一、开发工具相关

1、tab键4个空格，代码结构清晰

二、目录结构相关

1、components、images、libs、pages、scripts、templates、utils等

2、components放组件，每个组件独立文件夹

3、images放图片，文件夹按tabBarItem和功能模块划分，公共image不放文件夹

4、libs放第三方library，每个library独立文件夹

5、pages放page

6、scripts放wxs

7、templates放模板，每个模板独立文件夹

8、utils存放network.js、urlManager、moduleManager及util等

三、命名相关

1、文件夹及文件使用驼峰命名法

2、image命名使用img\_module\_/icon\_mudule\_前缀，如icon\_home\_arrow.png

3、Page外全局不变变量和常量如数字或枚举类Object使用const修饰，名字大写，如const P\_MANAGER = …，其余酌情使用const、var和let

4、URL命名例URL\_MODULE\_DATA如：URL\_ORDER\_LIST，moduleManager的function命名例moduleData如orderList，复杂场景格式自行统一

四、代码格式相关

1、function前后空一行，class前后空一行，运算符前后空一格，诸如此类

2、log带前缀，如：console.log(‘orderList = ’, res.data.tableData)

五、网络相关

1、network.js，提供网络请求如post、get、upload和download方法的简易封装

2、urlManager，提供包括host、测试host以及登录接口在内的小程序使用的所有接口

3、moduleManager/appNameManager，提供针对某模块/所有模块的所有数据请求方法，传参可使用option或多参+callback形式；所有globalData中的已知数据如userid、qyid等直接写在body体内，不以方法参数的形式传递

六、app全局相关

1、app.js中除了写登录逻辑外，还可写全局的网络状况判断方法供page调用，便于在不稳定的网络环境下小程序发生异常时给出友好处理，以及其他有必要全局调用的方法

2、app.wxss写page全局统一样式，写标签常用的、通用的局部样式如flex、float、font28、theme-color、theme-red等。不是多个页面共用的极度相似的标签，不在app.wxss中写样式，须单独在page.wxss中写

七、JS相关

1、app.js全局实例，使用getApp()获取

2、尽量方法间传值代替data存储

3、data中属性写必要注释

4、尽量减少setData数据量，尽量减少setData频率，存储仅使用this.data.temp = temp

5、使用(res) => 代替function (res)，使用this代替that

6、请求返回值res判空及错误提示

7、function避免多层嵌套，做函数式封装调用

8、function写必要注释

9、工具方法不写在page.js中，util.js写工具方法，如字符串拼接、日期格式转换等

10、timer不使用时及时clear

八、JSON相关

1、写页面标题、navigationBar定制及其他独有内容

九、WXML相关

1、页面分模块，模块写注释如：<-- 轮播图 -->

2、标签尽量减少逻辑判断，复杂的逻辑判断使用wxs封装调用

3、控制属性如wx:if、wx:for等可使用<block></block>标签，避免标签过长

十、WXSS相关

1、推荐逻辑：position、float、flex、width/height/line-height/text-align、margin/padding、border、font-size、color等，由外向内，先定位后描述，上述属性仅做逻辑参考，不必需

2、同WXML，模块尽量写注释如：// 轮播图

十一、template相关

1、template的name及标签class均以temp开头，如temp-example

十二、WXS相关

1、较长及较复杂的逻辑写在wxs中，wxs可写在WXML上方或单独使用.wxs文件存放

注：以上内容不完全标准，且均为建议，不强制使用