爱创课堂前端培训

# 微信小程序

第1天课堂笔记（本课程共4天）

班级：北京前端训练营19期

讲师：李兰波

日期：2019年2月21日

爱创课堂官网 ：[www.icketang.com](http://www.icketang.com)

# 注册微信小程序开发者

下载并安装好微信开发工具之后，下一步就是登录开发工具并进行项目的开发。

首先，必须要有微信账号，其次，将该微信账号注册为开发者。

第一步：登录微信公众平台

网址：<https://mp.weixin.qq.com/>

立即注册=>点击小程序=>填写邮箱、密码、验证码=>去邮箱中激活邮件=>得到APPID

注册完毕之后，可以登录微信公众平台。查看自己的APPID：

|  |
| --- |
|  |

得到APPID之后，就可以登录微信开发工具，进行项目的开发了。

第二步：登录微信开发工具

打开程序

|  |
| --- |
|  |

扫码登录：

|  |
| --- |
|  |

选择小程序开发：

|  |
| --- |
|  |

填写相关信息：

|  |
| --- |
|  |

项目目录：开发微信小程序的目录 必须是空的

AppID：开发者的身份，可以点击测试号中的小程序使用测试号开发

项目名称：唯一的项目的名称

当全部填写完毕时，会出现一个勾选框：

|  |
| --- |
|  |

有建立普通快速启动模板、建立云快速启动模板等一系列选项。

我们选择普通快速启动模板即可。

点击确定，开发者工具进入开发模式。

|  |
| --- |
|  |

# 微信小程序的特点

## 2.1 渲染模式

微信小程序是两个线程。

一个线程用于渲染视图。使用的是web-view。

一个线程用于执行脚本。

## 2.2 数据驱动

微信小程序通过监听数据，驱动程序。

## 2.3 模块化

微信小程序也是模块化的，支持的规范是CommonJS。

# 目录结构

通过观察自动创建的目录结构，我们发现，两个文件夹：pages、utils。四个文件：app.js、app.json、app.wxss、project.config.js

## 3.1 pages

该目录是用于存放所有页面的。它里面的每一个成员都是一个文件夹，每一个文件夹代表一个页面。

每一个文件夹下都有四个文件：文件夹名称.js、文件夹名称.json、文件夹名称.wxss、文件夹名称.wxml

这四个文件共同组成一个页面。

注：该pages文件夹的名称是可以自定义的。

## 3.2 utils

该目录是用于存放所有的工具模块的。

## 3.3 app.js

这是一个全局的小程序的脚本文件。里面可以定义小程序的各个生命周期执行函数。以及全局使用的数据等。

查看内容：

|  |
| --- |
| 1. App({ 2. // 配置各个生命周期函数 3. onLaunch: function() {}, 4. onShow: function() {}, 5. onHide: function() {}, 6. // 配置全局对象 7. aaa: {} 8. }) |

App是一个构造函数，负责构造应用程序。所以也叫做“应用程序构造器”。它的作用是注册当前应用程序。

## 3.4 app.json

这个文件是用于配置小程序的全局信息的。

比如，当前小程序中一共有多少个页面、小程序的布局、小程序的底部按钮、小程序中各类请求的超时时间等。

官方地址：

<https://developers.weixin.qq.com/miniprogram/dev/framework/config.html#%E5%85%A8%E5%B1%80%E9%85%8D%E7%BD%AE>

app.json:

|  |
| --- |
| 1. { 2. "pages":[ 3. "pages/index/index", 4. "pages/logs/logs" 5. ], 6. "window":{ 7. "backgroundTextStyle":"light", // 背景的文字颜色 8. "navigationBarBackgroundColor": "#fff", // 导航条的背景颜色 9. "navigationBarTitleText": "ICKT", // 导航条的文本 10. "navigationBarTextStyle":"black" // 导航条的文字颜色 11. } 12. "tabBar": { 13. "list": [ 14. { 15. "pagePath": "pages/index/index", // 当点击当前按钮的时候显示的页面 16. "text": "微信", // 文本 17. "iconPath": "/icons/message\_normal.png", // 当按钮处于不选中状态时的图片 18. "selectedIconPath": "/icons/message\_active.png" // 当按钮处于选中状态时的图片 19. }, 20. { 21. "pagePath": "pages/logs/logs", 22. "text": "联系人", 23. "iconPath": "/icons/friends\_normal.png", 24. "selectedIconPath": "/icons/friends\_active.png" 25. } 26. ] 27. }, 28. "networkTimeout": { 29. "request": 20000, 30. "connectSocket": 20000, 31. "uploadFile": 20000, 32. "downloadFile": 20000 33. } 34. } |

