
* 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

#### ****9、小程序和Vue写法的区别****

* 循环遍历：小程序是wx:for="list"，vue是v-for="inforin list"
* 调用data模型：小程序是this.data.unifo，vue是this.unifo
* 给模型赋值：小程序是this.setData({unifo:1})，vue是直接this.unifo=1

#### ****10、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({})

#### ****11、小程序的优点和缺点****

**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

#### ****12、简述小程序原理****

* 小程序分为两个部分webview和appService，webview用来展现UI，appService用来处理业务逻辑、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，完成UI渲染、事件处理。

#### ****13、提高小程序的应用速度的方法****

* 减少默认data的大小
* 组件化方案，公用的如弹框等写个自定义的组件，然后调用

#### ****14、简述小程序原理****

* 小程序分为两个部分webview和appService，webview用来展现UI，appService用来处理业务逻辑、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，完成UI渲染、事件处理

#### ****15、setData的回调函数****

微信小程序的setData实现是和react的setData实现类似的，所以它也是一个异步函数，并且有回调函数的参数，当然平时小量数据我们可能并没有感觉到它的异步，但是为了确保逻辑的正确执行，在需要用到setData后 data里的数据的步骤，请写入setData的回调函数中，如下示例：

this.setData({

a: this.data.a++

},()=>{

})

#### ****16、如何实现下拉刷新****

* 先在app.json或page.json中配置enablePullDownRefresh:true
* page里用onPullDownRefresh函数，在下拉刷新时执行
* 在下拉函数执行时发起数据请求，请求返回后，调用wx.stopPullDownRefresh停止下拉刷新的状态

#### ****17、bindtap和catchtap的区别是什么****

* bindtap不会阻止冒泡事件，catchtap阻止冒泡

#### **18、微信小程序与H5的区别？**

①运行环境不同（小程序在微信运行，h5在浏览器运行）；

②开发成本不同（h5需要兼容不同的浏览器）；

③获取系统权限不同（系统级权限可以和小程序无缝衔接）；

④应用在生产环境的运行流畅度（h5需不断对项目优化来提高用户体验）；

#### **19、小程序关联微信公众号如何确定用户的唯一性？**

使用wx.getUserInfo方法 withCredentials为true时，可获取encryptedData，里面有union\_id.后端需要进行对称解密。

#### ****20、webview中的页面怎么跳回小程序中****

* 先在管理后台配置域名白名单，
* 然后引入jweixin-1.3.2.js（[https://res.wx.qq.com/open/js/jweixin-1.3.0.js）](https://link.zhihu.com/?target=https://res.wx.qq.com/open/js/jweixin-1.3.0.js%EF%BC%89" \t "/Users/zouming/Documents\\x/_blank)

wx.miniProgram.navigateTo({url: '/pages/login/login'+'$params'})

wx.miniProgram.navigateTo({url: '/path/to/page'})

#### ****21、webview的页面怎么跳转到小程序导航的页面？****

小程序导航的页面可以通过switchTab，但默认情况是不会重新加载数据的。若需加载新数据，则在success属性中加入以下代码即可：

success: function (e) {

var page = getCurrentPages().pop();

if (page == undefined || page == null) return;

page.onLoad();

}

//webview的页面，则通过

wx.miniProgram.switchTab({

url: '/pages/index/index'

})

#### ****22、小程序和小程序之间的跳转****

* 在同一主体公众号上关联2个小程序appid，
* 用**navigator**，对应设置一些属性即可
* **target：miniProgram——其他小程序**
* target：self——当前小程序
* 简单描述下微信小程序的相关文件类型？
* 一、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。与html差不多。
* 二、WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式，与css差不多
* 二、js 逻辑处理，网络请求
* 三、json 小程序设置，如页面注册，页面标题及tabBar。
* 有哪些参数传值的方法？
* 一、给HTML元素添加data-\*属性来传递我们需要的值，然后通过e.currentTarget.dataset或onload的param参数获取。但data-名称不能有大写字母和不可以存放对象
* 二、设置id 的方法标识来传值通过e.currentTarget.id获取设置的id的值,然后通过设置全局对象的方式来传递数值
* 三、在navigator中添加参数传值（？传的值的名称=所传的值在onLoad(option)用option来接收并获取）
* 3、你使用过哪些方法，来提高微信小程序的应用速度？
* 一、提高页面加载速度
* 二、用户行为预测
* 三、减少默认data的大小
* 四、组件化方案
* 4、小程序与原生App哪个好？
* 都各有各的好处，都有又缺点。
* 小程序的优点：
* 基于微信平台开发，享受微信本身自带的流量，这个是最大的优势
* 无需安装，只要打开微信就能用，不占用用户手机内存，体验好
* 开发周期短，一般最多一个月可以上线完成
* 开发所需的资金少，所需资金是开发原生APP一半不到
* 小程序名称是唯一性的，在微信的搜索里权重很高
* 容易上手，只要之前有HTML+CSS+JS基础知识，写小程序基本上没有大问题；当然如果了解ES6+CSS3则完全可以编写出即精简又动感的小程序；
* 基本上不需要考虑兼容性问题，只要微信可以正常运行的机器，就可以运行小程序；
* 发布、审核高效，基本上上午发布审核，下午就审核通过，升级简单，而且支持灰度发布；
* 开发文档比较完善，开发社区比较活跃；
* 最近刚开放的牛x功能，新增webview组件，可以展示网页啦，这个比较爽；
* 支持插件式开发，一些基本功能可以开发成插件，供多个小程序调用；
* 缺点：
* 1.局限性很强，（比如页面大小不能超过1M。不能打开超过5个层级的页面。样式单一。小程序的部分组件已经是成型的了，样式不可以修改。例如：幻灯片、导航。）只能依赖于微信依托于微信，无法开发后台管理功能。
* 2.不利于推广推广面窄，不能分享朋友圈，只能通过分享给朋友，附近小程序推广。其中附近小程序也受到微信的限制
* 3.后台调试麻烦，因为API接口必须https请求，且公网地址，也就是说后台代码必须发布到远程服务器上；当然我们可以修改host进行dns映射把远程服务器转到本地，或者开启tomcat远程调试；不管怎么说终归调试比较麻烦。
* 4.前台测试有诸多坑，最头疼莫过于模拟器与真机显示不一致
* 5.js引用只能使用绝对路径，很蛋疼；基于安全性及MINA框架实现原理，小程序中对js使用做了很多限制，不能使用：new Function，eval，Generator，不能操作cookie，不能操作DOM；
* 原生App优点：
* 1、原生的响应速度快
* 2、对于有无网络操作时，譬如离线操作基本选用原生开发
* 3、需要调用系统硬件的功能（摄像头、方向传感器、重力传感器、拨号、GPS、语音、短信、蓝牙等功能）
* 4、在无网络或者若网的情况下体验好。
* 缺点：
* 开发周期长，开发成本高
* 需要下载
* 简述微信小程序原理？
* 答：微信小程序采用JavaScript、WXML、WXSS三种技术进行开发，从技术讲和现有的前端开发差不多，但深入挖掘的话却又有所不同。
* 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。
* 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
* 分析下微信小程序的优劣势？
* 答：
* 优势：
* 1、无需下载，通过搜索和扫一扫就可以打开。
* 2、良好的用户体验：打开速度快。
* 3、开发成本要比App要低。
* 4、安卓上可以添加到桌面，与原生App差不多。
* 5、为用户提供良好的安全保障。小程序的发布，微信拥有一套严格的审查流程， 不能通过审查的小程序是无法发布到线上的。
* 劣势：
* 1、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。
* 2、样式单一。小程序的部分组件已经是成型的了，样式不可以修改。例如：幻灯片、导航。
* 3、推广面窄，不能分享朋友圈，只能通过分享给朋友，附近小程序推广。其中附近小程序也受到微信的限制。
* 4、依托于微信，无法开发后台管理功能。
* 小程序的发布流程（开发流程）
* 参考：https://www.cnblogs.com/ssrstm/p/6855572.html
* 注册微信小程序账号
* 获取微信小程序的 AppID
* 下载微信小程序开发者工具
* 创建demo项目
* 去微信公众平台配置域名
* 手机预览
* 代码上传
* 提交审核
* 小程序发布
* 8、webview中的页面怎么跳回小程序中？参考：https://blog.csdn.net/qq\_36742720/article/details/81501955
* 首先，需要在你的html页面中引用一个js文件。
* <script type="text/javascript" src="https://res.wx.qq.com/open/js/jweixin-1.3.0.js"></script>
* 然后为你的按钮标签注册一个点击事件：
* $(".kaiqi").click(function(){
* wx.miniProgram.redirectTo({url: '/pages/indexTwo/indexTwo'})
* });
* 这里的redirectTo跟小程序中的wx.redirectTo()跳转页面是一样的，会关闭当前页跳转到页面。
* 你也可以替换成navigateTo，跳转页面不会关闭当前页。
* 使用webview直接加载要注意哪些事项？参考：https://blog.csdn.net/rolan1993/article/details/78709463
* 答: 一、必须要在小程序后台使用管理员添加业务域名；
* 二、h5页面跳转至小程序的脚本必须是1.3.1以上；
* 三、微信分享只可以都是小程序的主名称了，如果要自定义分享的内容，需小程序版本在1.7.1以上；
* 四、h5的支付不可以是微信公众号的appid，必须是小程序的appid，而且用户的openid也必须是用户和小程序的。
* 10、小程序授权登录流程。
* （授权，微信登录获取code，微信登录，获取 iv , encryptedData 传到服务器后台，如果没有注册，需要注册。）
* （授权，微信登录获取code，微信登录，获取 iv , encryptedData 传到服务器后台，如果没有注册，需要注册。）
* 小程序支付如何实现？
* 我们做要简单，其实我更认为是小程序只有1M，更多的东西给后台吧
* 1、小程序注册，要以公司的以身份去注册一个小程序，才有微信支付权限；
* 2、绑定商户号。
* 3、在小程序填写合法域
* 4.调用wx.login()获取appid
* 5.调用
* wx.requestPayment(
* {
* 'timeStamp': '',//时间戳从1970年1月1日00:00:00至今的秒数,即当前的时间
* 'nonceStr': '',//随机字符串，长度为32个字符以下。
* 'package': '',//统一下单接口返回的 prepay\_id 参数值，提交格式如：prepay\_id=\*
* 'signType': 'MD5',//签名类型，默认为MD5，支持HMAC-SHA256和MD5。注意此处需与统一下单的签名类型一致
* 'paySign': '',//签名,具体签名方案参见微信公众号支付帮助文档;
* 'success':function(res){},//成功回调
* 'fail':function(res){},//失败
* 'complete':function(res){}//接口调用结束的回调函数（调用成功、失败都会执行）
* })
* ————————————————
* 版权声明：本文为CSDN博主「雨生百谷，方为谷雨」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。
* 原文链接：https://blog.csdn.net/weixin\_44058725/article/details/102695538