

```
/*
*@ 初识小程序、目录结构、项目配置、场景值、逻辑层、基础语法
*@ 张斌 （php 全栈开发 微信产品开发：公众号 微信小程序 ）
*@ time 2021/12/17
**/
```

一、阶段概述

1.课程安排

课程市场：14+1

课程安排：

微信小程序：5

uniapp：5

数据可视化：4

答辩：1

2.项目展示

小u商城

数据可视化

3.阶段目标

- 1、培养学员能力独立完成小程序原生开发
- 2、培养学员能够独立完成使用**uni-app**多端框架进行宽平台开发，同时掌握与服务端配合工作的能力
- 3、培养学员能力独立查看手册，完成自我提升的能力
- 4、学员能够掌握数据可视化的原理，能够独立完成数据可视化图表的开发

4.学习方法

- 1.多练习，每天按时完成作业及课堂代码练习
- 2.保存好课堂笔记，多背诵
- 3.多参考官方手册
- 4.多横向对比，对比vue语法，对比react语法，记忆力更深刻，理解起来更容易

二、微信开发简介（了解）

1.微信开发概述

概述

微信开发即将企业信息、服务、活动内容通过微信网页的方式进行表现，用户通过简单的设置，就能生成微信网站（比如微信订阅号、服务号 企业账号、微信小程序等）。

微信对外提供了例如聊天、支付、分享、收藏等功能，同时还提供了丰富的封装好的接口，开发者利用这些接口和功能，写入程序中，进行的开发。

目的

- (1) 企业开发的需要,使自己更加符合企业发展的需求
- (2) 个人（开发者）的发展以及技能的提升
- (3) 发展前景大、生态系统丰富

平台

- (1) 微信开放平台（服务端使用）
- (2) 微信公众平台（前端使用）

2.微信开放平台

概述

微信开放平台是微信对外提供微信开发接口的一个平台，这些开发出来的微信接口，供第三方的网站或App使用,使用户可将第三方案程序的内容发布给好友或分享至朋友圈，第三方内容借助微信平台获得更广泛的传播。

平台地址：<https://open.weixin.qq.com/>

微信开放平台提供的能力

微信分享、微信支付、微信登录、微信收藏、微信分享等等

产品应用

- (1)网站应用开发
- (2)移动应用开发
- (3)第三方平台开发
- (4)公众帐号开发

只有通过 开发者资质认证后，才能使用开发平台提供的能力

个人基本不具备使用开放平台接口的权限，一般是需具有企业资质，比如微信支付，微信登录等

3.微信公众平台

3.1微信公众平台概述

微信公众平台，简称公众号。曾命名为官号平台、媒体平台、微信公众号，最终定位为公众平台。

微信公众平台是运营者通过公众号为微信用户提供资讯和服务的平台。

利用公众账号平台进行自媒体活动，简单来说就是进行一对多的媒体性行为活动，如商家通过申请公众微信服务号通过二次开发展示商家微官网、微会员、微推送、微支付、微活动、微报名、微分享、微名片等，已经形成了一种主流的线上线下微信互动营销方式。

平台网址：<https://mp.weixin.qq.com>

3.2账号分类

微信公众账号分为订阅号、服务号、小程序(小游戏)、企业微信(原企业号)

4.开放平台和公众平台的区别

1. 开放平台
 - (1) 微信对外开放接口的平台
 - (2) 开放的接口,供其他网站及App使用
 - (3) 后端程序员是开放平台开发的主力军
2. 公众平台
 - (1) 基于微信公众号,为微信用户提供服务的平台
 - (2) 所用公众号,都属于微信内开发
 - (3) 前端程序员是公众平台开发的主力军

三、认识小程序

1.小程序概述（了解）

微信小程序,简称小程序,英文名**wechat Mini Program**,是一种不需要下载安装即可使用的应用,它实现了应用“触手可及”的梦想,用户扫一扫或搜一下即可打开应用。

小程序是一种新的开放能力,开发者可以快速地开发一个小程序。小程序可以在微信内被便捷地获取和传播,同时具有出色的使用体验。

全面开放申请后,主体类型为企业、政府、媒体、其他组织或个人的开发者,均可申请注册小程序。小程序、订阅号、服务号、企业号是并行的体系。

2010-2011年智能手机流行开始:

2011年为了解决智能移动设备及时通讯研发了微信,同步产品有米聊,先后微信又发布了微信公众号产品,如订阅号、服务号等,但是这些账号产品在功能无法媲美原生App,所以这就有了微信小程序这种新的账号类型的形态;

2017年1月初微信小程序产品正式上线发布。(微信之父:张小龙)