pages: 是一个数组，用于规定当前小程序中一共包含多少个页面。

window: 是一个对象，用于配置当前的视口内容。

tabBar: 是一个对象，对象中有一个属性list 用于定义底部的按钮。

networkTimeout: 是一个对象，用于定义各种请求的超时时间。

## 3.5 app.wxss

wxss: 微信样式 其实就是CSS，只是微信将它封装，在这里面，可以使用任何特性。

全局样式配置，等价于common.css

## 3.6 project.config.json

项目配置

# 页面结构

每一个页面由四个文件组成：.wxml、.wxss、.js、 .json

wxml: 负责结构 等价于HTML代码

wxss: 负责样式 等价于CSS代码

js: 负责交互 等价于JS代码

json: 负责页面的配置

## 4.1 wxml

负责结构

|  |
| --- |
| 1. <!--logs.wxml--> 2. <view class="container log-list"> 3. <block wx:for="{{logs}}" wx:for-item="log"> 4. <text class="log-item">{{index + 1}}. {{log}}</text> 5. </block> 6. </view> |

## 4.2 wxss

负责页面样式 与全局的app.wxss不同在于app.wxss负责全部页面的样式 而页面中的wxss只对该页面负责

|  |
| --- |
| 1. /\*\*index.wxss\*\*/ 2. .userinfo { 3. display: flex; 4. flex-direction: column; 5. align-items: center; 6. } 7. .userinfo-avatar { 8. width: 128rpx; 9. height: 128rpx; 10. margin: 20rpx; 11. border-radius: 50%; 12. } 13. .userinfo-nickname { 14. color: #aaa; 15. } 16. .usermotto { 17. margin-top: 200px; 18. } |

## 4.3 js

当前页面的脚本都存放在页面中的js文件中。

当前页面的生命周期函数也可以在js中配置。

|  |
| --- |
| 1. //index.js 2. //获取应用实例 3. const app = getApp() 4. Page({ 5. data: { 6. motto: 'Hello World', 7. userInfo: {}, 8. hasUserInfo: false, 9. canIUse: wx.canIUse('button.open-type.getUserInfo') 10. }, 11. //事件处理函数 12. bindViewTap: function() { 13. wx.navigateTo({ 14. url: '../logs/logs' 15. }) 16. }, 17. onLoad: function () { 18. if (app.globalData.userInfo) { 19. this.setData({ 20. userInfo: app.globalData.userInfo, 21. hasUserInfo: true 22. }) 23. } else if (this.data.canIUse){ 24. // 由于 getUserInfo 是网络请求，可能会在 Page.onLoad 之后才返回 25. // 所以此处加入 callback 以防止这种情况 26. app.userInfoReadyCallback = res => { 27. this.setData({ 28. userInfo: res.userInfo, 29. hasUserInfo: true 30. }) 31. } 32. } else { 33. // 在没有 open-type=getUserInfo 版本的兼容处理 34. wx.getUserInfo({ 35. success: res => { 36. app.globalData.userInfo = res.userInfo 37. this.setData({ 38. userInfo: res.userInfo, 39. hasUserInfo: true 40. }) 41. } 42. }) 43. } 44. }, 45. getUserInfo: function(e) { 46. console.log(e) 47. app.globalData.userInfo = e.detail.userInfo 48. this.setData({ 49. userInfo: e.detail.userInfo, 50. hasUserInfo: true 51. }) 52. } 53. }) |

