**微信小程序之旅**

尚硅谷前端微信小程序课程

版本：V 2.0

讲师： 陈辉鸿

# 第1章 什么是小程序？

1. 2017年度百度百科十大热词之一
2. [微信](https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1/3905974" \t "https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E5%B0%8F%E7%A8%8B%E5%BA%8F/_blank)小程序（wei xin xiao cheng xu），简称[小程序](https://baike.baidu.com/item/%E5%B0%8F%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E5%B0%8F%E7%A8%8B%E5%BA%8F/_blank)，英文名Mini Program，是一种不需要下载安装即可使用的[应用](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8" \t "https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E5%B0%8F%E7%A8%8B%E5%BA%8F/_blank) ( 张小龙对其的定义是无需安装，用完即走，实际上是需要安装的，只不过小程序的体积特别小， 下载速度很快，用户感觉不到下载的过程 )
3. 小程序刚发布的时候要求压缩包的体积不能大于1M，否则无法通过，在2017年4月做了改进，由原来的1M提升到2M(该标准沿用至今)；
4. 2017年1月9日0点，万众瞩目的微信第一批小程序正式低调上线。

# 第2章 小程序可以干什么？

1. 同App进行互补，提供同app类型的功能，比app使用方便简洁
2. 通过扫一扫或者在微信搜索即可下载
3. 用户使用频率不高，但又不得不用的功能软件，目前看来小程序是首选
4. 连接线上线下
5. 开发门槛低， 成本低





# 第3章 小程序开发资料

## 3.1 相关资料

1. 官网：<https://mp.weixin.qq.com/>

开发文档地址: <https://developers.weixin.qq.com/miniprogram/dev/framework/>

1. 微信开发工具

开发工具IDE文件夹中获取

1. 下载地址

[https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html](https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html?t=2018315)

## 3.2 注册小程序账号

1. 有账号



1. 无账号



# 第4章 开发小程序储备知识

## 4.1 Flex布局简介

### 4.1.1 什么是flex布局？

1) Flex是Flexible Box的缩写，意为”弹性布局”，用来为盒状模型提供最大的灵活性。

2) 任何一个容器都可以指定为Flex布局。

3) display: ‘flex’



### 4.1.2 flex属性

1) flex-direction:

row（默认值）：主轴为水平方向，起点在左端。

row-reverse：主轴为水平方向，起点在右端。

column：主轴为垂直方向，起点在上沿。

column-reverse：主轴为垂直方向，起点在下沿。

### 4.1.3 学习地址：

<http://www.runoob.com/w3cnote/flex-grammar.html>

Flex小青蛙

## 4.2 移动端相关知识

### 4.2.1 物理像素(专门用于描述屏幕大小)

1) 屏幕的分辨率

2) 设备能控制显示的最小单元，可以把物理像素看成是对应的像素点

### 4.2.2 设备独立像素(用于连接物理像素和CSS像素的专用单位) & css像素(专门用于描述浏览器中内容大小)

设备独立像素(也叫密度无关像素)，可以认为是计算机坐标系统中的一个点，这个点代表一个可以由程序使用并控制的虚拟像素(比如：CSS 像素,只是在android机中CSS 像素就不叫”CSS 像素”了而是叫”设备独立像素”)，然后由相关系统转换为物理像素。

### 4.2.3 dpr比 & DPI & PPI

1) dpr: 设备像素比，物理像素宽度/设备独立像素宽度 = dpr， 一般以Iphone6的dpr为准 dpr = 2

2) PPI: 一英寸显示屏上的像素点个数

3) DPI：最早指的是打印机在单位面积上打印的墨点数，墨点越多越清晰







## 4.3 移动端适配方案

### 4.3.1 viewport适配

1. 为什么做viewport适配
   1. 手机厂商在生产手机的时候大部分手机默认页面宽度为980px
   2. 手机实际视口宽度都要小于980px，如: iphone6为375px
   3. 开发需求： 需要将980的页面完全显示在手机屏幕上且没有滚动条
2. 实现：

<meta name="viewport" content="width=device-width,initial-scale=1.0">

### 4.3.2 rem适配

1. 为什么做rem适配
   1. 机型太多，不同的机型屏幕大小不一样
   2. 需求： 一套设计稿的内容在不同的机型上呈现的效果一致，根据屏幕大小不同的变化，页面中的内容也相应变化
2. 实现：

|  |
| --- |
| ***function*** *remRefresh****()******{***  ***let*** *clientWidth* ***=*** *document.documentElement.clientWidth****;***  **// 将屏幕等分10份**  ***let*** *rem* ***=*** *clientWidth* ***/*** *10****;***  *document.documentElement.style.fontSize* ***=*** *rem* ***+*** *'px'****;***  *document.body.style.fontSize* ***=*** *'12px'****;***  ***}***  *window.addEventListener****(****'pageshow'****,******()******=>******{***  *remRefresh****()***  ***})***  *// 函数防抖*  ***let*** *timeoutId****;***  *window.addEventListener****(****'resize'****,******()******=>******{***  *timeoutId* ***&&*** *clearTimeout****(****timeoutId****);***  *timeoutId* ***=*** *setTimeout****(()******=>{***  *remRefresh****()***  ***},*** *300****)***  ***})*** |

3. 第三方库实现

lib-flexible + px2rem-loader

## 4.4 扩展内容

视网膜屏幕是分辨率超过人眼识别极限的高分辨率屏幕，由[苹果公司](https://baike.baidu.com/item/%E8%8B%B9%E6%9E%9C%E5%85%AC%E5%8F%B8" \t "https://baike.baidu.com/item/%E8%A7%86%E7%BD%91%E8%86%9C%E5%B1%8F%E5%B9%95/_blank)在2010年在[iPhone 4](https://baike.baidu.com/item/iPhone%204" \t "https://baike.baidu.com/item/%E8%A7%86%E7%BD%91%E8%86%9C%E5%B1%8F%E5%B9%95/_blank)发布会上首次推出营销术语。

Iphone4的dpr = 2; 人类肉眼分辨的极限

问题： Iphone6的dpr为多少？Iphone6Plus比Iphone显示图像清晰吗？

# 第5章 小程序特点

## 5.1 小程序特点概述

1. 没有DOM(小程序并没有遵守W3C的语法规范)
2. 组件化开发
3. 体积小，单个压缩包体积不能大于2M，否则无法上线
4. 小程序的四个重要的文件
   1. \*.js
   2. \*.wxml ---> view结构 ----> html
   3. \*.wxss ---> view样式 -----> css
   4. \*. json ----> view 配置 -----> json文件
5. 小程序适配方案: rpx (responsive pixel响应式像素单位)
   1. 小程序适配单位： rpx
   2. 规定屏幕宽度为750rpx(屏幕750等分的rem)
   3. Iphone6下： 1rpx = 1物理像素 = 0.5px



## 5.2 小程序配置

### 5.2.1 全局配置： app.json

1. 作用： 用于为整个应用进行选项设置
2. 链接:
   1. <https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html>
3. 配图:



### 5.2.2 页面配置: 页面名称.json

1. 作用: 用于为指定的页面进行配置
2. 链接:
   1. <https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html>
3. 注意事项: 页面配置的优先级高于全局配置
4. 配图:



### 5.2.3 sitemap配置: sitemap.json

1. 作用：用于被微信搜索爬取页面
2. 链接:
   1. <https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html>
3. 配图:



## 5.3 小程序框架接口

### 5.3.1 App

1. 全局app.js中执行App()
2. 生成当前应用的实例对象
3. getApp()获取全局应用实例

### 5.3.2 Page

1. 页面.js中执行Page()
2. 生成当前页面的实例
3. 通过getCurrentPages获取页面实例

# 第6章 wxml语法

## 6.1 数据绑定

### 6.1.1 初始化数据

1. 页面.js的data选项中



### 6.1.2 使用数据

1. 模板结构中使用双大括号 {{message}}
2. 注意事项: 小程序中为单项数据流 model ---> view



### 6.1.3 修改数据

1. this.setData({message: ‘修改之后的数据’}, callback)
2. 特点：
   1. 同步修改： this.data值被同步修改
   2. 异步更新： 异步将setData 函数用于将数据从逻辑层发送到视图层（异步）



## 6.2 事件绑定

### 6.2.1 事件分类

1) 冒泡事件

a) 定义：冒泡事件：当一个组件上的事件被触发后，该事件会向父节点传递。

b) 冒泡事件列表：

<https://mp.weixin.qq.com/debug/wxadoc/dev/framework/view/wxml/event.html>

1. 非冒泡事件

a) 定义：当一个组件上的事件被触发后，该事件不会向父节点传递。

b) 非冒泡事件：表单事件和自定义事件通常是非冒泡事件

<https://mp.weixin.qq.com/debug/wxadoc/dev/framework/view/wxml/event.html>

### 6.2.2 绑定事件

1. **bind**绑定：事件绑定不会阻止冒泡事件向上冒泡

|  |
| --- |
| <**view bindtap="handleTap" class='start\_container'**>  <**text class='start'**>开启小程序之旅</**text**> </**view**> |

1. **catch** 绑定: 事件绑定可以阻止冒泡事件向上冒泡

|  |
| --- |
| <**view catchtap="handleTap" class='start\_container'**>  <**text class='start'**>开启小程序之旅</**text**> </**view**> |

### 6.2.3 向事件对象传参

1. 语法： data-key=value



1. 获取: event.target.dataset.key || event.currentTarget.dataset.key



1. Event.target 和 event.currentTarget的区别
   1. Event.target是触发事件的对象，但不一样是绑定事件的对象，如： 事件委托，冒泡
   2. currentTarget触发时间的对象一定是绑定事件的对象， 没有事件委托

## 6.3 列表渲染

### 6.3.1 语法说明

1. wx:for=’{{arr}}’
2. wx:key=’{{唯一值}}’

### 6.3.2 注意事项

1. 默认的个体: item
2. 默认的下标: index
3. 自定义个体变量名称: wx:for-item=’myItem’
4. 自定义下标变量名称: wx:for-index=’myIndex’

## 6.4 条件渲染

### 6.4.1 语法说明

1. wx:if=’条件’
2. wx:elif=’条件’
3. wx:else



### 6.4.2 wx:if VS hidden

1. hidden用法: <view hidden=’{{true}}’ ></view>
2. wx:if 等同于 v-if, 条件为false的时候不加载，条件切换的时候决定元素销毁或者重新加载渲染
3. hidden 等同于 v-show, 始终加载元素, 条件切换的时候决定元素的显示和隐藏

## 6.5 模板使用

### 6.5.1 定义模板



### 6.5.2 引入模板

1. 引入模板结构: <import src=’模板结构相对路径’ />
2. 引入模板样式: @Import ‘模板样式路径’

### 6.5.3 使用模板



### 6.5.4 向模板导入数据并使用数据



## 6.6 生命周期

### 6.6.1 对应阶段说明

1. onLoad(Object query)
   1. 页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数。
   2. **参数：**

| 名称 | 类型 | 说明 |
| --- | --- | --- |
| query | Object | 打开当前页面路径中的参数 |

1. onShow()
   1. 页面显示/切入前台时触发
   2. 会执行多次
2. onReady()
   1. 页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。
3. onHide()
   1. 页面隐藏/切入后台时触发。 如 [wx.navigateTo](https://developers.weixin.qq.com/miniprogram/dev/api/route/wx.navigateTo.html) 或底部 tab 切换到其他页面，小程序切入后台等。
4. onUnload()
   1. 页面卸载时触发。如[wx.redirectTo](https://developers.weixin.qq.com/miniprogram/dev/api/route/wx.redirectTo.html)或[wx.navigateBack](https://developers.weixin.qq.com/miniprogram/dev/api/route/wx.navigateBack.html)到其他页面时。

### 6.6.2 官网图示说明



### 6.6.3 官网对应地址

<https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/page-life-cycle.html>

# 第7章 小程序API

## 7.1 API使用说明

1. 小程序提供了很多实用的方法供开发者使用
2. 小程序全局对象是: wx
3. 所有的API都保存在wx对象中



## 7.2 常用API

1. 界面交互
   1. 显示消息提示框: wx.showToast()
   2. 显示消息加载框: wx.showLoading()
   3. 关闭消息提示框: wx.hideToast()
   4. 关闭消息加载框: wx.hideLoading()
2. 路由跳转
   1. wx.navigateTo()
   2. wx.redirectTo()
   3. wx.switchTab()
3. 网络请求
   1. wx.request()
4. 本地存储
   1. wx.setStorage()
   2. wx.setStorageSync()
   3. wx.getStorage()
   4. wx.getStorageSync()
5. 媒体
   1. wx.getBackgroundAudioManager()
   2. wx.playVoice()

## 7.3 快速查找技巧

1. 小程序的初学者可能对于小程序的官网的众多内容一时毫无头绪，无从下手不知道从哪来找想要的内容
2. 当在小程序中想要实现某一种布局，查看：组件
3. 当在小程序中想要实现某一个功能，查看: API
4. 当在小程序中想要进行某一个配置或者某一种页面语法，查看: 框架 + 指南
5. 查看小程序官网的时候要细心，最好是将要使用的API的相关内容看完整，因为API的配置及限制较多

# 第8章 小程序重点知识汇总

## 8.1 小程序本地存储

### 8.1.1 语法说明

1. 存入数据
   1. wx.setStorage() 异步
   2. wx.setStorageSync() 同步



1. 读取数据
   1. wx.getStorage()异步
   2. wx.getStorageSync() 同步



1. 删除数据
   1. wx.removeStorage() 异步
   2. wx.removeStroageSync() 同步



1. 清空数据
   1. wx.clearStorage() 异步
   2. wx.clearStorageSync() 同步



### 8.1.2 注意事项

1. 除非用户主动删除或因存储空间原因被系统清理，否则数据都一直可用
2. 单个 key 允许存储的最大数据长度为 1MB
3. 所有数据存储上限为 10MB

### 8.1.3 官网对应地址

<https://developers.weixin.qq.com/miniprogram/dev/api/storage/wx.setStorage.html>



## 8.2 小程序前后端交互

### 8.2.1 语法说明

1. wx.request()



### 8.2.2 相关配置

1. 每个微信小程序需要事先设置通讯域名，小程序**只可以跟指定的域名进行网络通信**
2. 服务器域名请在 「小程序后台-开发-开发设置-服务器域名」 中进行配置
3. 默认超时时间和最大超时时间都是 **60s**
4. 超时时间可以在 app.json 中通过 [networktimeout](https://developers.weixin.qq.com/miniprogram/dev/framework/config.html) 配置

### 8.2.3 注意事项

1. 小程序为了安全起见只支持Https请求
2. wx.request最大并发限制10个

### 8.2.4 官网对应地址

<https://developers.weixin.qq.com/miniprogram/dev/api/storage/wx.setStorage.html>



## 8.3 小程序页面通信

### 8.3.1 路由传参

1. 传参方式
   1. 路由地址中 + query传参数
   2. 示例： url?a=123
2. 获取参数
   1. 跳转目标页面的onLoad函数中的options实参中获取



### 8.3.2 消息订阅发布

1. 使用第三方库: pubsub-js
2. 安装: npm install pubsub-js
3. 使用：
   1. Import PubSub from ‘pubsub-js’
   2. 订阅消息: PubSub.subscribe(‘eventName’, callback)
   3. 发布消息: PubSub.publish(‘eventName’, data)
   4. 取消订阅: PubSub.unsubscribe(‘eventName’)

### 8.3.3 eventChannel 事件通道

1. 订阅事件
   1. wx.navigateTo()跳转的时候在events选项中定义事件名及事件对应的回调



1. 获取事件总线对象
   1. 目标页面中通过: 实例.getOpenerEventChannel()
   2. 示例: const eventChannel = this.getOpenerEventChannel()
2. 触发事件
   1. eventChannel.emit(‘事件名’, data)



## 8.4 小程序自定义组件

### 8.4.1 创建组件

1. 开发工具中右键新建组件
2. 组件对应的json文件中设置: component: true



### 8.4.2 使用组件

1. 使用组件的页面的json文件中注册使用组件



## 8.5 小程序使用npm包

### 8.5.1 初始化package.json

npm init -y

### 8.5.2 勾选允许使用npm



### 8.5.3 下载npm包

npm install packageName

### 8.5.4 构建npm

1. 开发工具 ---> 工具 ---> 构建npm
2. 会将node\_modules中的包打包到miniprogram\_npm中

### 8.5.5 流程执行不完整带来的错误



## 8.6 小程序获取用户基本信息

### 8.6.1 首次登陆获取

1. Button组件设置open-type属性为getUserInfo
2. <button open-type=’getUserInfo’></button>
3. 设置后首次登陆点击button可以弹出授权窗口
4. 注意： 授权的动作只发生一次，除非清除缓存，点击butotn授权一次之后再点击失效，不会弹出授权窗口
5. 官网对应地址

<https://developers.weixin.qq.com/miniprogram/dev/component/button.html>



### 8.6.2 授权之后获取

1. wx.getUserInfo()
2. 官网对应地址:

<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/user-info/wx.getUserInfo.html>



## 8.7 小程序获取用户唯一标识（openId）

### 8.7.1 官网图解



### 8.7.2 获取流程

1. wx.login()



1. 发送code给服务器端
2. 服务器端发送请求携带参数(code, appSecret, appId)给微信服务器获取openId
   1. 接口地址:
3. appSecret，appId在小程序首页获取



1. 服务器获取openId后进行加密返回给前端

### 8.7.3 获取接口地址

请求方法： GET https://api.weixin.qq.com/sns/jscode2session?appid=APPID&secret=SECRET&js\_code=JSCODE&grant\_type=authorization\_code

## 8.8 小程序分包流程

### 8.8.1 什么要分包

1. 小程序要求压缩包体积不能大于2M，否则无法发布
2. 实际开发中小程序体积如果大于2M就需要使用分包机制进行发布上传
3. 分包后可解决2M限制，并且能分包加载内容，提高性能
4. 分包后单个包的体积不能大于2M
5. 分包后所有包的体积不能大于12M

### 8.8.2 分包形式

1. 常规分包
2. 独立分包
3. 分包预下载

### 8.8.3 常规分包

1. 开发者通过在 app.json subpackages 字段声明项目分包结构
2. 特点：
   1. 加载小程序的时候先加载主包，当需要访问分包的页面时候才加载分包内容
   2. 分包的页面可以访问主包的文件，数据，图片等资源
   3. 主包：
      1. 主包来源： 除了分包以外的内容都会被打包到主包中
      2. 通常放置启动页/tabBar页面





### 8.8.4 独立分包

1. 设置independent为true
2. 特点:
   1. 独立分包可单独访问分包的内容，不需要下载主包
   2. 独立分包不能依赖主包或者其他包的内容
3. 使用场景
   1. 通常某些页面和当前小程序的其他页面关联不大的时候可进行独立分包
   2. 如：临时加的广告页 || 活动页



### 8.8.5 分包预下载

1. 配置
   1. app.json中设置preloadRule选项
   2. key(页面路径): {packages: [预下载的包名 || 预下载的包的根路径])}





1. 特点:
   1. 在加载当前包的时候可以设置预下载其他的包
   2. 缩短用户等待时间，提高用户体验

### 8.8.6 分包效果演示



### 8.8.7 官网对应地址

<https://developers.weixin.qq.com/miniprogram/dev/framework/subpackages.html>

## 8.9 小程序转发分享

### 8.9.1 分享实现

1. Button组件设置open-type为share
2. <button open-type=’share’ ></button>

### 8.9.2 自定义分享内容

1. 生命周期回调中onShareAppMessage回调中return 对象设置自定义内容



### 8.9.3 设置体验权限

1. 开发阶段分享给微信好友，默认没有体验权限，无法打开分享小程序，需要在开发页面设置
2. 最多添加15个微信好友



## 8.10 小程序支付流程

### 8.10.1 支付流程官网图解



### 8.10.2 支付流程详细说明

1. 用户在小程序客服端下单(包含用户及商品信息)
2. 小程序客户端发送下单支付请求给商家服务器
3. 商家服务器同微信服务器对接获取唯一标识openID
4. 商家服务器根据openId生成商户订单(包含商户信息)
5. 商家服务器发送请求调用统一下单API获取预支付订单信息
   1. 接口地址: <https://api.mch.weixin.qq.com/pay/unifiedorder>
6. 商家对预支付信息签名加密后返回给小程序客户端
   1. 签名方式： MD5
   2. 签名字段：小程序ID, 时间戳， 随机串，数据包，签名方式
   3. 参考地址: <https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=7_7&index=3>
7. 用户确认支付（鉴权调起支付）
   1. API: wx.requestPayment()
8. 微信服务器返回支付结果给小程序客户端
9. 微信服务器推送支付结果给商家服务器端

### 8.10.3 官网对应地址

<https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=7_3&index=1>