所以小程序可以理解为是一个简化版app应用

米聊 雷军 更新慢

2.应用场景

微信小程序开发,非常适合一些轻量级、即开即用、用完即走的应用。

例如点外卖、打车、代驾、共享单车、乘车码、行程码等。对于一些大型的重依赖应用,不太适合,比如大型手机游戏类、音乐播放器、银行金融类app等

小程序常见案例:学习城一卡通、北京健康宝、国务院客户端等

3.亮点与不足（重点）

1) 亮点

对于用户而言:

不需要下载,节省了手机内存,使用方便,即开即用、用完即走

打开速度比普通的H5要快,接近原生APP

对于开发者而言:

功能上媲美原生app,比h5强大;
成本上比原生app低(时间 服务器资金),接近h5应用开发成本;
入门门槛较低,掌握基本的html、css、js语法即可快速上手;
生态圈越来越丰富,依托微信、流量巨大

2)不足

小程序目前只有2M大小,使用分包加载可达到20M
ios目前不能直接分享到朋友圈,Android目前支持Beta测试版
不能即时发布,需要微信团队进行审核,审核周期1-7天
对个人开发者开发的服务项目限制非常大。

四、注册小程序账号

1.小程序账号注册流程

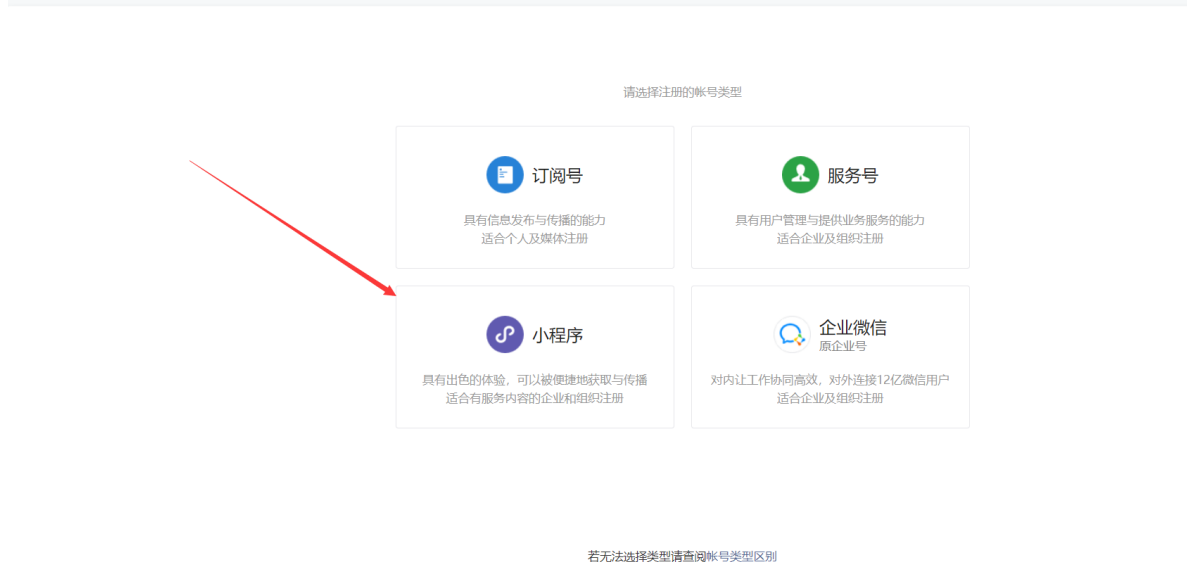
- 1.在微信公众平台官网首页(mp.weixin.qq.com)点击右上角的“立即注册”按钮
- 2.选择注册的帐号类型(小程序)
- 3.填写邮箱和密码(请填写未注册过公众平台、开放平台、企业号、未绑定个人号的邮箱)
- 4.激活邮箱(登录邮箱,查收激活邮件,点击激活链接)
- 5.填写主体信息(选择主体类型(个人)、完善主体信息和管理员信息)

1、



2.

注册



3.

① 帐号信息 — ② 邮箱激活 — ③ 信息登记

每个邮箱仅能申请一个小程序

必须是没有使用过的邮箱地址

163注册新的邮箱

已有微信小程序? [立即登录](#)

创建测试号, 免注册快速体验小程序开发。 [立即申请](#)

邮箱 2209146505@qq.com

作为登录帐号, 请填写未被微信公众平台注册, 未被微信开放平台注册, 未被个人微信号绑定的邮箱

密码

字母、数字或者英文符号, 最短8位, 区分大小写

确认密码

请再次输入密码

验证码



[换一张](#)

☐ 你已阅读并同意《[微信公众平台服务协议](#)》及《[微信小程序平台服务条款](#)》

[注册](#)

4.激活邮箱

注意如果邮件发送成功在收件箱看不到就去垃圾箱找找

5.填写主体类型

填写个人即可

2.小程序信息完善

小程序发布流程

1

小程序信息

补充小程序的基本信息, 如名称、图标、描述等。

[填写](#)