Page是一个构造函数，用于向小程序中，注册当前页面。

生命周期函数:

onLoad: 页面加载完成时 触发一次

onReady: 页面渲染完成时 触发一次

onShow: 页面进入前台 触发多次

onHide: 页面进入后台 触发多次

onUnload: 页面被卸载 触发一次

事件函数：

onPullDownRefresh: 下拉刷新

onReachBottom: 上拉触底

onShareAppMessage: 分享时

数据对象：

data

# 数据驱动

## 5.1 插值语法

插值：将JS文件中的数据，插入到页面中。

数据源：JS页面中的Page构造函数所接收的对象的data属性。

|  |
| --- |
|  |

插值语法： {{data的属性名}}

注：{{}}之间开辟的不是一个真正的JS环境，而是一个伪环境。能够执行运算符但是无法执行方法。

注：可以是单层属性，也可以是多层属性

demo:

|  |
| --- |
| 1. <view>{{title}}</view> // 单层属性 2. <view>{{student.username}}</view> // 多层属性 |

## 5.2 setData

有时，我们在操作页面的时候，将data中的数据进行了修改。此时我们希望它能够实时的展示在页面中。

此时，我们可以直接修改data的属性。获取data的方式：先获取data所在的对象，再获取data。获取对象的方式就是在该对象的方法中调用this即可。

demo: 获取对象

|  |
| --- |
| 1. onLoad: function (options) { 2. console.log(this); 3. }, |

输出:

|  |
| --- |
|  |

该对象的data属性就是我们要获取的数据源。

如果直接修改该对象的属性:

|  |
| --- |
| 1. this.data.title = "爱创课堂"; |

此时，数据源中的数据确实会被改变。但是页面不发生变化。

所以我们要通过微信小程序提供给我们的方法来修改数据。就是this.setData.

setData: 该方法用于修改数据源中的数据。

接收一个参数 参数是一个对象

对象的key属性：要修改的数据的key 可以是单层级属性 可以是连续打点的属性

对象的value属性: 修改后的值

demo: 在onLoad事件函数中

|  |
| --- |
| 1. // 3秒之后 修改data.title 2. setTimeout(function() { 3. this.setData({ 4. title: "爱创课堂" 5. }) 6. }.bind(this), 3000) |

注意this的指向问题

## 5.3 数据丢失

所谓的数据丢失指的是，数据源中的数据与页面中显示的数据不一致。原因是因为直接修改了数据源中的数据。没有通过setData修改。

# 组件

## 6.1 简介

在HTML文件中，<div></div>这样的内容被称为标签。

在WXML文件中，<view></view>这样的内容被称为组件。

## 6.2 view组件

该组件等价于DIV。

组件名称：view

组件属性：

hover-class: 当手指触碰组件的时候 生效的类

hover-stay-time: 当hover-class生效之后，停留多久 单位是毫秒值

hover-start-time: 当手指触碰组件之后多久，hover-class生效.是一个延迟效果

hover-stop-propagation: 是否阻止冒泡 如果拥有该属性，就会阻止冒泡。

demo:

|  |
| --- |
| 1. <view hover-class='red h1' hover-stay-time='3000' hover-start-time="3000"> 2. 我是一个view组件 3. <view hover-class="blue" hover-stop-propagation='true'>我是一个view子组件</view> 4. </view> |

## 6.3 text组件

该组件等价于span

组件名称：text

组件属性：

space: 该属性决定空白如何表现 默认情况下是空白折叠现象

nbsp: 一个空格 具体多大要看当前的字号

emsp: 一个中文汉字大小

ensp: 半个中文汉字大小  
decode: 是否转义 true表示转义

seletable: 是否可选文字

emsp、ensp:

|  |
| --- |
| 1. <text>你好你好你好</text> 2. <view></view> 3. <text space="emsp"> 好你好你好</text> 4. <view></view> 5. <text space="ensp"> 好你好你好</text> |

