微信小程序基础面试题

1. 微信小程序有几个文件?

- 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 。
- 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 配置全局 css

2. 微信小程序怎样跟事件传值?

- 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。

3. 小程序的 wxss 和 css 有哪些不一样的地方?

- wxss 的图片引入需使用外链地址。
- 的没有 Body；样式可直接使用 import 导入

4. 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。

5. 微信小程序与 vue 区别?

- 生命周期不一样，微信小程序生命周期比较简单
- 数据绑定也不同，微信小程序数据绑定需要使用 {{}} ， vue 直接 : 就可以
- 显示与隐藏元素， vue 中，使用 v-if 和 v-show 控制元素的显示和隐藏，小程序中，使用 wx-if 和 hidden 控制元素的显示和隐藏
- 事件处理不同，小程序中，全用 bindtap(bind+event) ，或者 catchtap(catch+event)绑定事件, vue： 使用 v-on:event 绑定事件，或者使用 @event 绑定事件
- 数据双向绑定也不也不一样在 vue 中,只需要再表单元素上加上 v-model ,然后再绑定 data 中对应的一个值，当表单元素内容发生变化时， data 中对应的值也会相应改变，这是 vue 非常 nice 的一点。微信小程序必须获取到表单元素，改变的值，然后再把值赋给一个 data 中声明的变量。

6. 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；
- `App.wxss` 公共样式，引入 iconfont 等；
- `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)；
- `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)；

7. 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。

8. 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

9. 怎么封装微信小程序的数据请求的？

- 使在根目录下创建 utils 目录及 api.js 文件和 apiConfig.js 文件；
- 在 apiConfig.js 封装基础的 get, post 和 put， upload 等请求方法，设置请求体，带上 token 和异常处理等；
- 在 api 中引入 apiConfig.js 封装好的请求方法，根据页面数据请求的 urls，设置对应的方法并导出；
- 在具体的页面中导入；

10. 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

11. 请谈谈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。

12. 简述微信小程序原理?

- 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。

13. 简述路由方式?

wx.navigateTo：

- 用于保留当前页面、跳转到应用内的某个页面，使用 wx.navigateBack 可以返回到原页面。对于页面不是特别多的小程序，通常推荐使用 wx.navigateTo 进行跳转， 以便返回原页面，以提高加载速度。当页面特别多时，则不推荐使用。

- wx.redirectTo：
  - 当页面过多时，被保留页面会挤占微信分配给小程序的内存，或是达到微信所限制的 5 层页面栈。这时应该考虑选择 wx.redirectTo。wx.redirectTo()用于关闭当前页面，跳转到

应用内的某个页面。这样的跳转，可以避免跳转前页面占据运行内存，但返回时页面需要重新加载，增加了返回页面的显示时间。

- wx.reLaunch：
  - wx.reLaunch()与 wx.redirectTo()的用途基本相同， 只是 wx.reLaunch()先关闭了内存中所有保留的页面，再跳转到目标页面。
- wx.switchTab：
  - 对于跳转到 tab bar 的页面，最好选择 wx.switchTab()，它会先关闭所有非 tab bar 的页面。其次，也可以选择 wx.reLaunch()，它也能实现从非 tab bar 跳转到 tab bar，或在 tab bar 间跳转，效果等同 wx.switchTab()。使用其他跳转 API 来跳转到 tab bar，则会跳转失败。
- wx.navigateBack：
  - 用于关闭当前页面，并返回上一页面或多级页面。开发者可通过 getCurrentPages() 获取当前的页面栈，决定需要返回几层。这个 API 需要填写的参数只有 delta，表示要返回的页面数。若 delta 的取值大于现有可返回页面数时，则返回到用户进入小程序的第一个页面。当不填写 delta 的值时，就默认其为 1（注意，默认并非取 0），即返回上一页面