小程序
助手

2

小程序开发与管理

自己开发

开发工具 下载开发者工具进行代码的开发和上传: [普通小程序开发者工具](#)、[小游戏开发者工具](#)。

添加开发者 添加开发者, 进行代码上传。

配置服务器 在 [开发设置](#) 页面查看AppID和AppSecret, 配置服务器域名, 或使用[微信云开发](#)或[微信云托管](#), 免配置服务器。

帮助文档 可以阅读入门介绍 ([普通小程序 | 小游戏](#))、开发文档 ([普通小程序 | 小游戏](#))、[设计规范](#)和[运营规范](#)。

服务平台 提供大量优质的小程序第三方服务, 为不同行业的商家提供合适的小程序解决方案。前往 [服务平台](#)。

[添加开发者](#)

找服务商开发

服务丰富 提供多行业、多功能的小程序开发服务。

省时省心 从小程序认证到发布上线, 全流程支持。

安全可靠 可免费试用后再支付、在线客服响应、官方售后保障。

[立即前往](#)

1. 基本信息

发布上线小程序之前需要填写基本信息，如小程序的头像、名称和简介等，注意项目服务类型，不要填写小游戏

2. 项目成员

管理员：小程序的拥有者，具有后台一切最高权限

项目成员：按照权限可以分为运营者 开发者 数据分析者 有管理员添加管理

体验成员：仅具有使用体验版小程序权限，管理员和项目成员都能添加删除管理

3. 版本管理

开发版本：本地代码开发完毕 有微信开发者工具提交产生，可以被项目成员和体验成员访问

审核版本：有开发版本提交产生，需要微信团队审核，审核周期1-7天

线上版本：有审核版本提交产生，可以被所有的微信用户访问

4. 开发管理

AppID(小程序ID)：项目创建、云开发使用、项目发布等

AppSecret(小程序密钥)：加解密用户的私密数据，如获取用户手机号

五、微信开发者工具及工程创建

首先小程序开发不同于普通的网页开发，项目不能运行在浏览器中，所以无法查看编程效果以及调试，微信开发者工具提供了代码的编程能力，调试能力以及展示运行效果能力等强大的功能

1. 工具下载安装

地址：<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>

建议大家下载稳定的版本即可，下载完毕直接安装即可，预发布版和开发版工具中的功能比较全但是不稳定容易卡死、出现起点启动失败等问题。

Windows 仅支持 Windows 7 及以上版本

下载稳定版

稳定版 Stable Build (1.05.2103200)

测试版缺陷收敛后转为稳定版

Windows 64、Windows 32、macOS

预发布版 RC Build (1.05.2103051)

预发布版，包含大的特性；通过内部测试，稳定性尚可

Windows 64、Windows 32、macOS

开发版 Nightly Build (1.05.2104192)

日常构建版本(基于 NW.js 0.49.3)，用于尽快修复缺陷和敏捷上线小的特性；开发自测验证，稳定性欠佳

Windows 64、Windows 32、macOS

全部更新日志

2. 工程创建

扫码登录工具点击加号创建项目 如下所示：

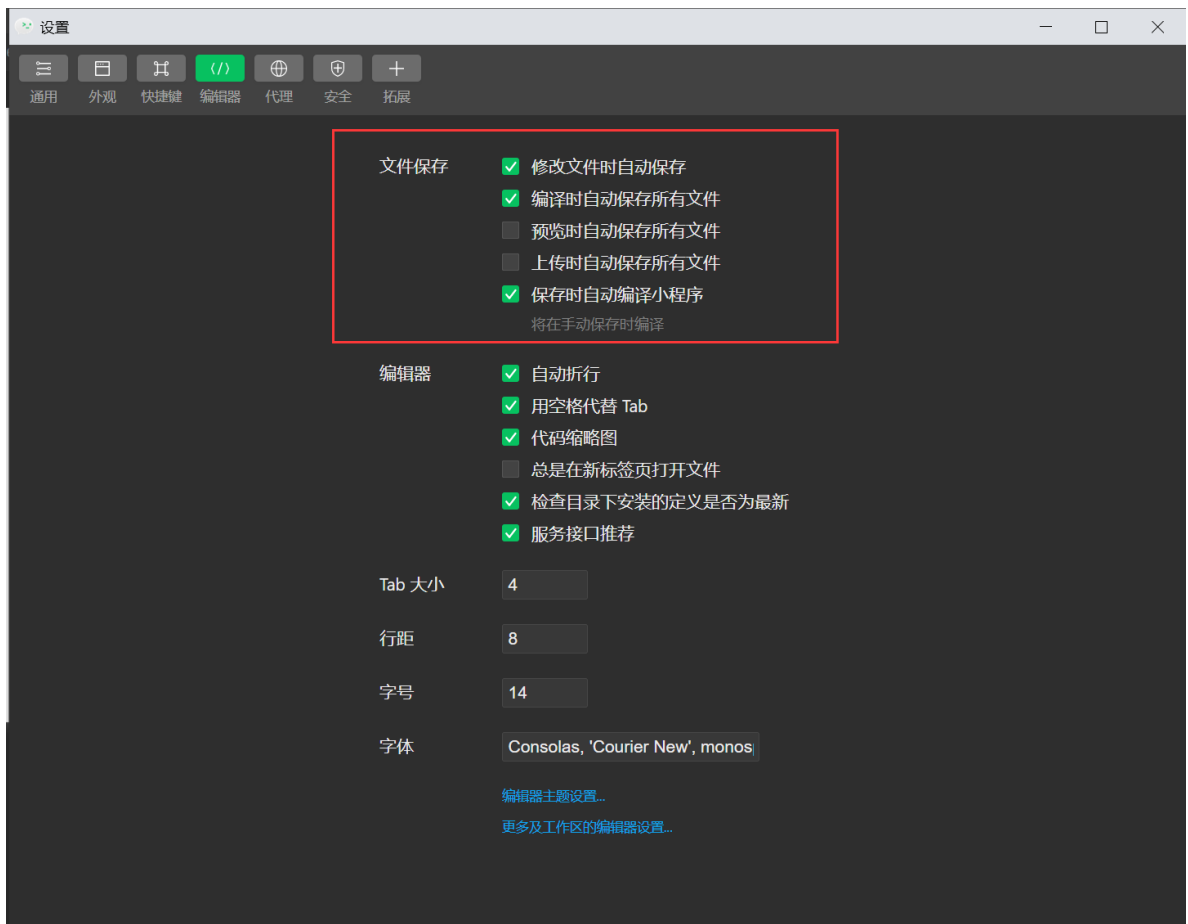


微信开发者工具

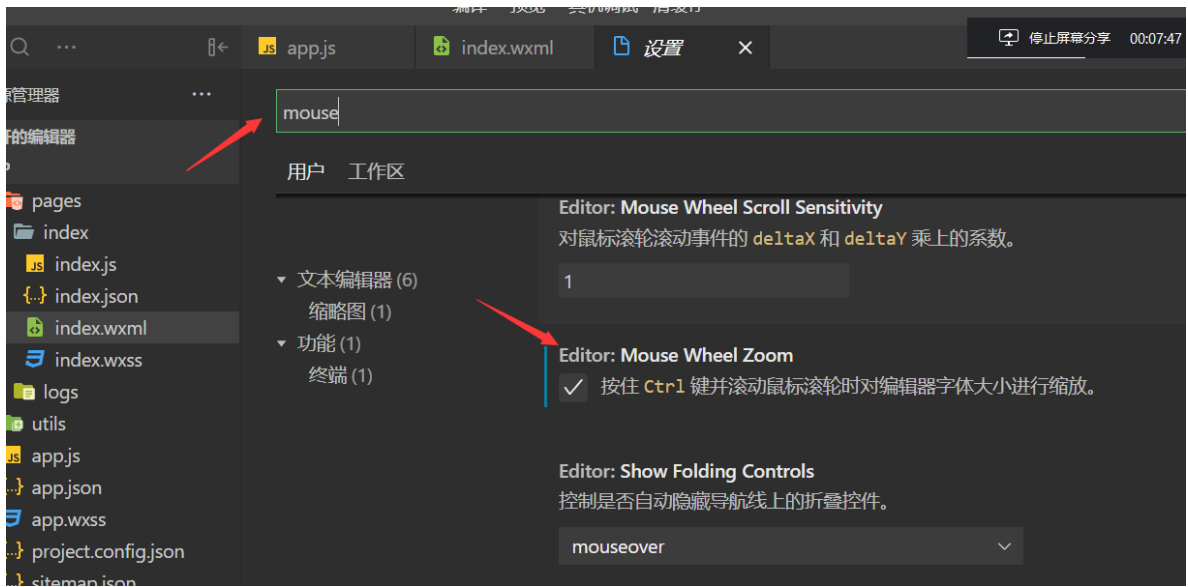
v1.05.2109131

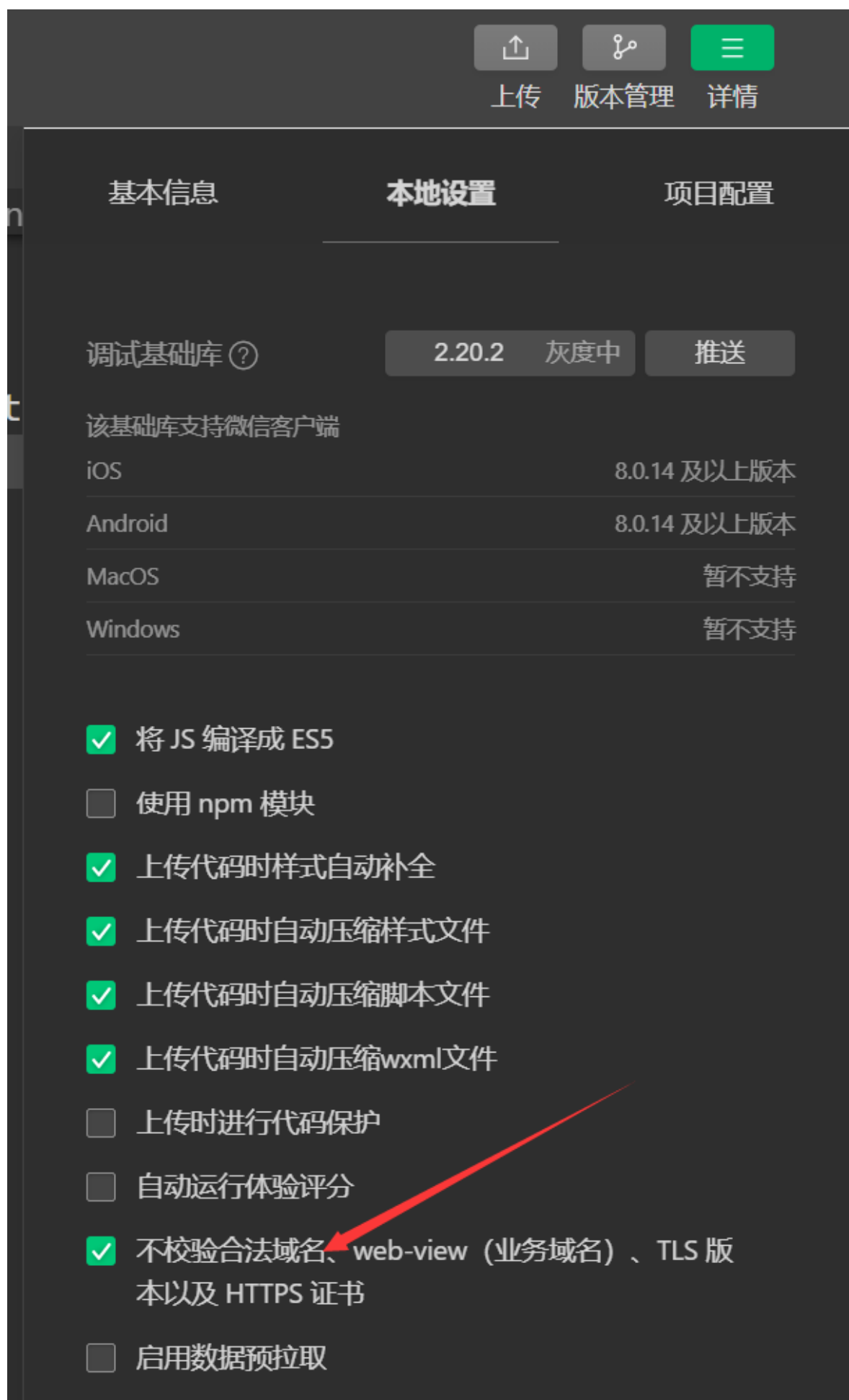


游客模式



设置自动缩放代码





2.小程序目录结构（面试题）

pages:

页面总目录

每个页面都是有 .js .json .wxml .wxss四个文件构成

.js: 页面逻辑 =====>pc端 js

.json: 页面配置文件

.wxml: 页面结构 =====>pc端 html

.wxss: 页面样式 =====>pc端 css

utils:

工具目录 常用函数等

app.js:

微信小程序入口文件 完成小程序的实例注册

app.json:

微信小程序的全局配置文件

app.wxss:

微信小程序的全局样式文件

project.config.json:

记录开发者的偏好设置，比如工具界面ui颜色、编译配置、appid、域名校验等

sitemap.json:

站点地图配置文件，用来配置是否允许微信爬虫爬取我们小程序的页面，可以优化提高我们小程序的曝光率

eslint.js:

eslint校验配置文件

七、小程序配置

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

1.配置概述

小程序提供了json文件用来对微信小程序进行配置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等,主要分为全局配置及页面配置

2.全局配置app.json

小程序根目录下的 app.json 文件用来对微信小程序进行全局配置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等。文件内容为一个 JSON 对象

1.entryPagePath:

功能:

小程序默认启动首页

语法:

```
"entryPagePath": "pages/logs/logs",
```

注意事项:

1. 如果不填, 将默认为 `pages` 列表的第一项。

2. pages:

功能:

1. 描述项目页面路由信息
2. 快速创建页面
3. 未指定 `entryPagePath` 时, 数组的第一项代表小程序的初始页面 (首页)。

使用语法:

```
"pages": [  
  "pages/order/order",  
  "pages/index/index",  
  "pages/logs/logs",  
  "pages/user/user",  
  "pages/cart/index"  
]
```

注意事项:

1. `app.json: ["pages"][0]` 不应该以 `'/'` 开头
2. `app.json: ["pages"][0]` 不应该以 `'.'` 开头

Some messages have been moved to the Issues panel.

✖ [app.json 文件内容错误] app.json: ["pages"][0] 不应该以 `'/'` 开头
(env: Windows,mp,1.05.2110290; lib: 2.21.1)

Some messages have been moved to the Issues panel.

✖ [app.json 文件内容错误] app.json: ["pages"][0] 不应该以 `'.'` 开头
(env: Windows,mp,1.05.2110290; lib: 2.21.1)

3.window:

窗口构成如下:



功能：

配置心目中的导航条 状态栏 窗口背景色 屏幕旋转等

使用语法：

```
"window": {
  "navigationBarBackgroundColor": "#f00",
  "navigationBarTitleText": "我是微信小程序",
  "navigationBarTextStyle": "white",
  "navigationStyle": "default",
  "backgroundColor": "#DDD",
  "backgroundTextStyle": "dark",
  "enablePullDownRefresh": true,
  "pageOrientation": "auto"
},
```

使用注意：

1. 如果自定义导航栏，右侧胶囊会保留去不掉
2. 刷新结束后需要手动关闭动画效果（`api`函数）

4.debug

功能：

方便测试调试，可以打印项目从启动到页面加载完毕的所有信息，比如：**Page** 的注册，页面路由，数据更新，事件触发

使用语法：

```
"debug": true,
```

5.tabbar

功能：

实现项目中置顶或者是在底部的选项卡效果

使用语法：

```
"tabBar": {
  "color": "#000000",
  "selectedColor": "#f00",
  "backgroundColor": "#DDD",
  "borderStyle": "white",
  "position": "top",
  "list": [
    {
      "pagePath": "pages/index/index",
      "text": "首页",
      "iconPath": "./images/icon/index.png",
      "selectedIconPath": "/images/icon/indexFull.png"
    },
    {
```

```
        "pagePath": "pages/user/user",
        "text": "我的",
        "iconPath": "images/icon/my.png",
        "selectedIconPath": "images/icon/myFull.png"
      }
    ]
  },
}
```

使用注意:

1. 只要页面底部绑定选项卡的才会显示选项卡效果
2. 选项卡数量最少2个最多5个, 符合人体工程学

3. 页面配置 .json

地址: <https://developers.weixin.qq.com/miniprogram/dev/framework/config.html#%E9%A1%B5%E9%9D%A2%E9%85%8D%E7%BD%AE>

每一个小程序页面也可以使用同名 `.json` 文件来对本页面的窗口表现进行配置, 页面中配置项会覆盖 `app.json` 的 `window` 中相同的配置项。

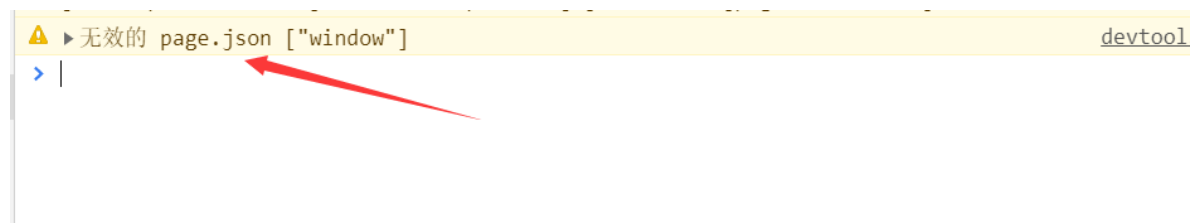
使用语法:

```
{
  "usingComponents": {},

  "navigationBarBackgroundColor": "#ffffff",
  "navigationBarTextStyle": "black",
  "navigationBarTitleText": "个人中心",
  "backgroundColor": "#eeeeee",
  "backgroundTextStyle": "light",
  "enablePullDownRefresh": false,
  "disableScroll": true
}
```

使用注意:

1. 页面配置中只能设置 `app.json` 中 `window` 对应的配置项, 以决定本页面的窗口表现, 所以无需写 `window` 这个属性。
2. `"disableScroll": true` 只能在页面配置项中设置全局无效



八、场景值（了解）

地址: <https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/scene.html>

1.场景值概述

场景值用来描述用户进入小程序的方式，比如小程序最近使用列表、扫描二维码、小程序主入口等

2.获取场景值

使用语法:

1. App 的 onLaunch 和 onShow
2. 通过wx.getLaunchOptionsSync()api函数获取

app.js文件:

```
// 使用api函数获取场景值(同步的api)
let info = wx.getLaunchOptionsSync();
console.log(3,info)

App({

  // 项目启动执行
  onLaunch(e) {

    console.log(1,e.scene);

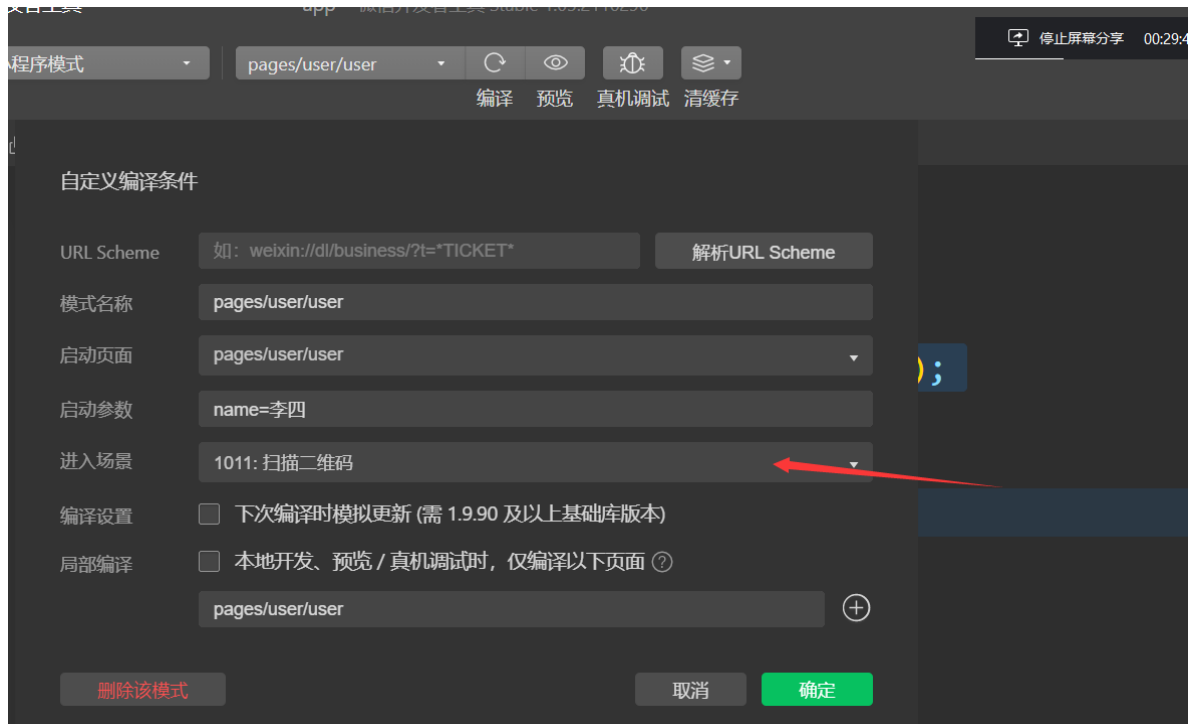
    // 展示本地存储能力
    const logs = wx.getStorageSync('logs') || []
    logs.unshift(Date.now())
    wx.setStorageSync('logs', logs)

    // 登录
    wx.login({
      success: res => {
        // 发送 res.code 到后台换取 openId, sessionKey, unionId
      }
    })
  },

  // 项目启动执行
  onShow(e){
    console.log(2,e.scene);
  },
  globalData: {
    userInfo: null
  }
})
```

获取更多场景值:

通过点击工具栏----->编译模式--->新增编译模式----->添加场景值



九、小程序逻辑层（重点）

整个小程序框架系统分为两部分：**逻辑层**（App Service）和**视图层**（View）。VC结构

注册小程序：app()

注册页面：page()

生命周期：执行顺序 应用级别生命周期 页面级别生命周期

1.逻辑层概述

小程序开发框架的逻辑层使用 **JavaScript** 引擎为小程序提供开发者 **JavaScript** 代码的运行环境以及微信小程序的特有功能。

逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈。

开发者写的所有代码最终将会打包成一份 **JavaScript** 文件，并在小程序启动的时候运行，直到小程序销毁。这一行为类似 **ServiceWorker**，所以逻辑层也称之为 **App Service**。

Tips：

小程序框架的逻辑层并非运行在浏览器中，因此 **JavaScript** 在 **web** 中一些能力都无法使用，如 **window**，**document** 等。

2.注册小程序

每个小程序都需要在 `app.js` 中调用 `App` 方法注册小程序实例，
app方法内容： 绑定生命周期回调函数、错误监听和页面不存在监听函数、自定义全局函数和自定义全局变量等。

`App()` 必须在 `app.js` 中调用，必须调用且只能调用一次。不然会出现无法预期的后果

app方法语法：

```
App({

  // 冷启动：如果用户首次打开，或小程序销毁后被用户再次打开，此时小程序需要重新加载启动，即冷启动。
  // 热启动：如果用户已经打开过某小程序，然后在一定时间内再次打开该小程序，此时小程序并未被销毁，
  // 只是从后台状态进入前台状态，这个过程就是热启动。

  /**
   * 当小程序初始化完成时，会触发 onLaunch（全局只触发一次，冷启动）
   */
  onLaunch: function () {
    console.log('触发了onlaunch')
    // 应用场景：
    // 1. 获取用户的场景值
    // 2. 检测用户登录
    // 3. 基础库兼容性判断（day5优化）
    // 4. 记录用户的访问日志
    // 5. 云开发环境初始化（day4）

    // 调用不存在函数
    // console.log(this); // app实例
    // this.fun();

    // 调用自定义函数
    this.user();
    this.users();
    this.userss();
  },

  /**
   * 当小程序启动，或从后台进入前台显示，会触发 onShow（冷启动 热启动）
   */
  onShow: function (options) {
    console.log('触发了onShow')
  },

  /**
   * 当小程序从前台进入后台，会触发 onHide
   */
  onHide: function () {
    console.log('触发了onHide')
  },

  /**
   * 当小程序发生脚本错误，或者 api 调用失败时，会触发 onError 并带上错误信息
   */
  onError: function (msg) {
    console.log('监听到错误了', msg)
  },
});
```

```

/**
 *小程序要打开的页面不存在时触发。
 *
 */
onPageNotFound(){

  console.log('访问的页面不存在');
  // 帮助用户定向到存在的页面，路由跳转
  wx.reLaunch({
    url: 'pages/index/index',
  })
},
// =====自定义函数=====
// 1.函数声明式
user:function(){
  console.log('我是声明式函数')
  console.log(this)
},
// 2.ES6简写（提倡）
users(){
  console.log('我是简写函数')
  console.log(this)
},
// 3.箭头函数
userss:()=>{
  console.log('我是箭头函数')
  console.log(this)
},
// =====自定义全局变量=====
globalData:{
  name:'李四',
  array:[1,2,3,4],
  buffer:true
}
})

```

获取app实例：

```

// 获取app实例
let app = getApp();
console.log(app.globalData.name)

```

3.注册页面

在每一个独立页面的js文件中，使用Page(Object object)注册页面，接受一个 Object 类型参数
page方法内容：页面的初始数据data、生命周期回调、事件处理函数、自定义事件等。

page方法语法：

```

// 获取app实例
let app = getApp();

```

```
// console.log(app.globalData.name)

Page({

  /**
   * 页面的初始数据
   */
  data: {
    age:20,
    hopy:'篮球'
  },

  //=====自定义事件=====
  // 1.函数声明式
  user:function(){
    console.log('我是声明式函数')
    console.log(this)
  },
  // 2.ES6简写（提倡）
  users(){
    console.log('我是简写函数')
    console.log(this)
  },
  // 3.箭头函数
  userss:()=>{
    console.log('我是箭头函数')
    console.log(this)
    // 定时器 回调函数
  },

  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function (options) {
    console.log('页面-onLoad')
    // 应用场景：1.获取页面跳转传递的参数 A(name=李四)-B（onload）
    // 2.获取网络数据（发起网络请求 使用接口）

    // 使用自定义函数
    this.user();
    this.users();
    this.userss();
  },

  /**
   * 生命周期函数--监听页面显示
   */
  onShow: function () {
    console.log('页面-onShow')
    // 应用场景：1.发起网络请求
  },

  /**
   * 生命周期函数--监听页面初次渲染完成
   */
  onReady: function () {
    console.log('页面-onReady')
    // 应用场景：1.获取页面上的元素位置
  }
})
```

```

},

/**
 * 生命周期函数--监听页面隐藏
 */
onHide: function () {
  console.log('页面-onHide')
},

/**
 * 生命周期函数--监听页面卸载
 */
onUnload: function () {
  console.log('页面-onUnload')
}
})

```

4.生命周期:

微信小程序的生命周期同vue一样，也是不需要立马搞明白，后续根据业务需要渐进式使用即可；
 微信小程序的生命周期可以分为两种：一种是app.js文件中的应用级别的生命周期，另一种就是页面.js文件中的页面生命周期；
 当我们进入小程序应用的时候首先执行app.js文件中的生命周期函数再次执行要访问的页面中的生命周期函数，切后台的时候先执行页面的生命周期函数再次执行app.js文件中的应用级别的生命周期函数

十、模块化

微信小程序中完全支持模块化规范，ES6和commonjs规范；

微信小程序完全支持commonjs和ES6模块化规范

1.commonjs

导出：module.exports exports 是module.exports的引用
 引入：require('文件路径')

2.ES6

导出：export {} export default
 引入：import {} '文件路径' import from '文件路径'

utils/tip.js文件:

```

// 定义消息提示方法
const tip=(title,icon)=>{
  // 弹窗的api函数
  wx.showToast({
    title,
    icon

```

```
    })  
  }  
  
  // 导出方法  
  module.exports={  
    tip  
  }  
}
```

在user.js文件中使用：

```
// 引入消息提示模块文件  
let {tip} = require('../utils/tip');  
// 使用方法  
tip('温馨提示','error');
```