表现:

|  |
| --- |
|  |

decode:

|  |
| --- |
| 1. <view> 2. <text decode="true">&gt;你好&lt;</text> 3. </view> |

表现:

|  |
| --- |
|  |

selectable:

|  |
| --- |
| 1. <text selectable='true' space="ensp"> 好你好你好</text> |

表现：内部的文字可以被长按选中

|  |
| --- |
|  |

## 6.4 rich-text组件

该组件可以允许用户使用HTML中的标签

组件名称：富文本组件

属性：

nodes: 是一个节点集合 或者是一个HTML字符串

HTML：

HTML:

|  |
| --- |
| 1. <rich-text nodes="<div class='red'>你好</div>"></rich-text> |

表现:

|  |
| --- |
|  |

Arr:

|  |
| --- |
| 1. data: { 2. arr: [ 3. { 4. name: "div", 5. attrs: { 6. "class": "red", 7. "style": "font-size: 50px" 8. }, 9. children: [ 10. { 11. type: "text", 12. text: "今天天气热" 13. } 14. ] 15. } 16. ] 17. }, |

表现：

|  |
| --- |
|  |

## 6.5 icon

微信小程序很小，一个小程序最大不超过2M。所以，为了节省空间。就将一些常用的图标制作成组件，嵌入在WXML中。

组件名称：icon组件 图标组件

组件属性：

type： 图标分类 可选值：success, success\_no\_circle, info, warn, waiting, cancel, download, search, clear

size： 图标尺寸

color: 图标颜色

demo：

|  |
| --- |
| 1. <icon type='clear' size='40px' color='black'></icon> |

表现：

|  |
| --- |
|  |

## 6.6 image组件

组件名称：image 图片组件

说明：image在微信小程序中，是有区域的。图片在微信小程序中，与image所占的区域是两个区域。

组件属性：

mode: 裁切模式

可选值：

缩放 scaleToFill 不保持纵横比缩放图片，使图片的宽高完全拉伸至填满 image 元素

缩放 aspectFit 保持纵横比缩放图片，使图片的长边能完全显示出来。也就是说，可以完整地将图片显示出来。

缩放 aspectFill 保持纵横比缩放图片，只保证图片的短边能完全显示出来。也就是说，图片通常只在水平或垂直方向是完整的，另一个方向将会发生截取。

缩放 widthFix 宽度不变，高度自动变化，保持原图宽高比不变

裁剪 top 不缩放图片，只显示图片的顶部区域

裁剪 bottom 不缩放图片，只显示图片的底部区域

裁剪 center 不缩放图片，只显示图片的中间区域

裁剪 left 不缩放图片，只显示图片的左边区域

裁剪 right 不缩放图片，只显示图片的右边区域

裁剪 top left 不缩放图片，只显示图片的左上边区域

裁剪 top right 不缩放图片，只显示图片的右上边区域

裁剪 bottom left 不缩放图片，只显示图片的左下边区域

裁剪 bottom right 不缩放图片，只显示图片的右下边区域

scaleToFill:

image的宽:320 高240

图片的宽: 540 高 258

比例不一致。

|  |
| --- |
|  |

aspectFit：

|  |
| --- |
|  |

widthFix：

|  |
| --- |
|  |

## 6.7 map组件

组件名称：map 地图组件

组件属性：

longitude：经度

latitude: 纬度

scale: 缩放等级 5~18 5最大距离地面最远 18最精确 距离地面很近

demo:

|  |
| --- |
| 1. <!--pages/map/map.wxml--> 2. <map longitude='116.3832502505' latitude='40.10367327010' scale='18'></map> |

表现：

|  |
| --- |
|  |

## 6.8 scroll-view组件

组件名称：scroll-view 滚动组件

组件属性：

scroll-y 指定竖直方向的滚动

scroll-x 指定水平方向的滚动

scroll-into-view 指定子组件的id 被指定的组件会尽可能的显示在scroll-view的第一个位置 注：id不可以是数字

注：因为scroll-view自身默认是被子组件们撑高的，所以如果不设置高度，会与所有子组件的高度和一致。这时是无法滚动的。

demo:

|  |
| --- |
| 1. <scroll-view style="height:400px" scroll-y scroll-into-view="three"> 2. <view>1</view> 3. <view>1</view> 4. <view>1</view> 5. <view>1</view> 6. <view>1</view> 7. <view>1</view> 8. <view>1</view> 9. <view>1</view> 10. <view>1</view> 11. <view>1</view> 12. <view>1</view> 13. <view>1</view> 14. <view>1</view> 15. <view id="two">2</view> 16. <view>1</view> 17. <view>1</view> 18. <view>1</view> 19. <view>1</view> 20. <view>1</view> 21. <view>1</view> 22. <view>1</view> 23. <view>1</view> 24. <view>1</view> 25. <view>1</view> 26. <view>1</view> 27. <view>1</view> 28. <view>1</view> 29. <view>1</view> 30. <view id="three">3</view> 31. <view>1</view> 32. <view>1</view> 33. <view>1</view> 34. <view>1</view> 35. <view>1</view> 36. <view>1</view> 37. <view>1</view> 38. <view>1</view> 39. <view>1</view> 40. <view>1</view> 41. <view>1</view> 42. </scroll-view> |

显示：

|  |
| --- |
|  |

## 6.9 swiper组件

组件名称：swiper 轮播图组件

子组件名称：swiper-item 一项

组件属性：

indicator-dots：决定是否显示小圆点

indicator-color: 决定小圆点的颜色

indicator-active-color: 决定当前小圆点的颜色

autoplay: 自动轮播

interval: 轮播间隔 毫秒值

## 6.10 moveable

组件名称：movable-area 滑动组件

子组件名称：movable-view 滑块

子组件属性：direction 决定滑动方向

可选值：

all：任意滑动

vertical：垂直方向可以滑动

horizontal: 水平方向可以滑动

none: 不滑动

demo:

|  |
| --- |
| 1. <movable-area class="a" > 2. <movable-view class="b" direction="horizontal"></movable-view> 3. </movable-area> |

## 6.11 多媒体组件

audio：音频组件

video：视频组件

camera：摄像机组件

live-player：直播组件

live-pusher：录播组件

# 属性

## 7.1 属性分类

在HTML中，属性分为：

标准属性： W3C规定的每一个属性

通用属性：指的是每一个标签都有的属性

非通用属性：指的是只有一部分标签拥有的属性

自定义属性：用户自己规定的属性 现在都有规范了，推荐以data-开头

在WXML中，属性分为通用属性和非通用属性

通用属性：

id: 定义组件的id

class: 定义组件的类名称

style: 定义组件的样式

hidden: 定义组件的显隐

data-\*: 定义自定义组件

bind-\*: 定义冒泡事件

catch-\*: 定义非冒泡事件

非通用属性：

src、width、height、type等一系列属性

# 事件

## 8.1 事件分类

按照冒泡不冒泡可以分成两类：冒泡事件、不冒泡事件

事件划分：

通用事件：

touchstart 触摸按下

touchmove 触摸并移动

touchend 手指离开

touchcancel 取消事件

tap 轻触

longtap 长按

longpress 长按

transitionend 过渡结束事件

animationstart 动画开始之前触发一次

animationend 动画结束之后触发一次

animationinteraion 每动画一次 执行一次

非通用事件：

form表单的submit事件

## 8.2 绑定事件

绑定事件是通过事件的属性去指定函数来绑定：

|  |
| --- |
| 1. <!--pages/index/index.wxml--> 2. <text bindtap='tap'>pages/index/index.wxml</text> |

bindtap是该组件的属性

tap是js文件中的一个函数。该函数定义在Page所接收的对象上。

JS文件：

|  |
| --- |
|  |

## 8.3 事件对象

事件对象指的是用户在触发事件的时候产生的信息的集合体。

demo:

|  |
| --- |
|  |

changedTouches: 类数组对象 里面装载的是每一根手指的信息

手指对象的属性：

clientX: 距离视口的左边位置

clientY: 距离视口的上边位置

pageX: 距离页面的左边位置

pageY: 距离页面的上边位置

currentTarget: 绑定事件的组件

id：该组件的id

offsetLeft：距离定位父组件的左边距离

offsetTop: 距离定位父组件的上边距离

dataset: 这是一个对象 该对象中的值由该组件的自定义属性组成

timestamp：时间戳 页面加载完毕到事件触发的毫秒值

touches：类数组对象 里面存放的也是手指的相关信息

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |