Application cache

小程序的页面加载顺序：先加载json，配置一些全局设置，之后加载wxml和wxss，进行页面渲染，最后进行js的渲染；js的data装载之后，页面渲染完毕，会有一个onload的回调

注意：小程序在加载app.js进行全局的配置之后，会加载pages内所有页面的js，外层的会被直接执行，Page（）不会被直接调用，当该页面加载，才会调用对应js的Page（）

Page是一个对象，Page（{。。。}）是实例化一个对象，其实应该是已经在加载pages目录的js时已经进行了实例化的操作，但是因为没有出发对应的操作或者方法调用，比如page.onload（）

微信小程序分为渲染层（Webview）和逻辑层（JSCore），之间的沟通是通过微信（Native）进行沟通的，多个页面对应多个渲染层，逻辑层其实都在一个JSCore中运行

通信模型也没有简略的讲逻辑层和渲染层的通信大致过程

数据驱动也没有解释为什么不是直接用js直接生成DOM树， 可能时类比的意思，大致解释一下相关的工作 下面其实解释了，就是渲染层其实使用的方式就是wxml转为js的dom对象，然后逻辑层的数据变化时，同步过来，对应的生成的dom内容也发生变化

因为在一个JSCore中运行，要注意settimeout等的处理

Setdata的一些使用注意：设置最小改变量；不能用this.data=修改；有线程消耗，每次的数据不要超过1024kb；不要把data的任何一个值设置为undefined

页面栈最大10级

小程序的安全机制：code换取openid（微信用户唯一标识）、sessionkey（一段时间内有效）