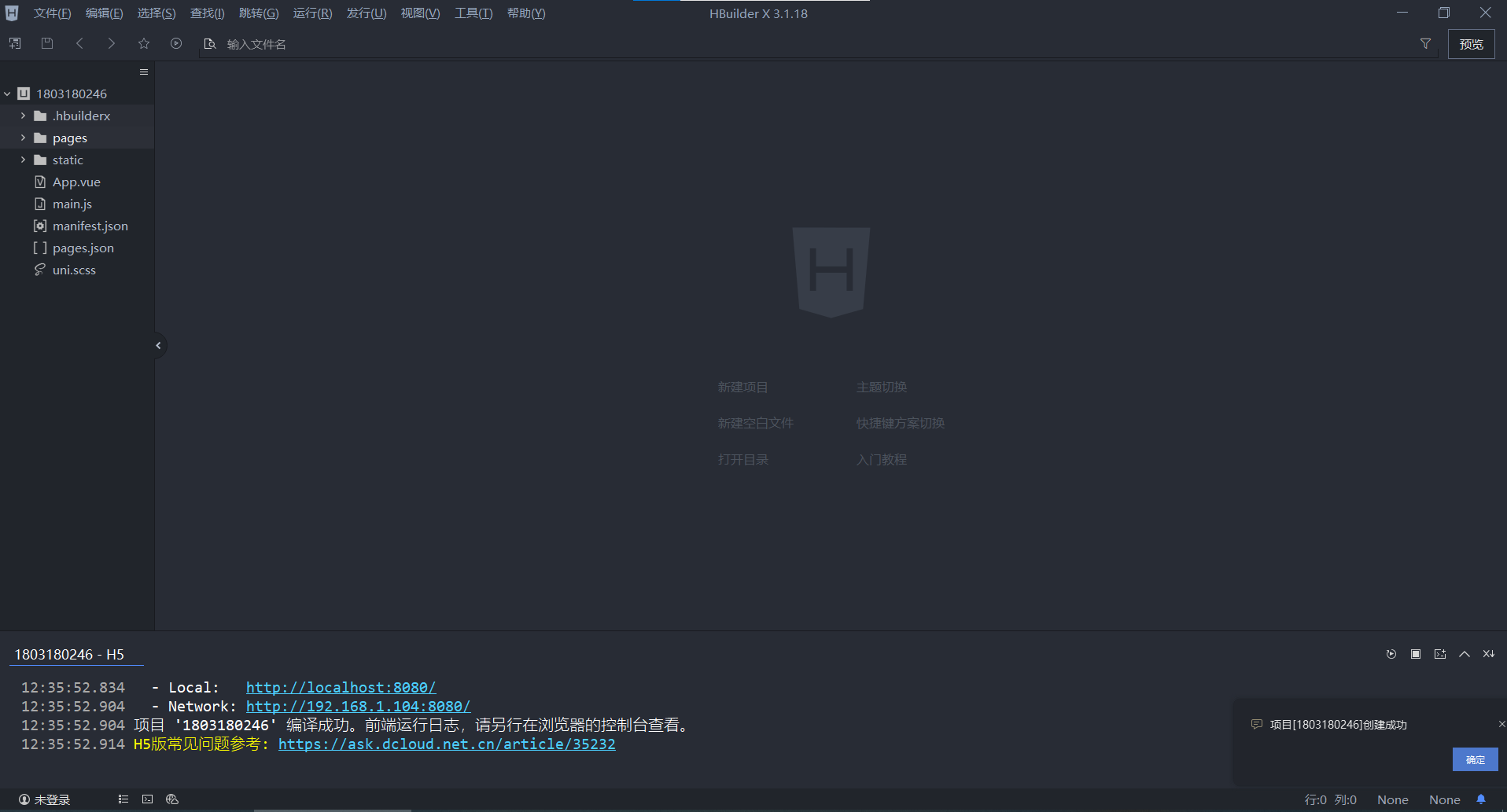
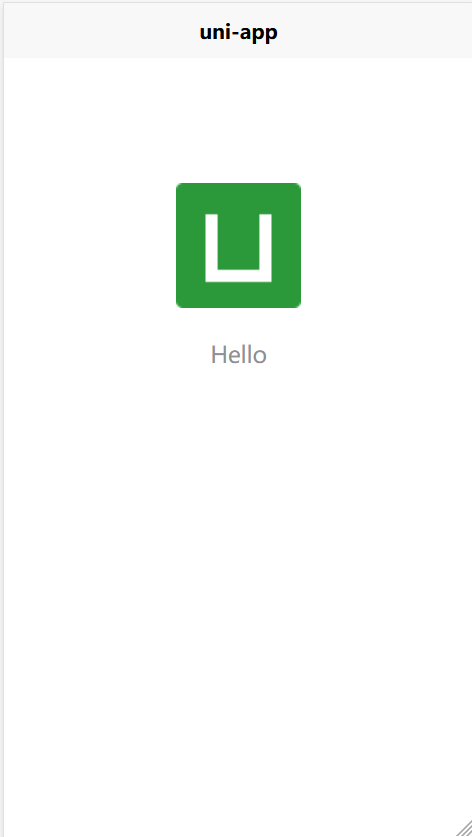
## Uniapp环境配置 ：（下载软件，安装必要插件）



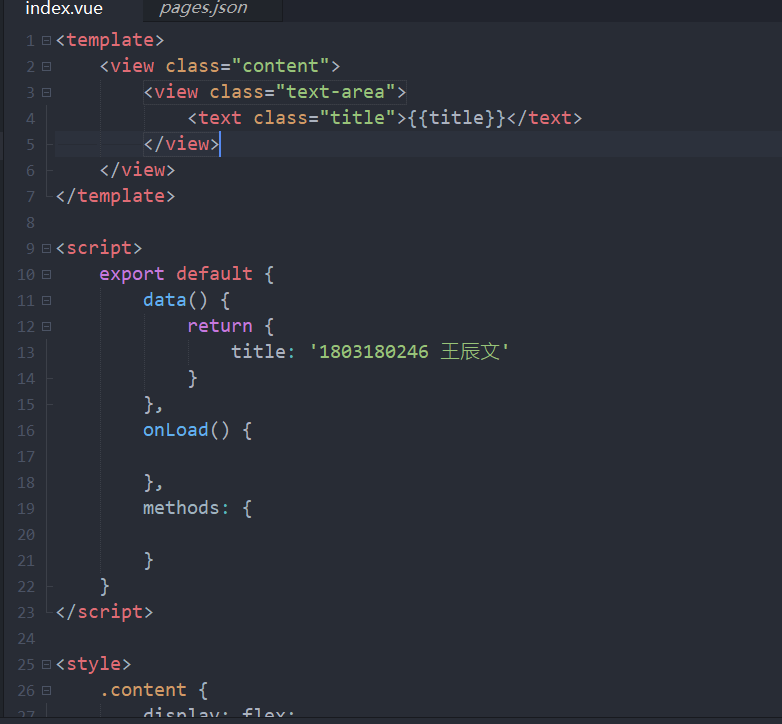
## 创建Uniapp项目运行





## 创建一个组件，并进行使用,组件显示学生的姓名和学号

创建基础view组件和text组件：



运行效果：



## 创建底部导航（系统）

源码：



运行效果：

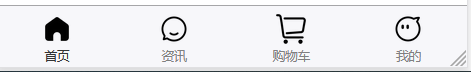


## 创建底部导航（自定义）

源码：

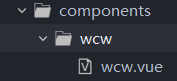


效果：

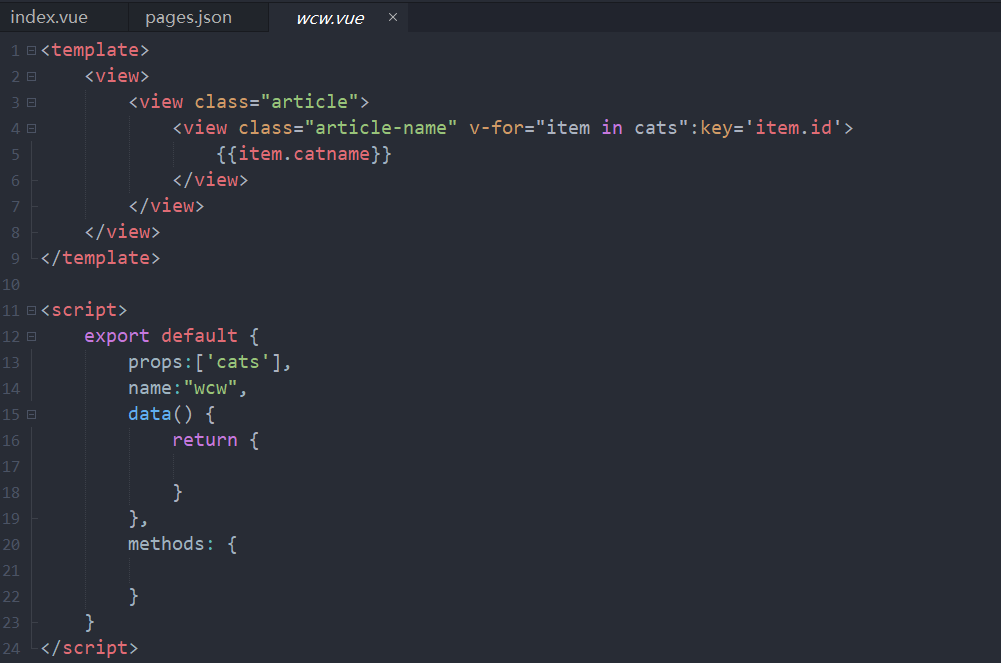


## 配置Vuex状态管理环境

创建Vuex组件，名为：components



创建wcw.vue：

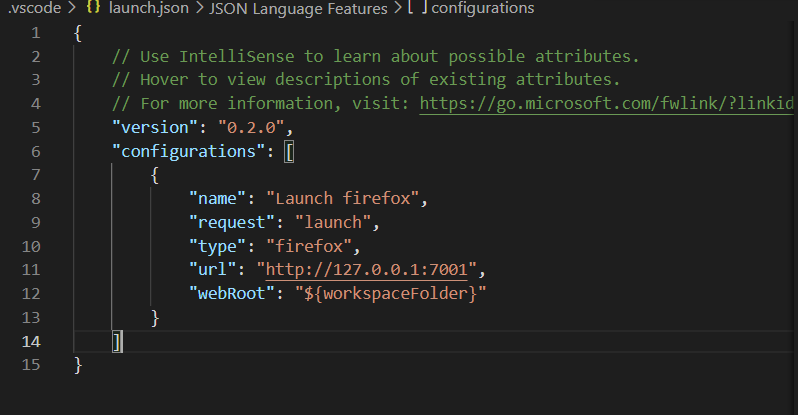


Index配置获取地址数据：

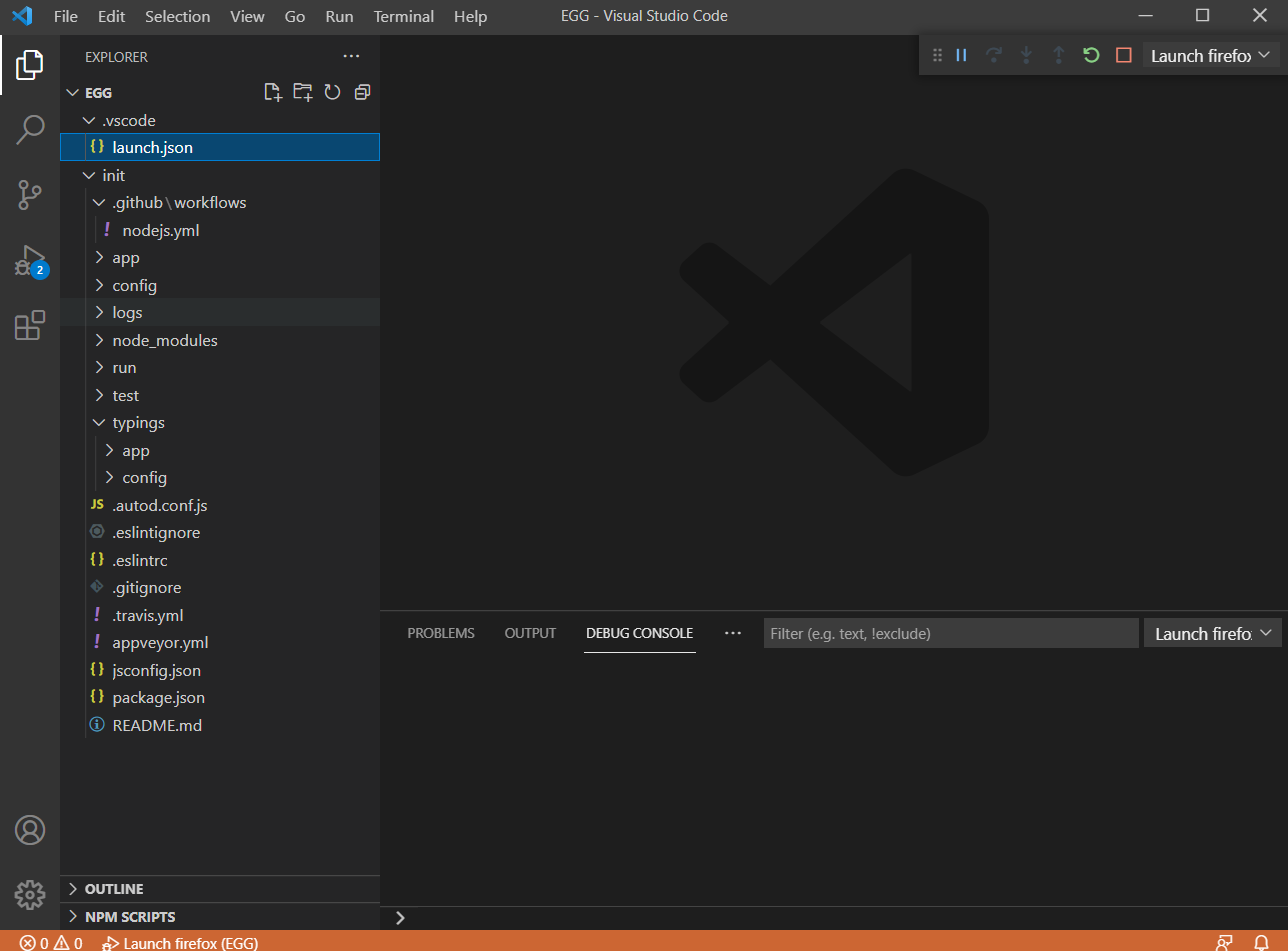


## 创建eggjs服务端并运行。

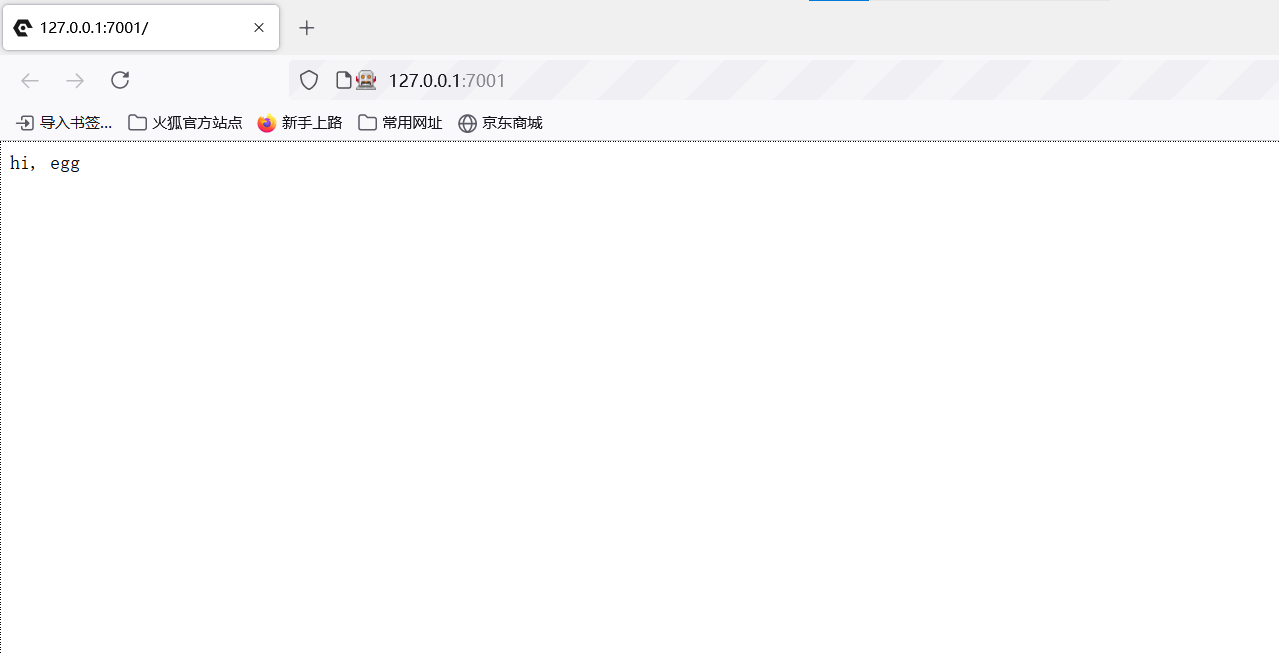
配置服务端运行环境：



运行：

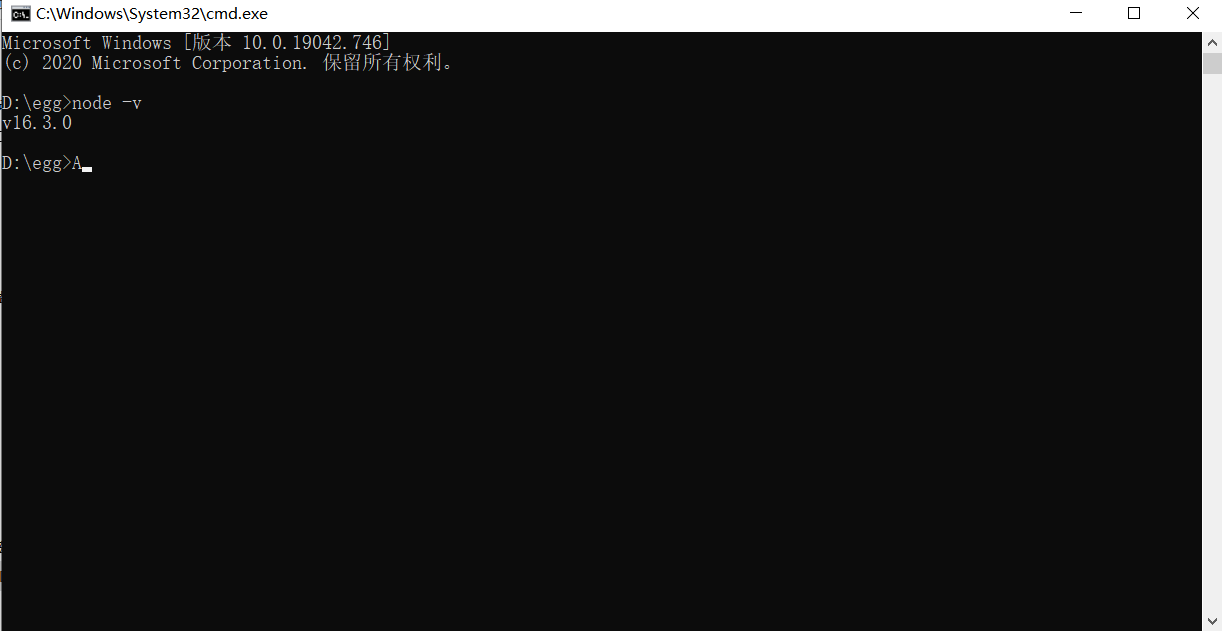


效果图：

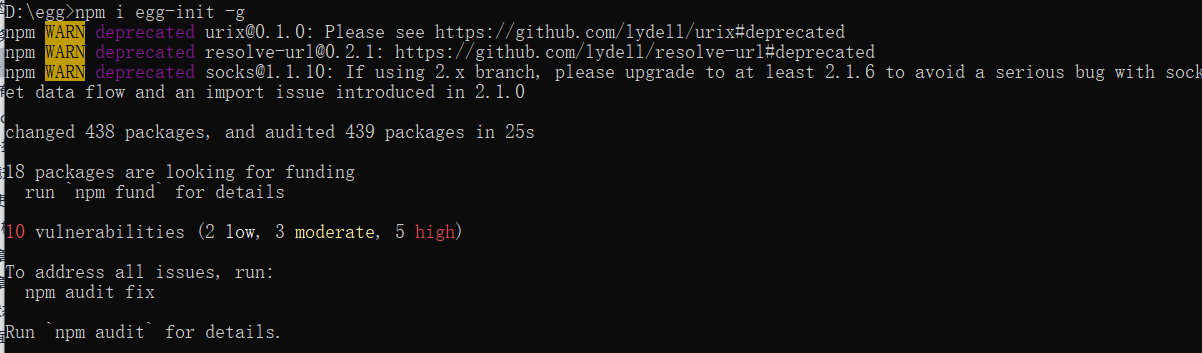


## 配置nodejs ，npm，cnpm 环境

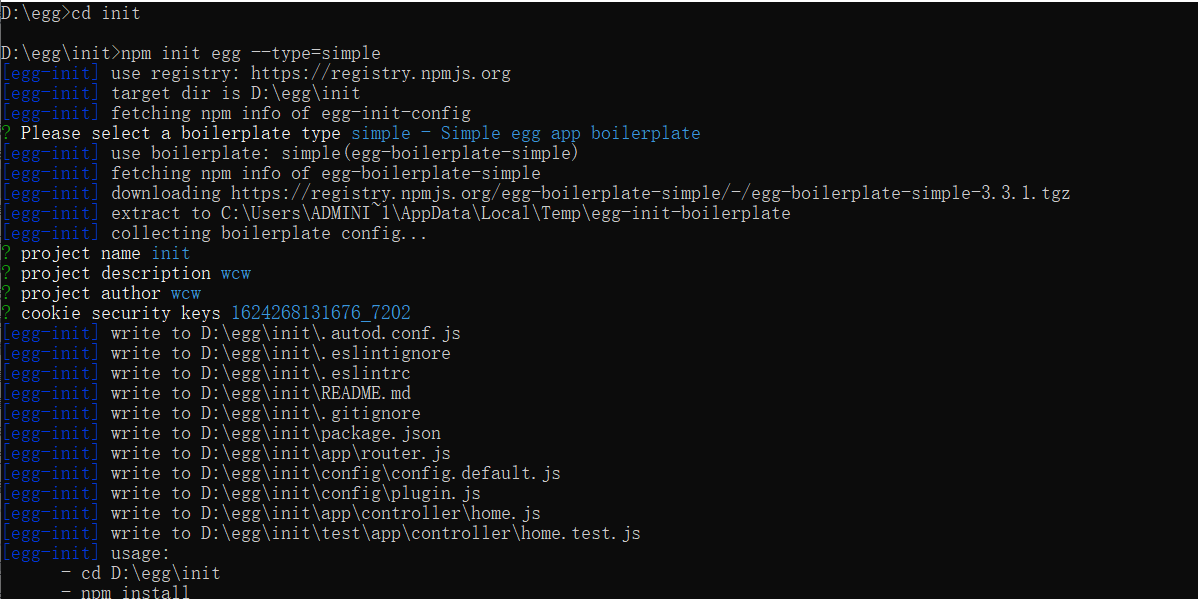
配置node.js：



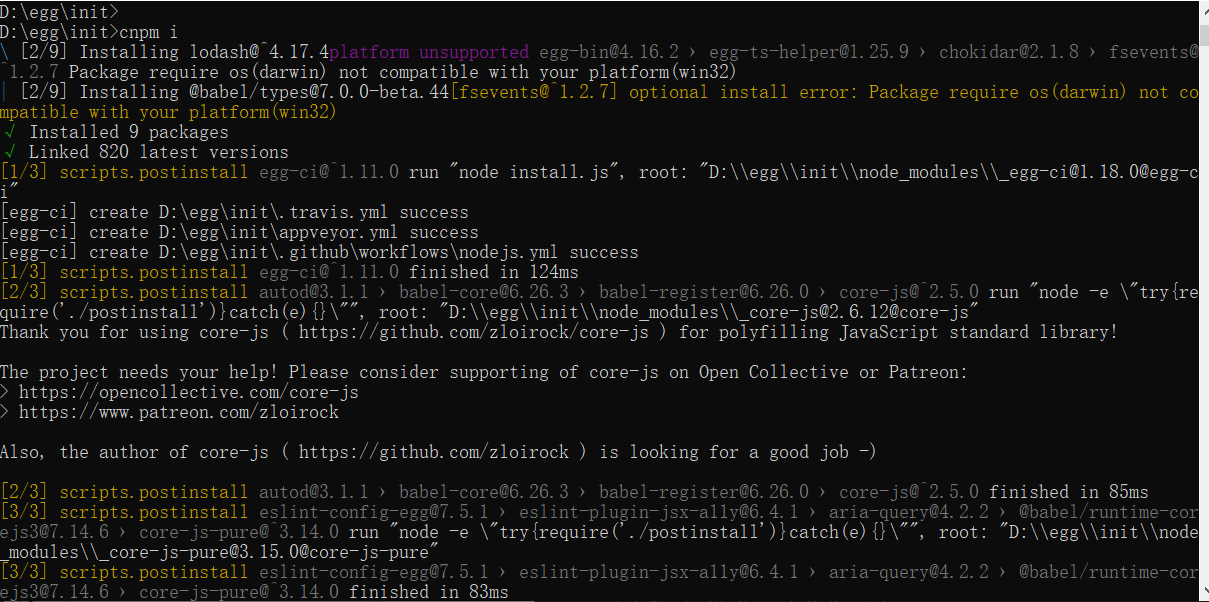
配置npm环境：



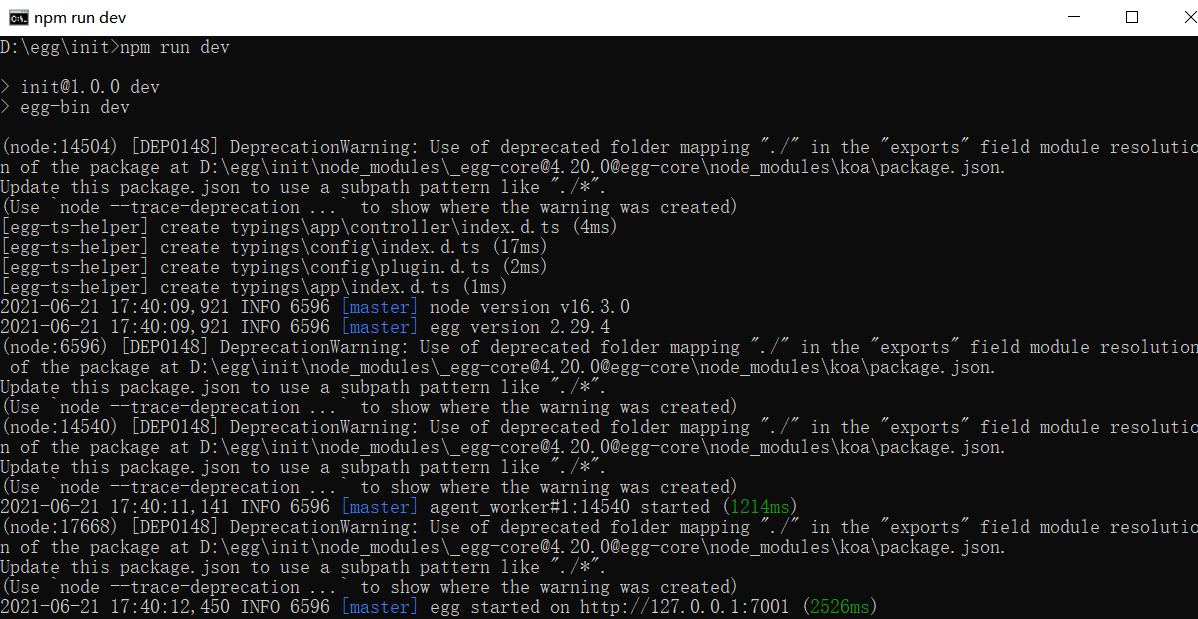
进入创建的项目目录中：



配置cnpm环境：



运行：



## Vue实例参数有哪些？

$data：对实例地data对象属性地访问；  
$props：对组件实例地props对象属性地访问；  
$el：访问实例使用的根 DOM 元素；  
$options：访问实例所有属性；  
$isServer：当前 Vue 实例是否运行于服务器；  
$attrs：包含了父作用域中不作为 prop 被识别 (且获取) 的 attribute 绑定 (class 和 style 除外)。当一个组件没有声明任何 prop 时，这里会包含所有父作用域的绑定 (class 和 style 除外)，并且可以通过 v-bind="$attrs" 传入内部组件——在创建高级别的组件时非常有用；  
$listeners：包含了父作用域中的 (不含 .native 修饰器的) v-on 事件监听器。它可以通过 v-on="$listeners" 传入内部组件——在创建更高层次的组件时非常有用。

## 组件的 easycom规范是什么

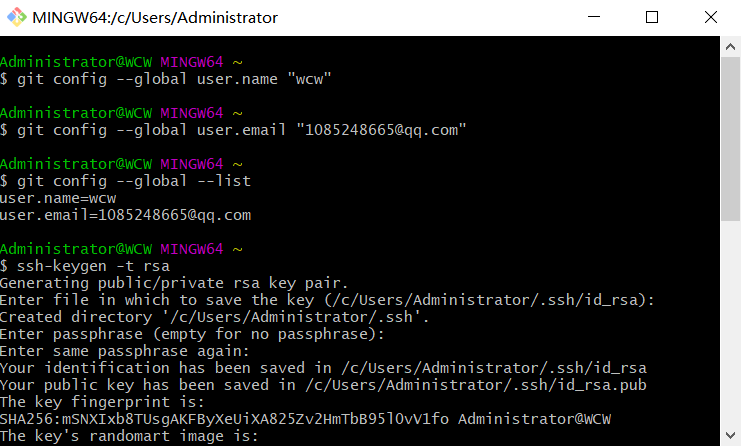
只要组件安装在项目的components目录下或uni\_modules目录下，并符合components/组件名称/组件名称.vue目录结构。就可以不用引用、注册，直接在页面中使用。  
不管components目录下安装了多少组件，easycom打包后会自动剔除没有使用的组件  
easycom是自动开启的，不需要手动开启。  
如果你的组件名称或路径不符合easycom的默认规范，可以在pages.json的easycom节点进行个性化设置，自定义匹配组件的策略

## Uniapp部署打包

从上到下设置：

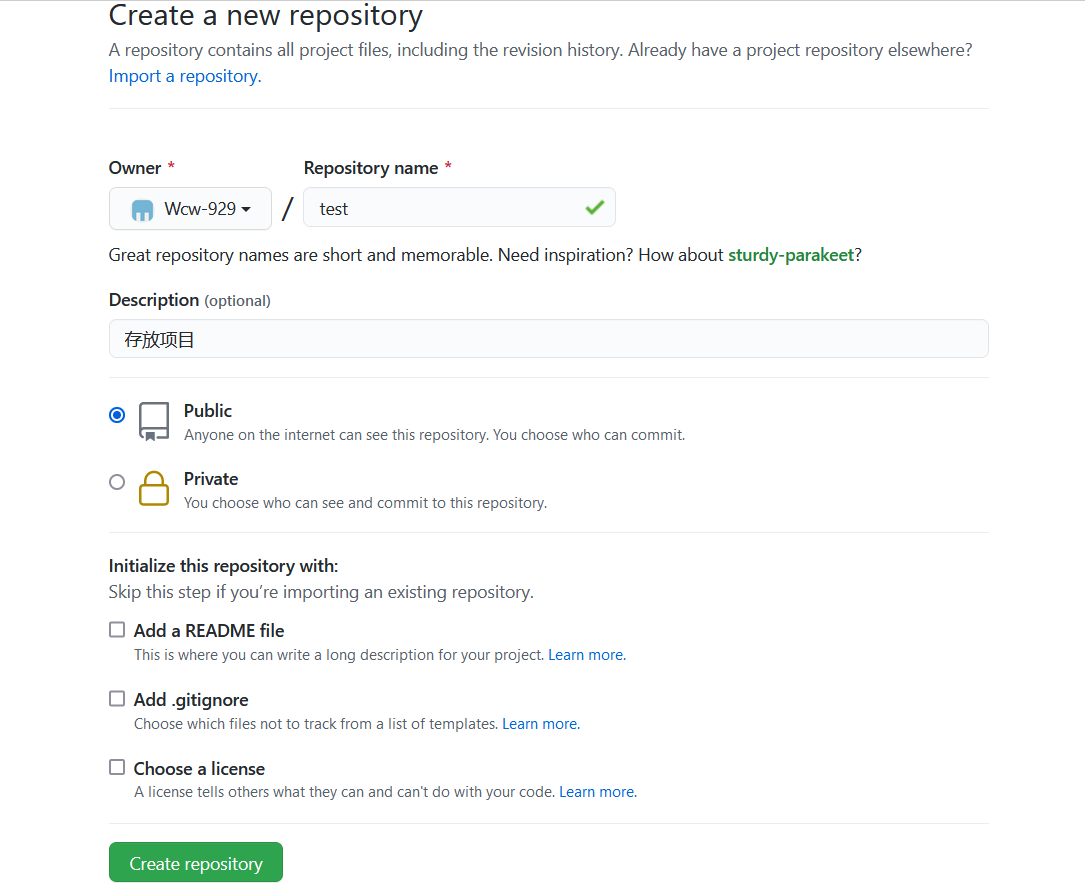


## 配置git环境并使用

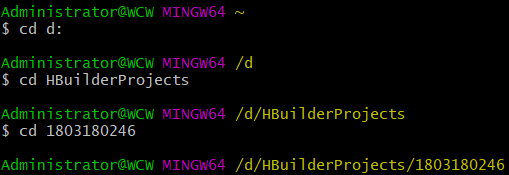


## 将项目源码使用git上传的github网站上。

创建仓库：



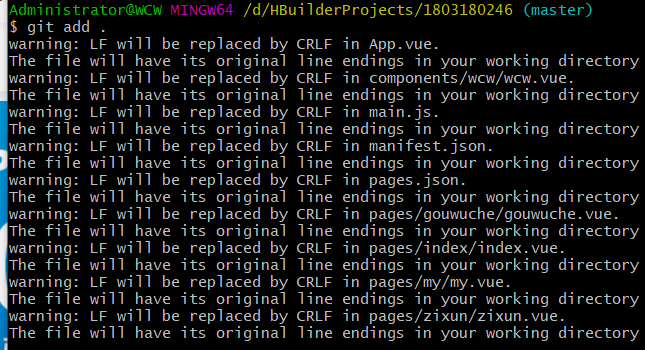
进入项目文件：



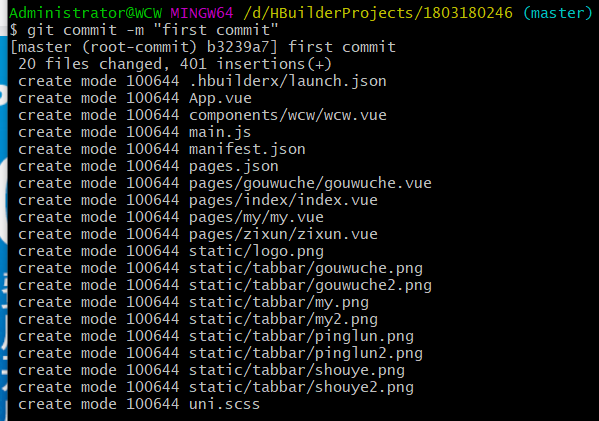
在当前项目的目录中生成本地的git管理：



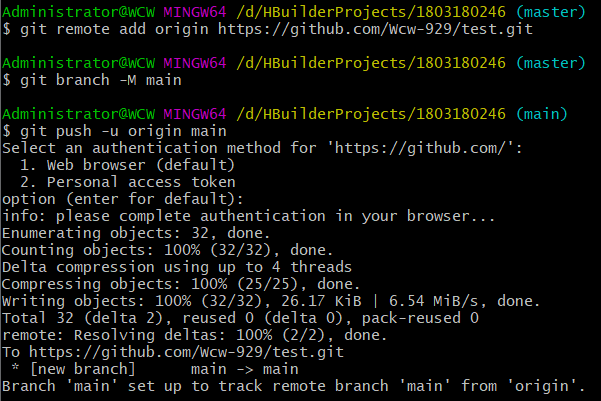
将项目上所有的文件添加到仓库中：



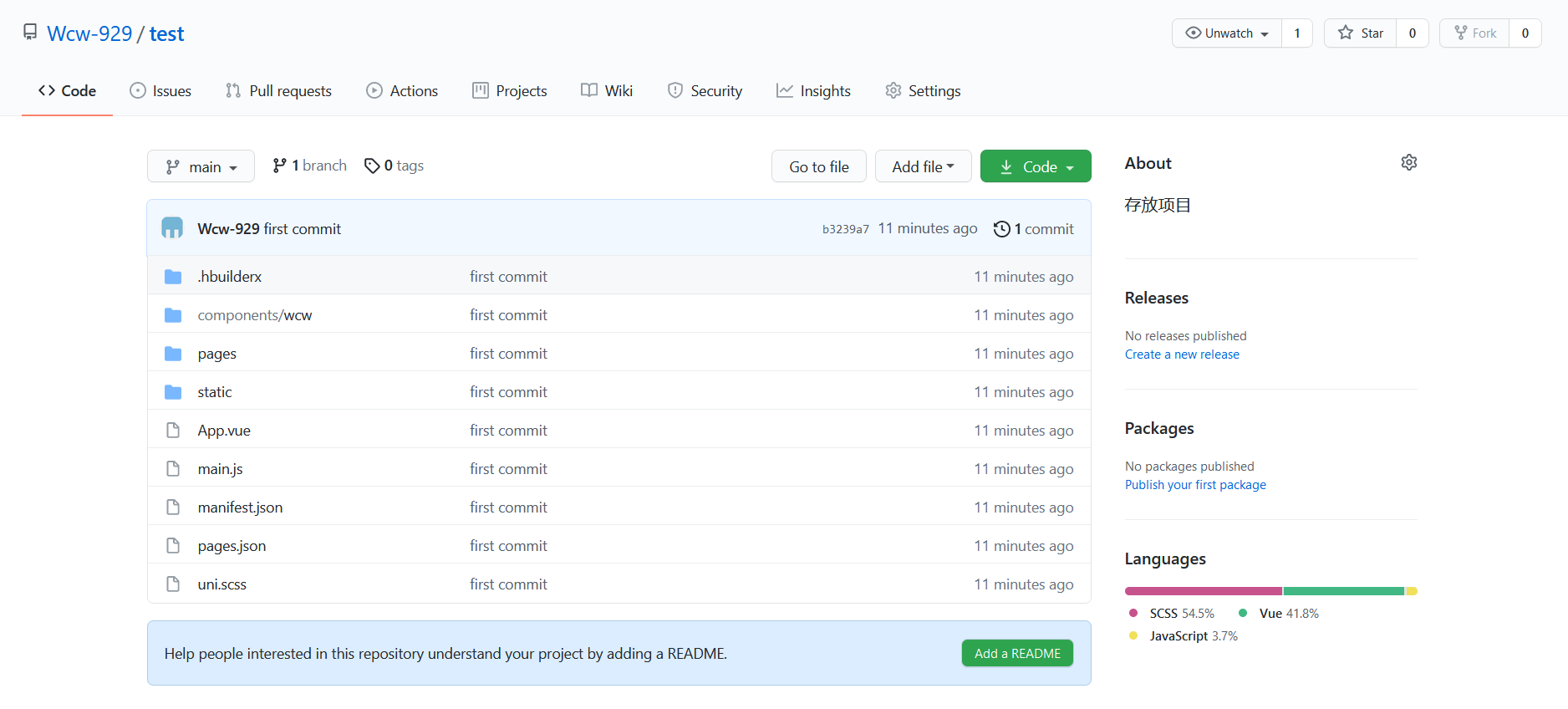
提交的注释，双引号里面的内容可以根据个人的需要修改：



关联并上传仓库：

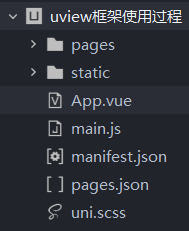


验证：



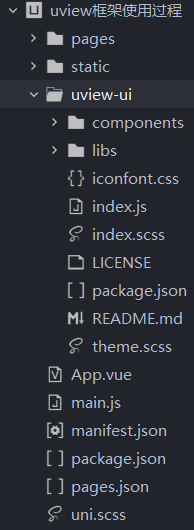
## uview框架的使用过程

创建uni-app项目：



下载uview框架并导入：





插件安装：



在项目根目录中的main.js中，引入并使用uView的JS库， 注意这两行要放在import Vue之后：



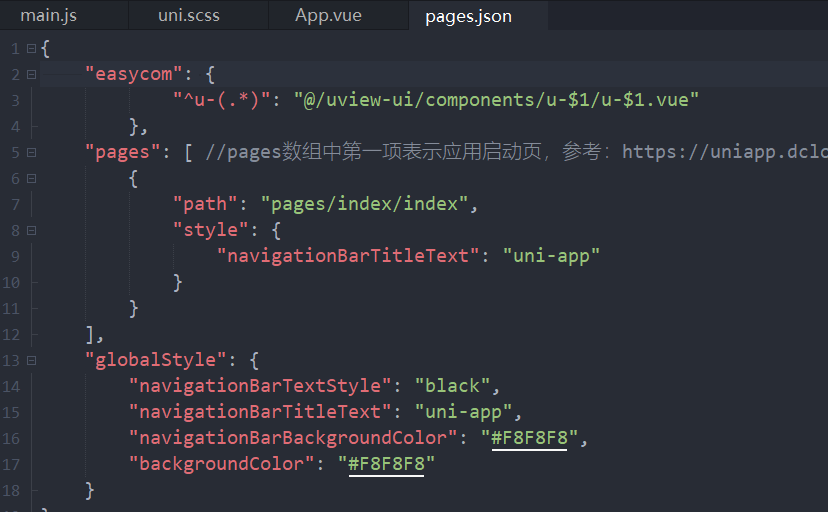
在项目根目录的uni.scss中引入此文件：



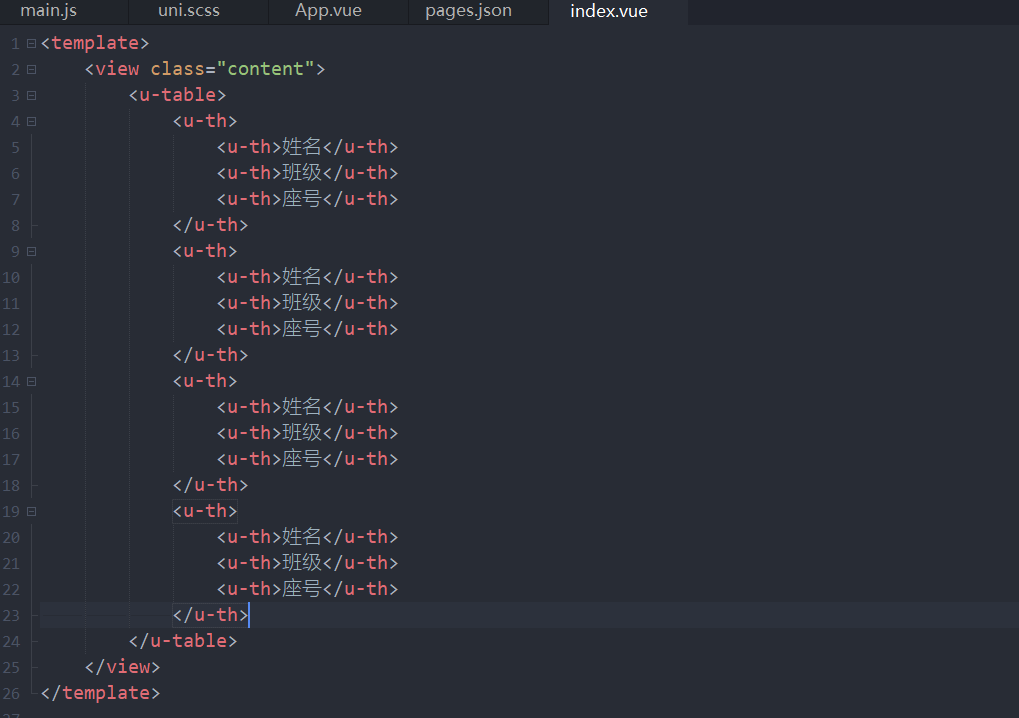
在App.vue中首行的位置引入，注意给style标签加入lang="scss"属性：



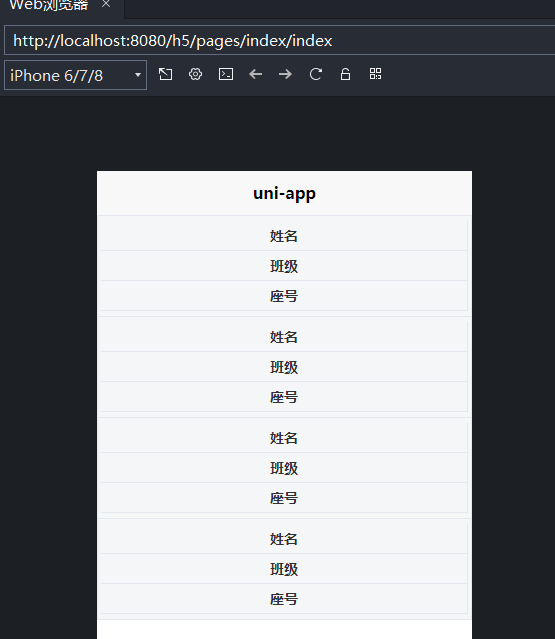
配置easycom组件模式：



到此处，我们已经将uview配置好了，接下来写个示例验证一下：



效果图：



## 根据文章数据，创建组件文章列表，点击查看文章详细页面

学习通作业

## 根据文章分类数据，创建组件文章分类，能够对文章分类进行管理

学习通作业

## Vuex状态管理的对象

·state：存储状态。也就是变量；  
·getters：派生状态。也就是set、get中的get，有两个可选参数：state、getters分别可以获取state中的变量和其他的getters。外部调用方式：store.getters.personInfo()。就和vue的computed差不多；  
·mutations：提交状态修改。也就是set、get中的set，这是vuex中唯一修改state的方式，但不支持异步操作。第一个参数默认是state。外部调用方式：store.commit('SET\_AGE', 18)。和vue中的methods类似。  
·actions：和mutations类似。不过actions支持异步操作。第一个参数默认是和store具有相同参数属性的对象。外部调用方式：store.dispatch('nameAsyn')。  
·modules：store的子模块，内容就相当于是store的一个实例。调用方式和前面介绍的相似，只是要加上当前子模块名，如：store.a.getters.xxx()。

## 实现页面跳转的组件和api有哪些？有哪些作用？

1.1：在组件内跳转至非tabbar页面并返回。只需在组件内加一个uniapp提供的跳转标签即可 <navigator url = " 地址 " > </navigator>  
1.2：在组件内跳转至tabbar页面并不返回。<navigator url = " 地址 " open-type = " switchTab" > </navigator>  
2.1：如在点击事件内跳转至非tabbar页面并返回。只需在点击事件内加一个uniapp提供的的跳转方法即可 uni.navigateTO ({   url : "地址"   })  
 2.2：如在点击事件内跳转至tabbar页面并不返回。 uni.switchTab({   url : "地址"   })

## Uni-app存储数据的方式？

[uni.setStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=setstorage)

将数据存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个异步接口。

[uni.setStorageSync(KEY,DATA)](https://uniapp.dcloud.io/api/storage/storage?id=setstoragesync)

将 data 存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个同步接口。

[uni.getStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=getstorage)

从本地缓存中异步获取指定 key 对应的内容。

[uni.getStorageSync(KEY)](https://uniapp.dcloud.io/api/storage/storage?id=getstoragesync)

从本地缓存中同步获取指定 key 对应的内容。

[uni.getStorageInfo(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=getstorageinfo)

异步获取当前 storage 的相关信息。

[uni.getStorageInfoSync()](https://uniapp.dcloud.io/api/storage/storage?id=getstorageinfosync)

同步获取当前 storage 的相关信息。

[uni.removeStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=removestorage)

从本地缓存中异步移除指定 key。

[uni.removeStorageSync(KEY)](https://uniapp.dcloud.io/api/storage/storage?id=removestoragesync)

从本地缓存中同步移除指定 key。

[uni.clearStorage()](https://uniapp.dcloud.io/api/storage/storage?id=clearstorage)

清理本地数据缓存。

[uni.clearStorageSync()](https://uniapp.dcloud.io/api/storage/storage?id=clearstoragesync)

同步清理本地数据缓存。

## Uni-app常见组件和api（不少于10个）及其作用

常见组件：

View 视图容器

它类似于传统html中的div，用于包裹各种元素内容

[scroll-view](https://uniapp.dcloud.io/component/scroll-view?id=scroll-view) 可滚动视图区域

用于区域滚动。

[swiper](https://uniapp.dcloud.io/component/swiper?id=swiper) 滑块视图容器

一般用于左右滑动或上下滑动，比如banner轮播图。

[icon](https://uniapp.dcloud.io/component/icon?id=icon)

图标

[text](https://uniapp.dcloud.io/component/text?id=text)

文本组件。

用于包裹文本内容

[button](https://uniapp.dcloud.io/component/button?id=button)

按钮

[switch](https://uniapp.dcloud.io/component/switch?id=switch)

开关选择器

[textarea](https://uniapp.dcloud.io/component/textarea?id=textarea)

多行输入框

[navigator](https://uniapp.dcloud.io/component/navigator?id=navigator)

页面跳转

[image](https://uniapp.dcloud.io/component/image?id=image)

图片

[video](https://uniapp.dcloud.io/component/video?id=video)

视频播放组件

[navigation-bar](https://uniapp.dcloud.io/component/navigation-bar?id=navigation-bar)

页面导航条配置节点，用于指定导航栏的一些属性。只能是 [page-meta](https://uniapp.dcloud.io/component/page-meta" \t "https://uniapp.dcloud.io/component/_blank) 组件内的第一个节点，需要配合它一同使用

[custom-tab-bar](https://uniapp.dcloud.io/component/custom-tab-bar?id=custom-tab-bar)

自定义tabBar组件

API：

[uni.request(OBJECT)](https://uniapp.dcloud.io/api/request/request?id=request)

发起网络请求

[uni.navigateTo(OBJECT)](https://uniapp.dcloud.io/api/router?id=navigateto)

保留当前页面，跳转到应用内的某个页面，使用uni.navigateBack可以返回到原页面

[uni.setNavigationBarTitle(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=setnavigationbartitle)

动态设置当前页面的标题

[uni.setNavigationBarColor(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=setnavigationbarcolor)

设置页面导航条颜色

[uni.setBackgroundColor(OBJECT)](https://uniapp.dcloud.io/api/ui/bgcolor?id=setbackgroundcolor)

动态设置窗口的背景色

[uni.setBackgroundTextStyle(OBJECT)](https://uniapp.dcloud.io/api/ui/bgcolor?id=setbackgroundtextstyle)

动态设置下拉背景字体、loading 图的样式

[uni.pageScrollTo(OBJECT)](https://uniapp.dcloud.io/api/ui/scroll?id=pagescrollto)

将页面滚动到目标位置

[onPullDownRefresh](https://uniapp.dcloud.io/api/ui/pulldown?id=onpulldownrefresh)

在 js 中定义 onPullDownRefresh 处理函数（和onLoad等生命周期函数同级），监听该页面用户下拉刷新事件

[uni.switchTab(OBJECT)](https://uniapp.dcloud.io/api/router?id=switchtab)

跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面

## 使用uni-app实现用户模拟登录注册（和服务端结合）

## vue常见指令有哪些，作用是什么？

v-model 多用于表单元素实现双向数据绑定  
v-for 循环数组  
v-show 显示内容 ( display:block )  
v-hide 隐藏内容 ( display:none )  
v-if 显示与隐藏  
v-else-if ( 必须和 v-if 连用 v-else 必须和 v-if 连用 不能单独使用 否则报错 模板编译错误 )  
v-bind 属性绑定 ( 可以缩写为 : )  
v-bind:class三种绑定方法 ( 1、对象型 ‘{red:isred}’ 2、三元型 ‘isred?“red”:“blue”’ 3、数组型 ‘[{red:“isred”},{blue:“isblue”}]’ )  
v-on:click 给标签绑定函数 ( 可以缩写为@ )  
v-text 解析文本  
v-html 解析 html 标签  
v-once 进入页面时 只渲染一次 不在进行渲染  
v-cloak 防止闪烁  
v-pre 把标签内部的元素原位输出

## uniapp生命周期有哪些？

应用生命周期  
页面生命周期  
组件生命周期

## props数据类型，如何实现props的数据验证

props:{

'参数名':{

 type:数据类型名称, //default:"默认展示的内容"

default:function(){

 return { msg:'默认展示的内容'

}

 }

 }

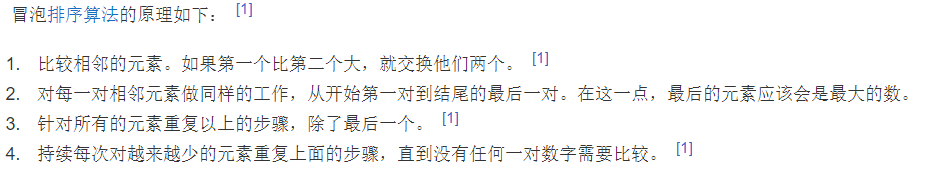
 }

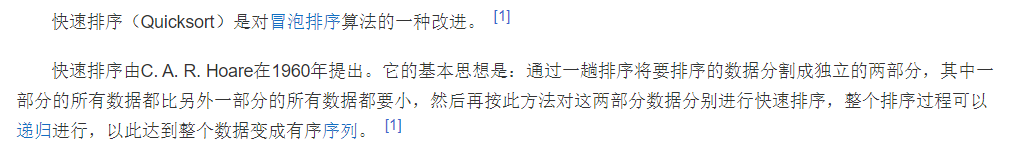
## Promise是什么？如何实现

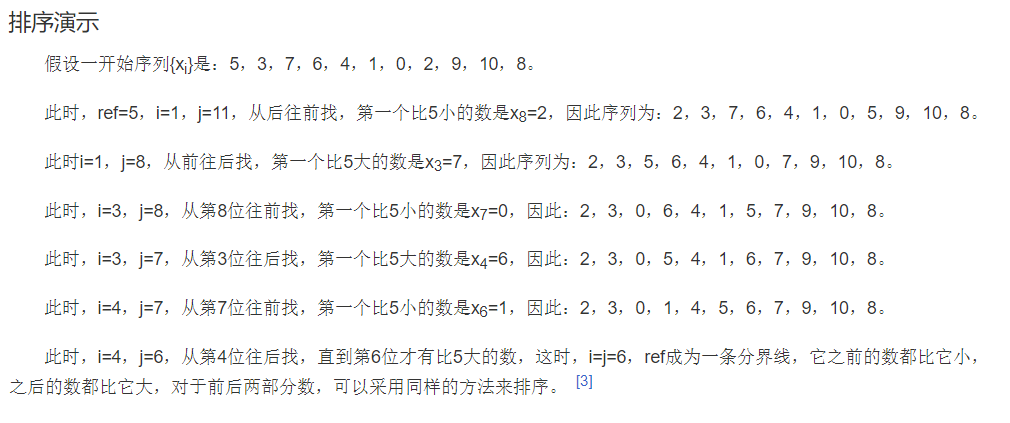
Promise 对象用于表示一个异步操作的最终完成 (或失败)及其结果值。

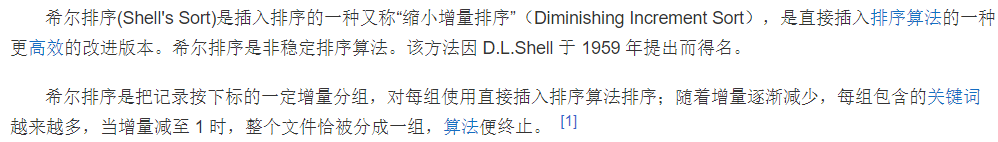
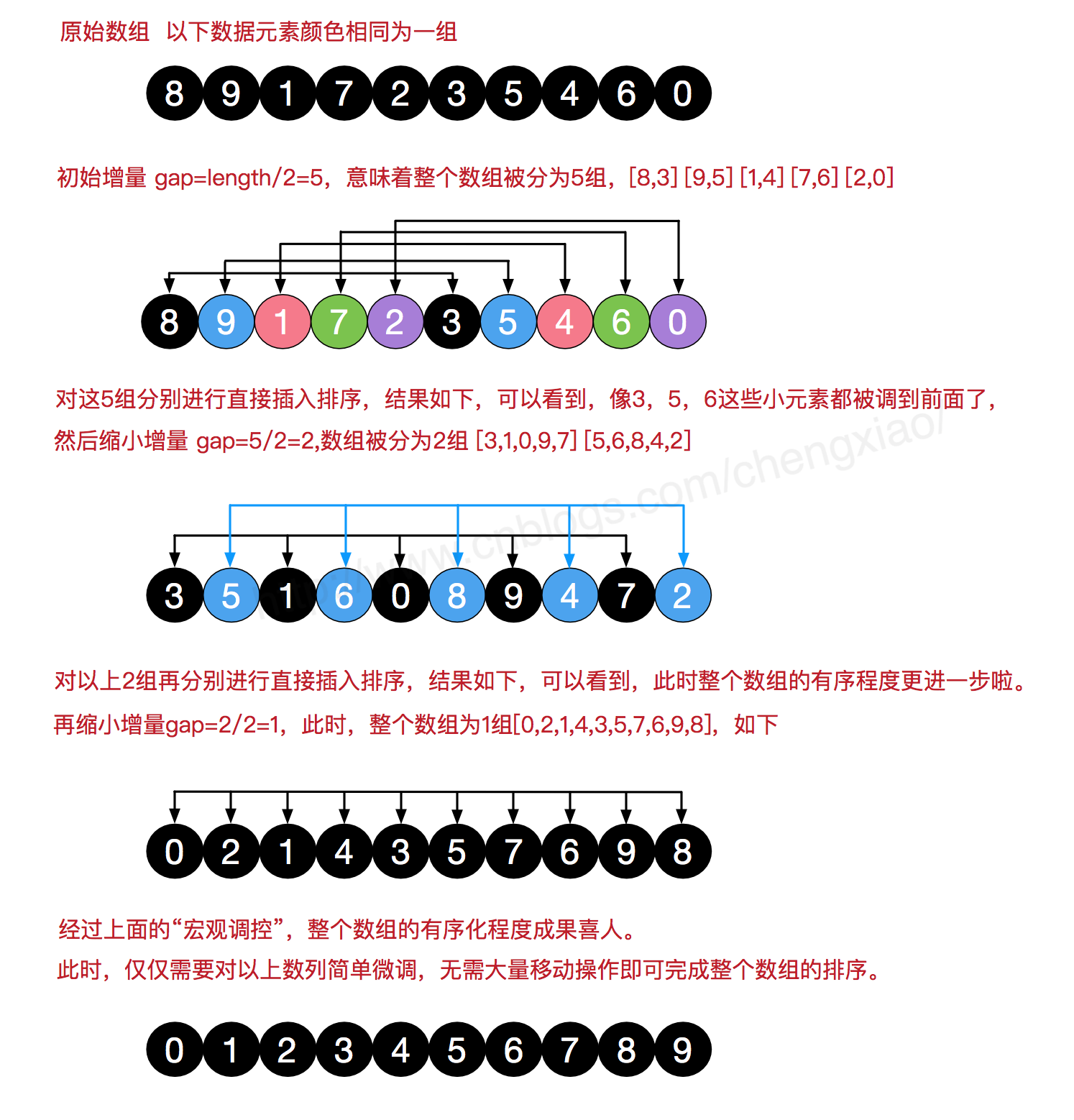
## 在nodejs的环境下，使用promise读取文件的数据

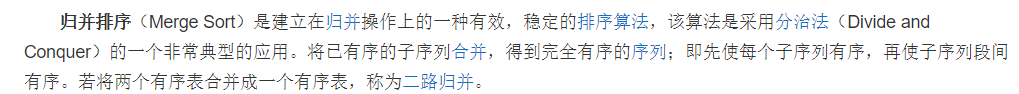
## javascript算法实现：冒泡算法，快速排序，希尔排序，归并排序，堆排序，计数排序，桶排序，基数排序。

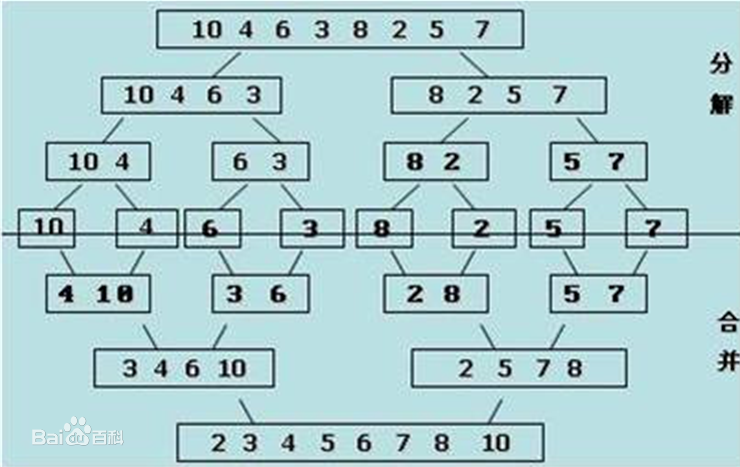


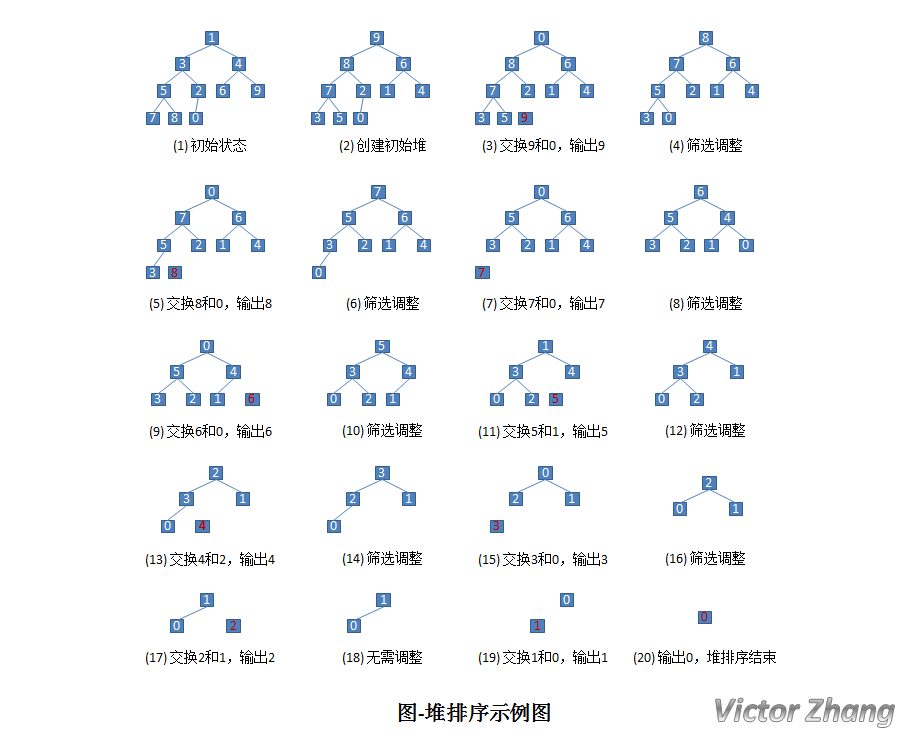


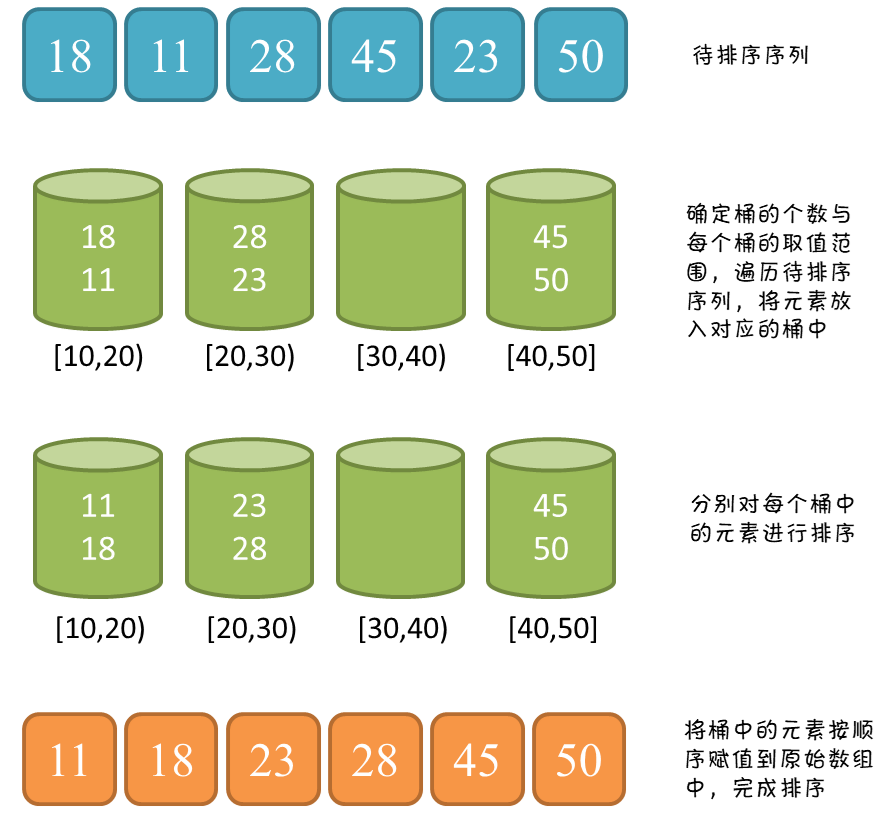


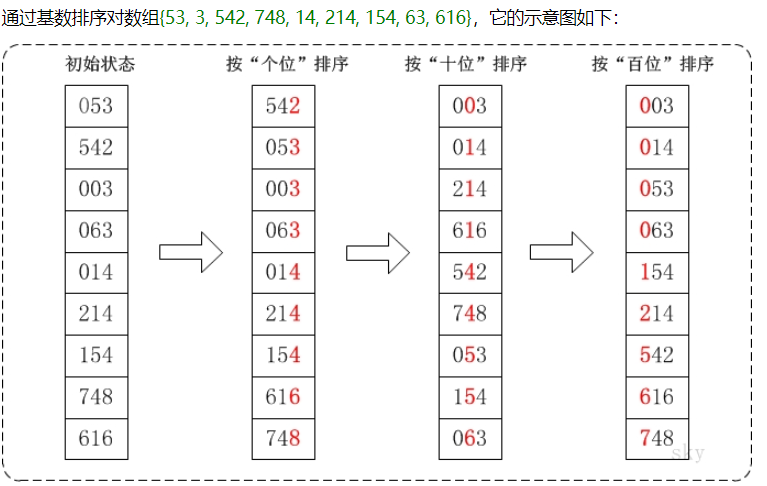
 











## pages.json 文件用来对 uni-app 进行全局配置，配置项有哪些，分别作用是什么？



## 组件的创建的过程？

## 组件通信的过程？