爱创课堂前端培训

# 微信小程序

第2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22日

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

# 复习

注册成为开发者：去微信公众平台注册并登录，点击小程序，填写邮箱和密码，最后选择类型等获取到appID。

下载并安装微信开发工具，扫码登录，创建项目，进行开发。

目录：

pages：用于存放所有的小程序中使用到的页面。

index: 每一个页面是一个文件夹

内部有四个文件

index.js 配置当前页面的脚本以及当前页面的生命周期函数、初始数据以及事件函数等

Page构造函数：页面构造器

index.json 配置当前页面

index.wxml 当前页面的结构

index.wxss 当前页面的样式

utils: 用于存放工具模块

内部的每一个文件都是JS模块。

支持CommonJS规范。 通过require引入其它模块的内容，通过module.exports暴露内容

app.js: 应用程序的脚本 可以配置生命周期函数以及全局对象

App构造函数：应用程序构造器 用于注册小程序。

app.json: 应用程序的配置

pages: 是一个数组，用于规定当前小程序中一共有多少个页面组成。

window: 配置窗体的内容比如导航的文字、导航的背景、导航的文字样式、背景颜色

tabBar: 配置底部按钮 list是一个数组 数组中的每一项都是一个底部按钮的配置对象 2~5个。

{

pagePath: 当点击该底部按钮时 出现的页面 这里的值必须出现在pages中

text: 文字

iconPath：当底部按钮不被选中时的图标

selectedIconPath: 当被选中时的图标

}

networkTimeout：超时时间

request：普通http请求

connectSocket：websocket的连接时间

uploadFile：上传文件

downloadFile：下载文件

数据驱动： 微信小程序通过数据驱动程序的运行。

微信小程序插值：将JS文件中的data数据源中的数据插入到页面中。

插值语法： {{data的属性名称(可以是多层级)}}

{{}}之间是一个伪JS环境，可以执行运算符，但是无法执行方法

修改数据：我们希望页面上的内容也实时修改。

setData：该方法是页面构造器函数所接收的对象的方法

参数是一个对象

对象的key：要修改哪个属性

对象的value：要修改成什么

数据丢失：指的是数据源中的内容与页面上的内容不一致。

原因：没有通过setData进行修改而是直接通过JS属性操作修改了数据。

双线程：微信小程序是双线程的。

1. 线程用于渲染页面
2. 线程用于执行脚本

组件：在HTML中，标签叫做标签、元素 在WXML中，标签叫做组件。

view：等价于div。

该组件是一个块级组件。

hover-class: 当手指触碰之后 生效的类

hover-start-time: 当手指触碰之后多少毫秒之后hover-class生效

hover-stay-time: 生效多少毫秒

hover-stop-propagation: 是否阻止冒泡

text：等价于span

该组件是一个行内组件

space: 决定空白如何表现

decode: 决定是否转义

seletable: 决定是否可以长按选中

rich-text: 富文本组件

nodes: 可以是一个数组 可以是一个HTML字符串 推荐使用数组

icon: 图标组件

type：图标类型

size: 图标尺寸

color: 颜色

map: 地图组件

longitude：经度

latitude：纬度

scale：缩放等级

image：图片组件 小程序中的图片组件是有区域的。不是完全由内部的图片撑起来的。

mode: 裁切、缩放方式

src: 线上的资源

scroll-view: 滚动组件

scroll-into-view: 值是子组件的id

scroll-x: 可以横向滚动

scroll-y: 可以纵向滚动

必须给scroll-view一个高度 而且高度必须小于所有的子组件之和

swiper：轮播图组件 等价于UL

indicator-dots: 是否有小圆点出现

indicator-color: 小圆点的颜色

indicator-active-color: 当前小圆点的颜色

autoplay: 自动轮播

interval: 轮播间隔

子组件：swiper-item 等价于LI

movable-area：滑动组件

子组件：movable-view

direction: 滑动方向

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：

通用属性：id、class、style、hidden、data-\*、bind-\*、catch-\*

非通用属性：type、src、width等

事件：

通用事件：

touchstart、touchmove、touchend、touchcancel、tap、longtap、longpress、transitionend、animationstart、animationend、animationineration

非通用事件：

submit

事件绑定：

通过组件的属性去绑定事件 值是一个字符串 表示一个函数 该函数必须定义在JS文件中。

bindtap: 绑定一个冒泡的tap事件

catchtap：绑定一个不冒泡的tap事件

事件对象的属性：

changedTouches：手指的坐标信息集合

currentTarget：绑定事件的组件

id: id

dataset：该对象是由该组件的所有自定义属性的属性名和属性值组成

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  
timestamp：当事件发生时，打开页面的计时

# 指令

微信小程序中，提供了“指令”。用于帮助我们进行页面的数据渲染以及控制组件的显示隐藏等。

## 1.1 wx:for

该指令用于循环。

书写的位置，在组件的属性上。值通常是数组或者对象。

index: 表示下标

item: 表示成员

使用方式:

|  |
| --- |
| 1. <view wx:for="{{arr}}" >{{index + item}}</view> |

问题：如果循环嵌套，并且想要在内层循环中使用外层循环的变量，该如何做？

如下：

|  |
| --- |
| 1. <view wx:for="{{obj}}" > 2. <text wx:for="{{arr}}" >{{index + item}}</text> 3. </view> 4. 标红的部分是arr的下标和成员，现在想要让它们变成obj的key和value。 |

解：

|  |
| --- |
| 1. <view wx:for="{{obj}}" wx:for-index="a" wx:for-item="b" > 2. <text wx:for="{{arr}}" >{{a + b}}</text> 3. </view> |

我们可以通过wx:for-index= 和wx:for-item= 来指定循环的下标和成员的变量名。

## 1.2 wx:key

该指令用于指定一个id。方便微信小程序追踪数据的状态。

如果是循环出来的数据，微信小程序无法直接获取到数据的状态。此时，我们需要显式指定。

以下是不指定时：报警告

|  |
| --- |
|  |

以下是指定时：没有任何问题

|  |
| --- |
|  |

## 1.3 wx:if

这是一个判定式的指令

如果wx:if的值是一个“真”值。则组件被显示 否则 组件被隐藏。

demo:

|  |
| --- |
| 1. <text wx:if="{{show}}">pages/if/if.wxml</text> |

此时，show为false，界面中没有任何显示.

# 路由

在微信小程序中，路由指的就是页面之间的切换。

## 2.1 页面的分类

在微信小程序中，页面分两类。

app.json中的pages定义了当前小程序的所有页面。

app.json中的tabBar定义了当前小程序的底部按钮使用的页面。

当pages中的页面数量多于tabBar中使用的页面数量时，两种页面就区分出来了。

在tabBar中的页面：tabBar页面

不在tabBar中的页面：普通页面

## 2.2 组件跳转

根据页面的不同分类，我们要有不同的跳转类型。

根据跳转的触发情况，划分出不同的跳转方式。

第一种：组件跳转

组件名称：navigator

url: 跳转的目标页面的url

open-type: 打开方式

可选值有三个：

navigateTo： 打开新的页面

swtichTab：打开tabBar页面

redirectTo：重定向到新页面

demo: 通过组件跳转到普通页面

|  |
| --- |
| 1. <navigator url='/pages/second/second' open-type='navigateTo'>跳转到第二个页面</navigator> |

注：open-type是什么，要看url的值是什么类型的页面。如果是普通页面，可以使用navigateTo和redirectTo。如果是tabBar中的页面，必须使用swtichTab。

demo: 通过组件跳转到tabBar页面

|  |
| --- |
| 1. <navigator url="/pages/second/second" open-type='switchTab'>跳转到第二个页面</navigator> |

## 2.3 代码跳转

wx是一个对象，是微信小程序中的全局对象。等价于window。

它身上有许多的方法。

**跳转普通页面**

wx.navigateTo(obj) 该方法用于跳转到普通页面

obj是一个对象

url：跳转到的页面的url 该url必须是普通页面的

demo:

|  |
| --- |
| 1. wx.navigateTo({ 2. url: '/pages/second/second' 3. }) |

**跳转tabBar页面**

wx.switchTab(obj) 该方法用于跳转到tabBar页面

obj是一个对象

url：跳转到的页面的url 该url必须是tabBar页面的

demo:

|  |
| --- |
| 1. wx.switchTab({ 2. url: '/pages/second/second' 3. }) |

## 2.4 其它跳转方式

除了navigateTo和switchTab之外，还有redirectTo和navigateBack两种跳转方式。

redirectTo：用于重定向

组件跳转：

|  |
| --- |
| 1. <navigator url="/pages/xxx/xxx" open-type="redirectTo"></navigator> |

代码跳转：

|  |
| --- |
| 1. wx.redirectTo({url: "/pages/xxx/xxx"}); |

navigateBack: 用于后退

组件后退：

|  |
| --- |
| 1. <navigator open-type="navigateBack"></navigator> |

代码后退：

|  |
| --- |
| 1. wx.navigateBack(); |

# 页面的生命周期

页面的生命周期函数：load、ready、show、hide、unload

## 3.1 执行顺序

当页面加载的时候，执行的顺序是：load => show => ready

|  |
| --- |
|  |

当页面隐藏的时候，执行的顺序是：hide

|  |
| --- |
|  |

当页面卸载的时候，执行的顺序是： unload

|  |
| --- |
|  |

## 3.2 navigateTo的特点

当一个页面是被navigateTo打开的时候，必然会执行:

load => show => ready

也就是说，load执行了。也就是打开了一个新的页面。

所以，如果两个页面来回navigateTo，并不是内存中的两个页面切换。而是每一次都打开一个新的页面。

## 3.3 redirectTo的特点

当一个页面是被redirectTo打开的时候，必然会执行：

load => show => ready

同样打开了一个新的页面。

但是不同的是，执行redirectTo的页面会被卸载掉，不会再留在历史记录中了。

## 3.4 navigateBack的特点

当一个页面是被navigateBack打开的时候，必然会执行：

show

没有打开新的页面。

与redirectTo相同，执行navigateBack的页面会被卸载掉。

## 3.5 swtichTab的特点

当一个页面是被swtichTab打开的时候

如果是第一次打开： load => show => ready

如果已经打开过： show

执行switchTab代码的页面，会隐藏：hide

# 下午复习

指令：wx:for 用于循环数组和对象

在生效范围内通过插值语法：{{index}} 使用索引值或者key

在生效范围内通过插值语法：{{item}} 使用成员或者value

指令：wx:for-index 用于自定义索引值名称

指令：wx:for-item 用于自定义成员值名称

指令：wx:key 用于微信小程序跟踪数据状态

指令：wx:if 用于控制某一个组件的显示/隐藏

页面：分为普通页面和tabBar页面

普通页面：在pages中但是没有在tabBar的list中

tabBar页面：既在pages中又在tabBar的list中

路由：

组件跳转：

open-type：通过该属性决定打开类型 其实就是页面的类型

navigateTo：普通页面

switchTab：tabBar页面

代码跳转：

wx.navigateTo({url:""}) 跳转到普通页面

wx.switchTab({url:""}) 跳转到tabBar页面

其它跳转：

redirectTo：普通页面

navigateBack：后退页面

生命周期执行顺序：

navigateTo： 当前页面hide 目标页面load、show、ready

swtichTab：

如果是第一次 当前页面hide 目标页面load、show、ready

如果不是第一次 当前页面hide 目标页面show

redirectTo： 当前页面unload 目标页面load、show、ready

navigateBack：当前页面unload 目标页面show

# 选择城市

wx.getSystemInfoSync() 该方法用于获取系统信息 是一个同步方法

返回值：是一个对象，该对象中拥有相关的系统信息。

demo:

|  |
| --- |
|  |

# 模板

## 5.1 定义

提前准备好的内容，在使用的时候只需要往内部填充不同的数据即可。模板的结构都是提前固定好的。

在微信小程序中，模板需要定义，再使用。

模板分两种：

通过template组件定义的模板

wxml文件本身就是模板

通过template组件定义模板：

|  |
| --- |
| 1. <template name="a"> 2. <button>2</button> 3. </template> |

通过wxml定义模板：

|  |
| --- |
| 1. <form> 2. <view> 3. <label>你好</label><input type="input"></input> 4. </view> 5. <button></button> 6. </form> |

## 5.2 使用

使用模板有三个组件：template、 import、 include

template既可以定义组件又可以使用组件

如果template在同一个页面中定义 在同一个页面中使用

|  |
| --- |
| 1. <template name="a"> 2. <button>2</button> 3. </template> 4. <template is="a"></template> 5. <template is="a"></template> 6. <template is="a"></template> 7. <template is="a"></template> 8. <template is="a"></template> 9. <template is="a"></template> 10. <template is="a"></template> 11. <template is="a"></template> |

通过name定义 通过is使用

import的作用是将外部的模板引入进来：之后就可以通过template is使用了

外部的模板：template1.wxml

|  |
| --- |
| 1. <template name="a"> 2. <button>2</button> 3. </template> |

引入到index.wxml中使用：

|  |
| --- |
| 1. <import src="./template1.wxml"/> 2. <template is="a"></template> 3. <template is="a"></template> 4. <template is="a"></template> 5. <template is="a"></template> 6. <template is="a"></template> 7. <template is="a"></template> 8. <template is="a"></template> 9. <template is="a"></template> |

以上是使用template模板的方式。

如果想要使用wxml定义的模板：我们可以通过include将外部wxml中的内容引入进来。

template.wxml:

|  |
| --- |
| 1. <form> 2. <view> 3. <label>你好</label><input type="input"></input> 4. </view> 5. <button>你好啊</button> 6. </form> |

现在想要将这些内容引入到index.wxml中

|  |
| --- |
| 1. <include src="./template.wxml" /> |

# 跨页面传递数据

## 6.1 通过全局传递

app.js：全局的

|  |
| --- |
| 1. App({ 2. // 在这里定义数据 3. a: { 4. name: "王老五" 5. } 6. }) |

index.js: 某一个页面

|  |
| --- |
| 1. var app = getApp(); 2. app就是全局的App所接收的对象。我们可以在当前页面设置它。 |

second.js: 另外一个页面

|  |
| --- |
| 1. var app = getApp(); 2. 我们可以在另外一个页面中获取它。 |

## 6.2 url传递数据

我们可以通过跳转页面的时候，传递数据。数据写在URL的query上。

第一个页面传递数据：

|  |
| --- |
| 1. <navigator url="/pages/second/second?a=1&b=2&c=3">跳转页面</navigator> |

第二个页面接收数据：

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

输出的内容：

|  |
| --- |
|  |

注：switchTab不能够使用该种方式



## 6.3 通过本地存储对象传递数据

微信小程序也提供了本地存储对象。

本地存储对象的容量是5M。也是永久存储。

设置本地存储：异步设置

|  |
| --- |
| 1. wx.setStorage({ 2. key: '存储时的键名称', 3. data: '存储的值', 4. success: "回调函数" 5. }) |

获取本地存储内容：异步获取

|  |
| --- |
| 1. wx.getStorage({ 2. key: "要提取什么", 3. success: function(e) { 4. e.data是提取出来的数据 5. } 6. }) |

删除本地存储的内容：异步

|  |
| --- |
| 1. wx.removeStorage({ 2. key: 要删除什么 3. success: 成功时的回调函数 4. }) |

清空本地存储的内容：异步

|  |
| --- |
| 1. wx.clearStorage({ 2. success: function() {} 3. }) |

以上四个方法都有同步方法，方法名称之后加Sync

# 新单位

微信小程序提供了rpx单位。

1 rpx 等于1/750手机宽度

|  |
| --- |
| 1. .rect { 2. width: 750rpx; 3. height: 750rpx; 4. } |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |