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25期

讲师：彭帅伟

日期：2019年9月21日

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

目录

[微信小程序](#_Toc32767)

[复习：](#_Toc945)

[一、 微信小程序](#_Toc7819)

[1.1 容器组件](#_Toc28635)

[1.1.1 scroll-view](#_Toc6981)

[1.1.2 swiper](#_Toc27687)

[1.1.3 movable-area](#_Toc11409)

[1.1.4cover-view](#_Toc3287)

[1.2引入文件](#_Toc5994)

[下午复习：](#_Toc1452)

[1.3表单组件](#_Toc8288)

[1.4 路由](#_Toc12684)

[1.4.1 页面分类](#_Toc25893)

[1.4.2 堆栈](#_Toc13359)

[1.5 模板](#_Toc27648)

[1.6 引入文件](#_Toc28273)

[1.7 wxss](#_Toc24417)

[1.7.1 rpx](#_Toc508)

[1.7.2 flex](#_Toc24410)

[1.8 开放组件](#_Toc18271)

[1.9 web-view](#_Toc24546)

[1.10 wxs](#_Toc29141)

# 复习：

目录部署：

app.json: pages window tabBar requestTimeout...

app.js

onLauch onShow onHide onError

wxss

全局方法:

App：创建应用程序

getApp: 获取应用程序

Page: 创建页面

onLoad onReady onShow onHide

getCurrentPages: 获取当前页面信息

数据驱动：

1 在脚本文件中的data中定义数据

2 在视图模板中使用插值语法

插值语法： {{}}

插值语法中是伪js环境，只能够做一些简单的操作，三元运算，加减乘除， 对象的点语法，数组中括号

数据更新：

通过this.setData方法更新数据

数据丢失：

所有的数据都存在data中，如果直接更新data中的数据，视图没有变化，此时就称为数据丢失

需要通过this.setData方法更新数据即可避免

属性：

分为共有属性和特有属性

共有属性：

id

class

style

hidden

data-\*

bind\*|catch\*

事件：

分为两类：

1 公有事件 2 特定事件

按照是否冒泡又分为两类：

1 冒泡事件

2 非冒泡事件

组件：

view 容器组件

text 定义内容，不会独占一行

rich-text: 富文本组件

1 html 2 nodes 3 str

image: 定义图片组件

icon: 字体图标

map: 定义地图组件

video: 视频组件

camare: 开启摄像头

live-player: 直播组件

live-pusher: 推送组件

指令：

条件指令：

wx:if=”statements”

循环指令：

wx:for=”data”

成员默认变量名称 item 默认索引值： index

更换成员变量名称

wx:for-item= ‘name’

更换索引值：

wx:for-index=”name”

# 微信小程序

## 容器组件

### 1.1.1 scroll-view

允许我们定义可以滚动的容器

scroll-x scroll-y 滚动的方向

scroll-into-view 用于初始化滚动的位置 属性值是某个组件的id

|  |
| --- |
| 1. <scroll-view class="demo" scroll-y scroll-into-view="a20"> 2. <view wx:for="{{30}}" wx:key="this" id="a{{index}}">{{item}}</view> 3. </scroll-view> |

### 1.1.2 swiper

组件名称： swiper

indicator-dots： 用于开启小圆点

indicator-active-color: 当前小圆点的颜色

autoplay： 自动播放

interval： 轮播时间

circular： 定义是否衔接

作用：制作轮播图

子组件

swiper-item

|  |
| --- |
| 1. <swiper indicator-dots="true" indicator-active-color="red" autoplay="true" interval="1000" circular="true"> 2. <swiper-item> 3. <image src="https://www.baidu.com/img/bd\_logo1.png"></image> 4. </swiper-item> 5. <swiper-item> 6. <image src="https://www.baidu.com/img/bd\_logo1.png"></image> 7. </swiper-item> 8. <swiper-item> 9. <image src="https://www.baidu.com/img/bd\_logo1.png"></image> 10. </swiper-item> 11. <swiper-item> 12. <image src="https://www.baidu.com/img/bd\_logo1.png"></image> 13. </swiper-item> 14. <swiper-item> 15. <image src="https://www.baidu.com/img/bd\_logo1.png"></image> 16. </swiper-item> 17. </swiper> |

### 1.1.3 movable-area

movabel-area 用于定义可移动区域

direction： 定义滑动方法

vertical： 垂直方法的滑动

hrizontal: 水平方向的滑动

all: 所有方法

node: 不滑动

inertia： 是否带有惯性

friction: 摩擦系数 值越大 摩擦力越大

damping： 阻尼系数

子组件：

movable-view 只能在movebel-area中滑动

|  |
| --- |
| 1. <movable-area class="parent"> 2. <movable-view class="child" direction="all" inertia="true" friction="2" damping="10"></movable-view> 3. </movable-area> |

### 1.1.4cover-view

在小程序中，总有一些组件是渲染在最顶层，我们不能通过z-indx定义覆盖顺序

cover-view可以实现定义覆盖在原生组件之上的文本视图。

|  |
| --- |
| 1. <map> 2. <cover-view> 3. <cover-image src="/images/logo.png"></cover-image> 4. </cover-view> 5. <cover-view> 6. <cover-image src="/images/logo.png"></cover-image> 7. </cover-view> 8. </map> |

## 1.2引入文件

在小程序的脚本中，我们使用commonjs规范。

引入文件用require方法

暴露接口用

exports

module.exports

注意：引入文件的时候，要使用相对地址

## 下午复习：

scroll-view 定义可以滚动的容器

scroll-x scroll-y 滚动的方法

scroll-into-view 初始化滚动的位置

swiper: 用于制作轮播图

indicator-dots： 用于开启小圆点

indicator-active-color: 当前小圆点的颜色

autoplay： 自动播放

子组件： swiper-item

movable-area: 用于定义可移动的区域

direction：移动的方法

vertical: 垂直方法

horizontal: 水平方法

all: 所有方法

子组件： movebele-view

cover-view: 可以渲染在一些源生组件之上

cover-image

引入文件：

在小程序的脚本文件中，使用的是commonjs规范

引入文件使用require

暴露接口：

export

module.exports

## 1.3表单组件

label 定义标题

for 用于关联控件

input 输入框

placeholder: 定义显示内容

password: 以密码的形式显示文字

form 定义表单控件

可以绑定submit事件 所有的数据可以在e.detail.value中获取

checkbox 定义多选框

一组多选框 必须放入checkbox-group组件中

radio 定义单选框

一组单选框 必须放入radio-grouo组件中

button 定义按钮

type 定义类型

primary

warn

form-type: 定义提交或者重置类型

其它组件：

进度条： progress

开关： swich

滑块： slider

选择器：picker

mode 定义类型

picker-view： 自定义选择器

子组件：picker-view-column

demo:

|  |
| --- |
| 1. <!-- 进度条 --> 2. <progress percent="10" show-info></progress> 3. <!-- 开关 --> 4. <switch>开关</switch> 5. <!-- 滑块组件 --> 6. <slider min="0" max="100" step="10"></slider> 7. <!-- 选择器 --> 8. <picker mode="time">时间</picker> 9. <picker mode="date">日期</picker> 10. <picker-view style="height: 100px; width: 100%;"> 11. <picker-view-column> 12. <view wx:for="{{colors}}" wx:key="this">{{item}}</view> 13. </picker-view-column> 14. <picker-view-column> 15. <view wx:for="{{colors}}" wx:key="this">{{item}}</view> 16. </picker-view-column> 17. </picker-view> |

## 1.4 路由

在小程序中切换页面的方式有两种：

第一种是通过navigator组件实现页面之间的切换

url 定义页面地址

并且可以在url的后面携带query数据， 在跳转到的页面中的load事件中的参数获取到

尽量写绝对路径

open-type 打开页面的方式

默认值是navigateTo

第二种是通过navigator方法，实现页面的切换

### 1.4.1 页面分类

在小程序中的页面分为两类

第一类普通页面

在pages里面定义

第二类在tabBar里面定义的

这两类页面的切换方法是不同的，方式有四种：

第一种 open-type = “navigateTo”

用于打开一个普通页面

第二种 open-type = “switchTab”

用于切换tab页面

第三种 open-type = “navigeteBack”

用于返回上一个页面

第四种 open-type = “redirect”

重定向到一个页面

### 1.4.2 堆栈

小程序也是通过堆栈管理页面的

navigateTo

保证原有堆栈不变，新的页面进入堆栈中

当前页面hide ， 新的页面是load和show

switchTab

清空原有堆栈，新的页面进入堆栈中

当前页面hide, 新的页面是load和show

navigeteBack

保证原有的堆栈不变，新的页面从堆栈中弹出

当前页面unload，新页面是show

redirect

保证原有的堆栈不变， 新的页面从堆栈中弹出，进入新的堆栈

当前页面unload, 新页面是load和show

demo:

|  |
| --- |
| 1. <!-- 定义navigator组件 --> 2. <navigator url="/pages/form/form" open-type="navigateTo">进入form页面</navigator> 3. <!-- 默认值就是navigateTo --> 4. <navigator url="/pages/form/form" >进入form页面</navigator> 5. <!-- tab页面 --> 6. <navigator url="/pages/index/index" open-type="switchTab">进入index页面</navigator> |

## 1.5 模板

通过template用于定义模板组件

name属性 定义组件的名称

is属性 使用模板

模板可以作为组件来使用，但是不会渲染在页面中

与block的区别，模板是可以复用的

|  |
| --- |
| 1. <!-- 定义模板 --> 2. <template name="hello"> 3. <view>你好，世界</view> 4. </template> 5. <!-- 使用模板 --> 6. <template is="hello"></template> 7. <template is="hello"></template> |

## 1.6 引入文件

在js文件中引入文件的方式通过require引入

在wxml中引入文件的方式是通过两个组件引入

import 组件 引入文件

include 组件 引入文件

通过src引入文件 相对路径

它们的区别：

import 引入模板内容

include 引入文件中的元素

|  |
| --- |
| 1. <!-- 引入模板 --> 2. <import src="./common.wxml"></import> 3. <template is="demo"></template> 4. <template is="demo"></template> 5. <template is="demo"></template> 6. <!-- 引入视图 --> 7. <include src="./common.wxml"></include> 8. <!-- <template is="demo"></template> --> |

## 1.7 wxss

在小程序的wxss文件中，引入css3.0的规范， 因此css中的特性，在小程序中都可以使用

### 1.7.1 rpx

rpx是一个单位，是对px单位的一个升级，可以实现响应式

在小程序中，将分辨率分为750份了，1rpx代表1份

例如：

iphone6的分辨率是375px, 真实像素是750px

在iphone6下 1rpx = 1px

但是dpr是2 ，所以 1px = 2rpx

### 1.7.2 flex

flex也可以实现响应式布局

要为元素设置： display: flex;

还可以为子元素设置固定的宽高，还可以设置放缩

flex-direction: 布局方向

row: 从主轴左侧到右侧渲染

row-reverse: 从主轴右侧到左侧渲染

column: 从侧轴的上方到下方渲染

column-recerse: 从侧轴的下方到上方渲染

justify-content: 设置成员在主轴上对齐方式

flex-start: 从主轴的起点对齐

flex-end: 从注重的终点对齐

center: 居中对齐

sapce-around: 成员之间均分空白，两边的成员空白减半

space-between: 成员之间均分空白，两侧的成员贴边

space-evenly: 成员之间均分空白

align-items: 设置成员在侧轴上的对齐方式

flex-start: 从侧轴的起点对齐

flex-end: 从侧轴的终点对齐

flex-center: 垂直居中

align-content: 设置多行 设置成员在侧轴的对齐方式

flex-start: 从主轴的起点对齐

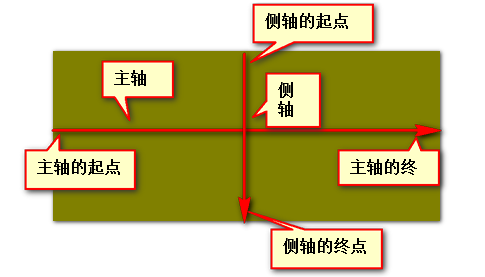
flex-end: 从注重的终点对齐

center: 居中对齐

sapce-around: 成员之间均分空白，两边的成员空白减半

space-between: 成员之间均分空白，两侧的成员贴边

space-evenly: 成员之间均分空白



|  |
| --- |
| 1. <view class="demo-px"></view> 2. <view class="demo-rpx"></view> 3. <view class="parent"> 4. <view class="item">111</view> 5. <view class="item">222</view> 6. <view class="item">333</view> 7. </view> |

## 1.8 开放组件

开放组件是为了开放用户信息

在目前我们可以通过wx.getUserInfo获取用户的信息

但是，小程序推荐我们使用open-data组件来获取用户的信息

通过type 获取用户信息

语法： type + userInfo信息

|  |
| --- |
| 1. <open-data type="userAvatarUrl"></open-data> 2. <open-data type="userNickName"></open-data> |

## 1.9 web-view

小程序也是在html、css、js基础之上衍生的一个框架，也是通过webview渲染

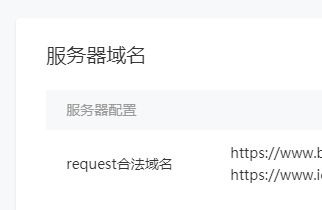
因此，小程序提供了web-view组件用于引入html页面

src定义引入的html页面地址

必须是https协议

引入的时候必须要在后台配置

要关闭https证书校验



搭建HTTPS服务器：

HTTP协议是一个无状态的协议，不会对请求进行校验，因此可能会注入一些广告，或者会被拦截，所以HTTPS协议就出现了，会对请求进行校验，所以更加安全

http默认端口号： 80

https默认端口号： 443

|  |
| --- |
| 1. <web-view src="https://www.baidu.com"></web-view> 2. <web-view src="https://localhost:3001/web/"></web-view> |

## 1.10 wxs

wxs组件允许我们定义一个可以执行js代码的空间模块

使用方式：

第一步 要创建一个以.wxs后缀名称的文件

在该文件中只能使用ES5规范

第二步 在页面中通过wxs组件定义可以执行js的代码空间模块

通过src引入wxs文件

第三步 通过wxs中的module属性定义空间模块的名称

第四步 通过插值语法使用向外暴露的接口 （通过定义的module属性值调用）

向外暴露接口是通过commonjs规范

|  |
| --- |
| 1. <wxs module="demo" src="./demo.wxs"></wxs> 2. <view>{{demo.num}}</view> 3. <view>{{demo.getMax(arr)}}</view> |

# 作业



|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |