**微信小程序之旅**

尚硅谷前端微信小程序课程

版本：V 1.0

# 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；
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类似的功能，比app操作更加简洁的轻应用
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/>
2. 微信开发工具



1. 下载地址

<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



## **3.3 效果演示**

# 第4章 开发小程序项目

## **4.1 前言叙述**

1. 小程序没有DOM对象，一切基于组件化
2. 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 数据 -----> json文件
3. 储备知识
   1. 理解事件机制
   2. 理解组件化
   3. 理解数据绑定
   4. Flex布局
   5. 移动端适配方案
4. 贴心小建议

学习vue后开发小程序的感觉会爽到飞起

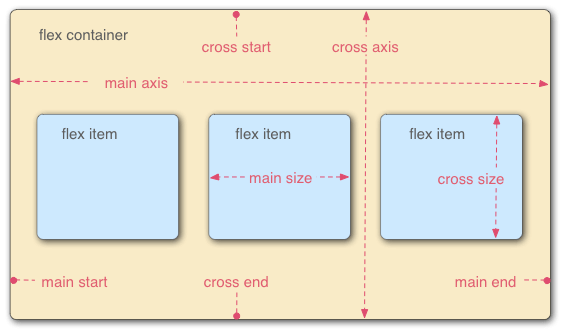
## **4.2 Flex布局简介**

### 4.2.1 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’



### 4.2.2 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。

### 4.2.3 学习地址：

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

## **4.3 移动端适配**

### 4.3.1 物理像素

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

### 4.3.2 设备独立像素 & css像素

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。

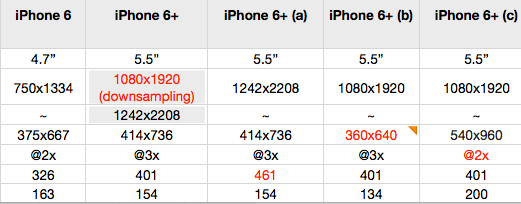
### 4.3.3 dpr比 & DPI & PPI

1) dpr: 设备像素比，物理像素/设备独立像素 = dpr， 一般以Iphon6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







## **4.4 小程序适配方案**

Iphon6： 1rpx = 1物理像素 = 0.5px

微信官方提供的换算方式：

1. 以iPhone6的物理像素个数为标准: 750;
2. 1rpx = 目标设备宽度 / 750 \* px;
3. 注意此时底层已经做了viewport适配的处理，即实现了理想视 口

## **4.5 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 4" \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。

Iphone的dpr = 2; 人类肉眼分辨的极限

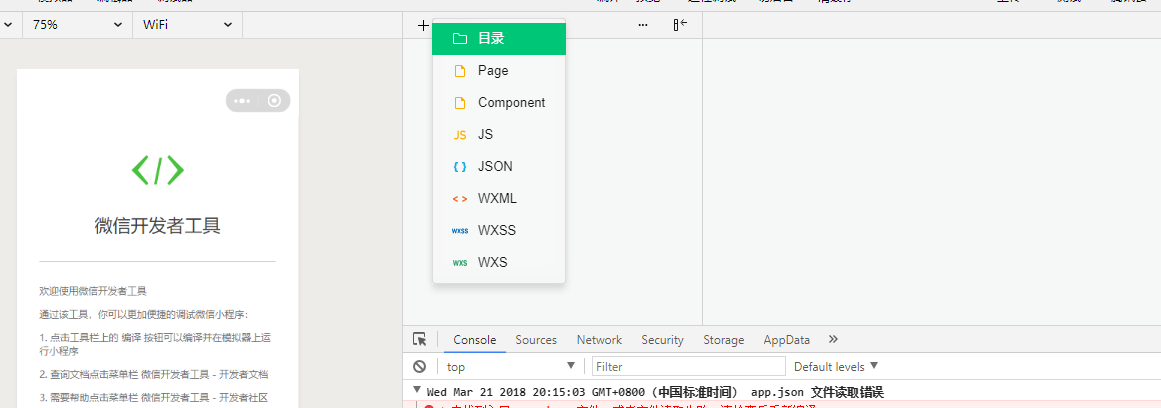
问题： Iphone6的dpr为多少？Iphone6Plus比Iphone显示图像清晰吗？

## **4.6登录页小试牛刀**

### 4.6.1 效果演示

****

### 4.6.2 注册程序，创建文件



### 4.6.3 应用主文件(入口文件)

A. App.wxml

B. App.wxss

C. App.js

D. App.json

### 4.6.4 主页结构： index.wxml

|  |
| --- |
| <**view class='container'**>  <**image class='user\_avatar' src='{{userInfo.avatarUrl}}'**></**image**>  <**text class='user\_name'**><**text style='color**:**red'**>hello</**text**> {{userInfo.nickName}}</**text**>  <**view class='start\_container' bindtap="handleTap"**>  <**text class='start'**>开启小程序之旅</**text**>  </**view**> </**view**> |

### 4.6.5 主页样式： index.wxss

|  |
| --- |
| page {    width: 100%;    height: 100%;    background: #C5EFFF;  }  .container {    display: flex;    flex-direction: column;    align-items: center;  }  .container image {    width: 200rpx;    height: 200rpx;    border-radius: 100rpx;    margin-top: 160rpx;  }  .username {    font-size: 32rpx;    font-weight: bold;    margin-top: 100rpx;  }  .start\_container {    width: 200rpx;    height: 80rpx;    margin-top: 200rpx;    border: 1px solid #405f80;    text-align: center;    border-radius: 10rpx;  }  .start\_container text {    font-size: 22rpx;    font-weight: bold;    line-height: 80rpx;    color: #405f80;  } |

### 4.6.6 入口文件： app.json配置

|  |
| --- |
| {  "pages": [  "pages/index/index"  ],    "window": {      "navigationBarBackgroundColor": "#C5EFFF"    }  } |

## **4.7 数据绑定 & 事件**

### 4.7.1 数据绑定

1) 思想： 同Vue一样数据可以初始化在当前页面的管理内存的data中，页面中使用数据会自动去data里找

2) 初始化数据： 当前页面的js文件

|  |
| --- |
| Page({    /\*\*     \* 页面的初始数据     \*/    data: {      msg: '开启小程序之旅'    },  }) |

1. 修改数据

|  |
| --- |
| this.setData({        msg: '我是修改之后的数据'      }) |

### 4.7.2 声明周期钩子函数

|  |
| --- |
| Page({    /\*\*     \* 页面的初始数据     \*/    data: {      msg: '开启小程序之旅'    },    /\*\*     \* 生命周期函数--监听页面加载     \*/    onLoad: function (options) {      },    /\*\*     \* 生命周期函数--监听页面初次渲染完成     \*/    onReady: function () {      },    /\*\*     \* 生命周期函数--监听页面显示     \*/    onShow: function () {      },    /\*\*     \* 生命周期函数--监听页面隐藏     \*/    onHide: function () {      },    /\*\*     \* 生命周期函数--监听页面卸载     \*/    onUnload: function () {      },    /\*\*     \* 页面相关事件处理函数--监听用户下拉动作     \*/    onPullDownRefresh: function () {      },    /\*\*     \* 页面上拉触底事件的处理函数     \*/    onReachBottom: function () {      },    /\*\*     \* 用户点击右上角分享     \*/    onShareAppMessage: function () {      }  }) |

### 4.7.3 获取登录用户的数据

1) 文档查看位置: API---> 开放接口

2) 代码示例：

|  |
| --- |
| // 获取登录用户的数据      wx.getUserInfo({        //withCredentials: true,        success: (res) => {          console.log(res);          let user = res.userInfo;          this.setData({            user          })        }      }) |

### 4.7.4 自定义事件

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>

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

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

### 4.7.6 跳转页面

1. wx.navigateTo(OBJECT)：跳转有记录，可通过回退键进行回退9

|  |
| --- |
| // 跳转到list页面      wx.navigateTo({        url: '/pages/list/list',        success(){          console.log('跳转成功');        }      }) |

1. wx.redirectTo(OBJECT)： 没有记录，不能进行回退

|  |
| --- |
| // 跳转到list页面      wx.redirectTo({        url: '/pages/list/list',        success(){          console.log('跳转成功');        }      }) |

## **4.8列表页：list**

### 4.8.1 效果演示





### 4.8.2 注册页面

1) list.wxml

2) list.wxss

3) list.js

4) list.json

## **4.9组件 & 模板使用**

### 4.9.1 组件

|  |
| --- |
| <view>    <swiper indicator-dots='true' indicator-color='deepPink'>      <swiper-item>        <image src='/images/detail/carousel/iqiyi.png'></image>      </swiper-item>      <swiper-item>        <image src='/images/detail/carousel/vr.png'></image>      </swiper-item >      <swiper-item>        <image src='/images/detail/carousel/wx.png'></image>      </swiper-item >    </swiper>  </view> |

### 4.9.2 模板

1) 定义模板： name=‘模板名字’

|  |
| --- |
| <template name='listTemplate'>    <view>      <text>tempalte content</text>    </view>  </template> |

2) 使用模板

|  |
| --- |
| <**import src="./detail\_template/detail\_item\_template.wxml"** /> |

1. 使用模板样式

|  |
| --- |
| @import '../templates/list-template.wxss'; |

## 4.10 数据存储

1) 存数据：setStorage， 同步： setStorageSync

|  |
| --- |
| // 数据存储      wx.setStorage({        key: 'isCollected',        data: oldCollectFlag      }) |

1. 获取数据：getStorage 同步: getStorageSync

|  |
| --- |
| let oldCollectFlag = wx.getStorageSync('isCollected'); |

## **4.10 音乐播放控制**

### 4.10.1 音乐播放

1) API: wx.playBackgroundAudio

2) 示例：

|  |
| --- |
| wx.playBackgroundAudio({          dataUrl: detailObj.music.dataUrl,          title: detailObj.music.title,          success() {            console.log('音乐播放成功');          }        }); |

### 4.10.2 音乐暂停

1) API: wx.pauseBackgroundAudio

2) 示例：

|  |
| --- |
| wx.pauseBackgroundAudio |

### 4.10.3 监听音乐播放暂停

1) 监听播放API: wx.onBackgroundAudioPlay

2) 监听暂停API: wx.onBackgroundAudioPause

2) 示例：

|  |
| --- |
| wx.onBackgroundAudioPlay(() => {        console.log('音乐播放'); // 注意真机上音乐播放监听只能执行一次。        this.setData({          isMusicPlay: true        })        // 修改app数据记录是否播放和播放页面的下标        appDatas.appData.isPlay = true;        appDatas.appData.playPageIndex = this.data.index;  })   // 监听音乐是否暂停      wx.onBackgroundAudioPause(() => {        console.log('音乐暂停');        this.setData({          isMusicPlay: false        })        appDatas.appData.isPlay = false;        appDatas.appData.playPageIndex = this.data.index;      }) |

### 4.10.4 app应用存储公共数据

1) 存取数据

|  |
| --- |
| App({  **data**: {  **isPlay**: **false**,  **playPageIndex**: **null**,  **movies**: {}  } }) |

2) 读取数据

|  |
| --- |
| **let *app*** = getApp();  app.data.isPlay = false; |

备注：app方法配置对象中的属性和方法均为app应用实例的属性和方法

## **4.11 事件委托 & 事件数据传递**

### 4.11.1 轮播图片点击跳转

1) 事件委托给父元素：swiper

|  |
| --- |
| <swiper catchtap='imgToDetail' indicator-dots='true' indicator-color='deepPink'>      <swiper-item>        <image data-detailId='{{4}}' src='/images/detail/carousel/01.jpg'></image>      </swiper-item>      <swiper-item>        <image data-detailId='{{5}}' src='/images/detail/carousel/02.jpg'></image>      </swiper-item >      <swiper-item>        <image data-detailId='{{6}}' src='/images/detail/carousel/03.jpg'></image>      </swiper-item >    </swiper> |

### 4.11.2 传递事件数据

1) data-’传递数据key’ = value

### 4.11.3 currentTarget 和target的区别

1) target指向的是触发事件的元素

2) currentTarget指向的是捕获事件的元素

## 4.12 数据交互

### 4.12.1 发送请求

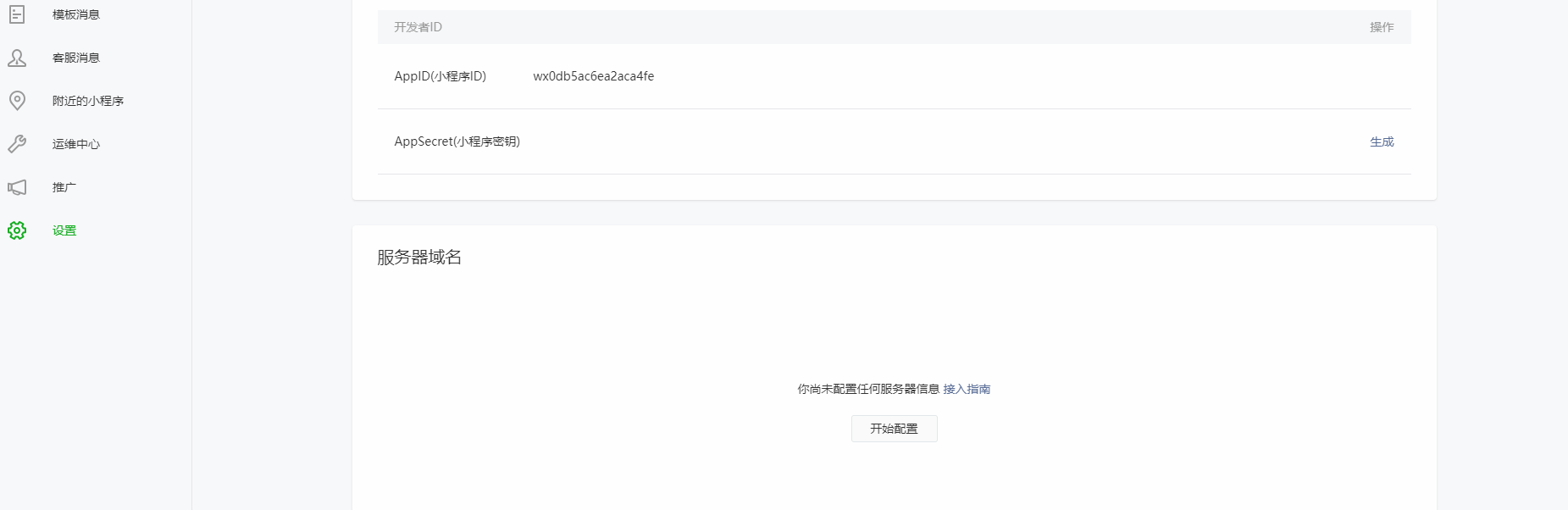
|  |
| --- |
| wx.request({ // 发送请求  **url**: ***API\_URL***, // 请求的url  **header**: { // 设置请求头  **'Content-type'**: **'json'**,  },  success: (res) => { // 请求成功的回调函数  *// 隐藏提示加载信息* wx.hideToast();  **console**.log(res); // 请求的成功的数据对象，注意是封装后的对象  **this**.setData({**movies**: res.**data**.subjects})  } }) |

### 4.12.2 小程序设置

备注：

1. 小程序出于安全考虑所有的协议都是https协议，且如果没有在开发设置中配置请求的连接是无法访问指定的链接的。
2. 一个微信小程序的并发网络请求数量被限制在最多5个

、



### 4.12.3 不同页面之间通信

* 1. a页面

|  |
| --- |
| **let *appData*** = getApp(); // 获取app中的数据对象  onLoad (options) {  *// 加载到数据之前给用户显示‘正在加载的提示信息’* wx.showToast({  **title**: **"加载中..."**,  **icon**: **"loading"**,  **duration**: 3000  });  wx.request({  **url**: ***API\_URL***,  **header**: {  **'Content-type'**: **'json'**,  },  success: (res) => {  *// 隐藏提示加载信息* wx.hideToast();  **console**.log(res);  **this**.setData({**movies**: res.**data**.subjects})  ***appData***.**appData**.**movies** = res.**data**.subjects; // 将数据更新至公共的app数据对象中  }  }) }, |

* 2. b页面

|  |
| --- |
| **let *appData*** = getApp();  onLoad: **function** (options) {  **console**.log(options.**id**);  **console**.log(***appData***.**appData**.**movies**);  **this**.setData({**movie**: ***appData***.**appData**.**movies**[options.**id**]}) }, |