# day1 快速进入小程序

1. 课程内容：

* 小程序 5天 1-5
* mpvue 3天 6-8
* 小游戏 2天 9-10
* react-native 4天 11-14
* 答辩 1天 15

1. 微信公众平台产品区别：

mp.weixin.qq.com

* 订阅号 每天推送1次
* 服务号 每月推送1次
* 企业号
* 小程序
* 小游戏

1. 小程序的概念、由来

* mini Program
* 触手可及、用完及走、随处可见
* 无需安装和卸载

1. 小程序优缺点：

* 优点：

成本比app低、开发难度低

* 缺点：

大小不能超过2M、

功能没有原生APP强大

发布需要腾讯审核，行业有限制

对个人开发者限制大

1. 与原生APP区别

* App用户25亿，小程序8亿
* 小程序仅限于微信接口。
* 小程序：低频、非刚需
* 上线：App产品商店 腾讯微信公众平台
* 下载安装

1. 小程序注册流程
2. 小程序后台讲解

* 版本管理
* 成员管理
* 小程序设置
* 开发设置

APPID

合法域名

1. 开发者工具

* 菜单栏、工具栏、模拟器、编辑器、调试器

1. 小程序新建步骤
2. 小程序目录结构
3. 全局配置文件

app.json tabbar、pages、window（enablePullDownRefresh）

局部配置文件

删除所有文件，新建一个项目

1. 小程序场景值
2. 腾讯历史

* 小时候

马化腾，海南小渔村，天文学家

大四->股票交易程序->卖了5W元

* QQ的诞生
* 95年，4条电话线+8台电脑->BBS（当时上网的有100人，其中有雷军）
* 网络+寻呼机🡪 网络寻呼机
* 以色列ICQ 🡪 腾讯OICQ—>因为侵权，改成了QQ
* 服务器费用昂贵：10W元🡪1W用户
* 没钱：卖公司🡪桌椅板凳

跟亲戚朋友借钱🡪 70W （一哥们没要股份，不然进了福布斯前10名了）

降薪🡪自己拿2500元/月（相当于大学毕业生）

* 投资：
* IDG 王树🡪合同从美国邮寄过来，450W（1周后纳斯达克5000点降到1000点）
* 李嘉诚二公子
* 钱又花完了，跟互联网大佬借钱，吃闭门羹

新浪 王志东

搜狐 张朝阳

雅虎中国

联想

金蝶

南非IMH 🡪 $6000W

* 找出路

168声讯台

QQ游戏虚拟道具🡪老外取经 🡪 为付费用户提供增值服务，付费养活免费

* 上市

美国 🡪ebay、amazon、MSB

香港上市 🡪 刘炽平

* 国内杀伐

网易泡泡 2004年的微信

米聊 雷军，硬件

微信 朋友圈、语音、视频通话

10086 飞信 🡪QQ打通

腾讯找联通🡪闭门羹

* 四面出击

拍拍网 --》马云淘宝

搜索 🡪百度

浏览器

邮箱

门户网站

买下Foxmail:邮件🡪高端

断点续传、网盘慢

腾讯QQ堂 盛大泡泡堂（盛大分心，做文学、音乐、旅游）

* 3Q大战

腾讯QQ医生 360杀毒软件

腾讯找人公证：金山、搜狗、百度

工信部：360败诉，赔偿￥500W

# day2 小程序基础语法

1. vscode插件

wechat-snippet 代码提示

wxapp-helper 导航栏右键Page

minapp wxml高亮

1. app.js中的APP()函数
2. 生命周期函数
3. getApp获取全局APP实例

|  |
| --- |
| var myapp = getApp()  console.log(myapp.globalData) |

1. Page() 注册页面实例
2. getCurrentPage() 获取当前页面栈
3. wxml语法

* 插值表达式 {{}}
* 运算

字符串运算

算数运算

三元运算

相对路径、绝对路径

if判断

|  |
| --- |
| <view wx:if="{{}}"> </view>  <view wx:elif="{{}}"> </view>  <view wx:else> </view> |

* 组件属性：id、class、data-、style、hidden、bind catch
* 列表渲染

wx:for

wx:key

wx:for-item

wx:for-index

* 条件渲染

wx:if

block与view区别

wx:if与hidden区别

1. wxss样式

* 尺寸转换
* 尺寸单位：rpx
* 内联样式、外部样式、样式导入

|  |
| --- |
| @import "../demo/demo.wxss" |

import 引入模板

include 引入模板之外的其他内容

* 选择器

.class

#id

element,element

::after

::before

* 全局样式、局部样式

1. 事件系统

* 绑定事件函数
* 定义事件函数

非箭头函数：指向其所在函数的对象🡪 调用使用时所在函数

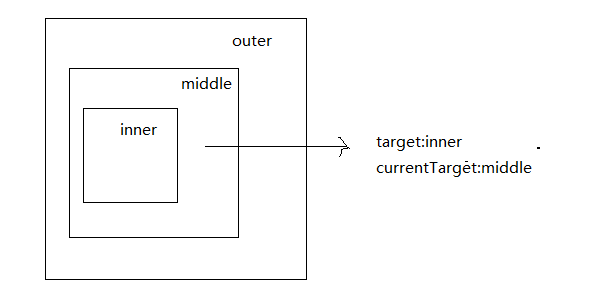
箭头函数 ：没有自己this，指向外面（指向定义时所在对象）

|  |
| --- |
| myFun:function(){......}  myFun(){......}  myFun:()=>{......} |

* 冒泡

bind 冒泡

catch 阻止冒泡



* 事件详解

touchstart

touchmove

touchcancle 来电、弹窗🡪 动作被打断

touchend 触摸结束

tap 点击后马上离开

longpress 超过350ms

transitionend

transitionstart

# Day3 组件及模块化

1. js逻辑文件

* 全局函数：定义、调用

普通

es6语法

箭头函数

* 全局变量：定义、获取
* data初始化数据

1. 视图容器组件

* 什么是组件
* 组件属性：id、class、data-、style、hidden、bind catch
* view视图容器

hover-class

hover-stop-propagation 阻止样式向上冒泡

hover-start-time

hover-stay -time

* flex布局

display

flex-direction

justify-content

align-items

flex

* swiper
* swiper-view
* scroll-view

scroll-x 🡪 父：white-space:nowrap;

子：display:inline-block;

* 自定义swiper

|  |
| --- |
| data:{currentIndex:0} |

|  |
| --- |
| <swiper  current=""  bindchange=""  >  ...  </swiper> |

|  |
| --- |
| <view  class="{{currentIndex==0?'active':''}}"  bindtap='myTap'  id="0"  >  ...  </view> |

1. 基础内容组件

* icon
* text
* rich-text:nodes、space（ensp emsp nbsp）
* progress

percent、show-info、border-radius、font-size、stroke-width、active-color

1. 表单组件

* button:size、type、form-type、

open-type="getUserInfo"

bindgetuserinfo="getInfo"

* checkbox
* checkbox-group
* radio
* radio-group
* input：type、confirm-type
* switch
* slider
* form
* picker

1. 导航组件

* navigator

target

url

1. 多媒体组件

* image
* audio
* video

1. 模块化

* module.exports

|  |
| --- |
| exports.xxx=xxx  module.exports.xxx=xxx  module.exports={} |

* require

1. 自定义组件

* component目录

json文件配置

|  |
| --- |
| {  "component": true  } |

* 参数类型：type、string、number、boolean、object、array、null
* 父子组件
* slot

1. 引用

* 引用模板 <import src="header.wxml">
* 引用非模板 <include src="header.wxml">
* 引用样式 @import "header.wxss"
* 引用组件

|  |
| --- |
| {  usingComponents:""  } |

# Day4 小程序API

1. 路由

* wx.navigateTo
* wx.navigateBack
* wx.redirectTo
* wx.switchTab
* wx.reLaunch

1. 登录

* wx.login 🡪 button 🡪 open-type:getUserInfo
* wx.checkSession
* wx.getUserInfo

1. 缓存

* wx.setStorage(object objetc)
* wx.getStorage(object object)
* wx.removeStorage
* wx.clearStorage

1. map
2. cameara
3. 媒体

* wx.createInnerAudioContext
* 视频
* 录音
* 拍照

1. 网络

* wx.request

1. 缓存
2. 项目：童话故事

* 欢迎页
* 分类页面、列表页面、详情页面
* 个人中心
* 下拉刷新
* 触底加载更多

# Day5 部署及上线

1.上传

2.提交审核

3.购买服务器

* 购买域名
* 备案
* 解析

4.云开发

* 云函数条件

node

npm

安装服务端SDK

|  |
| --- |
| npm install –save wx-server-sdk@latest |

* 云数据库
* 云存储

5.用云开发实现猫眼电影

要求：用云函数从云数据库中读数据；从云存储中获取图片

* + 列表页
  + 详情页

# Day6 mpvue基础

1. 概念

vue.js in mini program

* swan 百度智能小程序
* tt 今日头条
* wx 微信小程序
* my （蚂蚁）支付宝小程序

MVVM ：model、view、viewModel（视图模型-Vue实例）

1. 安装
2. npm install vue-cli -g 下载vue脚手架
3. vue init mpvue/mpvue-quickstart my-project 初始化项目
4. cd my-project 进入项目根目录
5. npm install 根据package.json安装项目依赖包
6. npm start || npm run dev 启动初始化项目
7. 文件、目录结构

* main.js 入口文件

|  |
| --- |
| productTip 提示信息  mpType 区分组件（应用级）  new Vue(App) 生成实例 |

* package.json 包配置文件，项目基本描述、依赖的第三方库
* project.config.json 项目配置文件，appid、代码主目录结构
* src

src/components

src/pages

src/utils

src/app.json

src/App.vue

src/pages/index/index.vue

src/pages/index/main.js

src/pages/index/main.json

* static目录
* build目录：项目编译打包目录，包含：node.js脚本、webpack配置文件
* dist

common

pages

app.js

app.json

app.wxss

* config目录 开发环境、生产环境的不同配置
* main.js 入口文件：声明组件类型、挂载组件

|  |
| --- |
| import Vue from 'vue'  import App from './App'  Vue.config.productionTip = false //提示  App.mpType = 'app' //应用级  const app = new Vue(App) //实例化  app.$mount() //挂载 |

3.注意事项

新建页面后，运行npm run dev

每个组件都要用$mount挂载当前组件

不能用bind，用@

1. mpvue生命周期

* beforeCreate
* created
* beforeMount
* mounted
* beforeUpdate
* updated
* beforeDestroy
* destroyed

1. mpvue语法

* 数据绑定：

|  |
| --- |
| v-bind:src=""  :src="" |

* 事件监听：

|  |
| --- |
| v-on:click=""  @click="" |

* 计算属性：

|  |
| --- |
| computed:{  fullName(){  return " "  }  },  watch:{  firstName:function(newVal,oldVal){  …  }  } |

循环：

|  |
| --- |
| {{username}}  v-for:"(item,index) of data"  @click=""  #传参  :data-index="index"  @click="handleClick(1)"  v-on:click=" "  v-bind:key="index"  v-bind:src="item.pic"  v-show="isShow"  v-if="isShow"  v-model="name"  v-text="" #text format  v-html=" " |

1. 备注：

* 关闭eslint；build/webpack.base.conf.js
* 组件需挂载 $mount()
* 新建页面需 npm run dev
* 不能用bind+事件名，而要用@click、v-on:click

1. 当当商城

* 详情页：

|  |
| --- |
| books、id、detail{} |

* 购物车页面：

|  |
| --- |
| data{  cart:[],  total:0,  isCheckAll:true  } |

# day7 Nodejs+mongodb实现后台

1.express安装

|  |
| --- |
| npm install express --save |

2.express配置路由

|  |
| --- |
| app.get('/search', function(req, res) {  res.send('搜索');  //?keyword=华为手机&enc=utf-8&suggest=1.his.0.0&wq  })  app.get('/login', function(req, res) {  res.send('登录');  }) |

1. express中ejs的使用

* 安装
* 使用

指定模板位置

引入模板

绑定数据

模板判断

循环数据

后缀改为.html

1. body-parser中间件

* 安装
* 使用

获取post过来的数据

1. 操作MongoDb

* 安装

|  |
| --- |
| npm install mongodb --save-dev |

* 连接数据库

|  |
| --- |
| MongoClient.connect(shujukuURL, function(err, db) {  ……  } |

* 查找

|  |
| --- |
| db.collection('user').find(); |

* 增加数据

|  |
| --- |
| db.collection('user').insertOne({ …… }) |

* 修改

|  |
| --- |
| db.collection('user').updateOne({"\_id":ObjectID(id)}, {  "name": name,  "age": age,  "score": {  "shuxue": shuxue,  "yuwen": yuwen  }  },function (err, results) {  console.log(results);  db.close();  res.redirect('/'); /\*路由跳转\*/  res.end('end');  }) |

* 删除

|  |
| --- |
| db.collection('user').deleteOne({"\_id":ObjectID(id)},function(error,result){  …  } |

1. 项目：实现商品后台管理系统

实现登录、退出功能

实现商品增删改查

# Day8 mpvue实现商城

1. 首页

* 轮播图
* 商品列表

1. 详情页

* 轮播图
* 标题
* 介绍
* 添加到购物车

1. 购物车

* 删除商品
* 修改数量
* 计算价格
* 全选/取消全选

# Day9 微信小游戏开发

1.canvas回顾

|  |
| --- |
| var c=document.getElementById("myCanvas");  var ctx=c.getContext("2d");  ctx.strokeStyle="#0000ff";  ctx.strokeRect(20,20,150,100);  #实心  fillStyle="#ccc";  ctx.fillRect(0,0,400,400);  ctx.fillText("hello",10,50);  #空心  ctx.strokeRect(50,50,100,100)  ctx.strokeText("hello",0,40);  #图片  var img=new Image()  img.src="./001.jpg";  image.onload=function(){  ctx.drawImage(img,0,0,100,100)  } |

* 绘制矩形
* 蓝色填充
* 清空矩形
* 旋转矩形
* 缩放矩形
* 绘制文本
* 绘制图像
* 绘制视频

2.面相对象

* 类
* 继承

1. 暴露

* node common.js

|  |
| --- |
| exports.xxx=xxx  module.exports.xxx=xxx  module.exports={} |

* type=module es6语法

不存在arguments、require、module、exports、\_\_filename、\_\_dirname

|  |
| --- |
| export default class demo{}  export default demo;  export {demo} |

1. 小游戏垃圾回收
2. 小游戏账号注册
3. 后台介绍、版本控制、appID
4. 初始化项目
5. 目录介绍
6. 实现官方文档中飞机游戏

# Day10 微信小游戏开发

* + - 1. 框架设计
* 触发思想
* 封装、模块化思想
* 场景思想
  + - 1. 几个概念
* 主循环
* 当前场景
* 角色
  + - 1. 摇骰子
* 开始页
* 结算页
* 排行榜
  + - 1. 发布、提交审核

# Day11 React Native环境安装

简介

* 历史由来
* 概要原理

环境搭建

* 安装jdk、jre

jdk Java development kid（Java开发工具包）

jre java runtime environment（java运行环境）

* 设置淘宝镜像
* 设置环境变量
* 安装react native命令行工具

|  |
| --- |
| npm install -g react-native-cli |

* 安装android studio
* 选中sdk platform
* 选中sdk tools

新建项目

* 初始化项目

|  |
| --- |
| react-native init FirstApp |

* 用android studio 构建项目（小锤子）：下载依赖

项目目录

android

ios

node\_modules

.babelrc

.buckconfig

.flowconfig

.gitattributes

.gitignore

.watchmanconfig

app.json 配置了name和displayName。

index.js Android 和 Ios 入口文件

package.json 定义项目所需各种模块

7.安装雷电模拟器

8.运行app

模拟器设置ip

9.adb命令：链接模拟器/手机

|  |
| --- |
| adb version  adb connect  adb devices  adb start-server  adb kill-server |

# Day12 React Native 目录及基本组件

1.项目目录介绍

index.js

app.json

app.js

2.Text组件

3.View组件

4.StyleSheet组件

5.Flex布局

flexDirection:

flexWrap: nowrap | wrap | wrap-reverse;

justifyContent: flex-start | flex-end | center | space-between | space-around;

alignItems:flex-start|flex-end|center| baseline

6.Image组件

本地图片 source={ require("./images/cat.png")}

网络图片 source={{uri:"…"}}

7.ImageBackground组件

8.TextInput

9.ScrollView

10. TouchableHighlight 背景变黑

TouchableOpacity 可调整大小

TouchableNativeFeedback 波纹

TouchableWithoutFeedback 背景无变化

1. Alert
2. 生命周期
3. 屏幕宽度

|  |
| --- |
| Dimensions.get("window").width; |

1. 练习

吹气球

# Day13 React Native之FlatList

1.props属性

父组件传值： <MyView name="优就业"/>

子组件获取值：this.props.name

2.state状态

3.setState更新状态

4.TextInput组件

* 属性：

value

placeholder

placeholderTextColor

password

maxLength

autoFocus

* 方法

onBlur

onChange

onChangeText

onFocus

5.Button组件

onPress

title

accessibilityLabel

color

disabled

6.TouchableCapacity组件

7.FlatList组件

1. ItemSeparatorComponent：分割线组件，
2. ListFooterComponent：结尾组件
3. ListHeaderComponent：头组件
4. data：列表数据
5. horizontal：设置为true则变为水平列表。
6. numColumns：列数 组件内元素必须是等高的,无法支持瀑布流布局
7. columnWrapperStyle：numColumns大于1时，设置每行的样式
8. getItemLayout：如果我们知道行高可以用此方法节省动态计算行高的开销。
9. refreshing：是否正在加载数据
10. onRefresh：设置此属性需要一个标准的 RefreshControl 控件，刷新数据
11. renderItem：渲染每个组件
12. onViewableItemsChanged：当一个新的Item渲染或者隐藏 的时候调用此方法。参数：info: {viewableItems: Array, changed: Array} viewableItems：当前可见的Item，changed：渲染或者隐藏的Item。
13. scrollToEnd({params?: ?{animated?: ?boolean}})：滚动到末尾，如果不设置getItemLayout属性的话，可能会比较卡。
14. scrollToIndexparams: {animated?: ?boolean, index: number, viewPosition?: number}：滚动到制定的位置
15. scrollToOffset(params: {animated?: ?boolean, offset: number})：滚动到指定的偏移的位置。

8.Fetch请求数据

# Day14 react native 导航

1.refreshControl下拉刷新

2.Flatlist中的onRefresh

FlatList中的refreshing

3.onEndReached上拉刷新

4.React-Navigation导航

5.打包APK