1)要求
1.1：每一天老师书写代码务必三遍
1.2:node + webpack + VScode + 谷歌浏览器 + git
1.3:数组的方法 + promise + await + async + 模块化.....

2)脚手架使用
2: vue init webpack 项目的名字
3|4：vue create 项目名称
脚手架目录:public + assets文件夹区别
node_modules:放置项目依赖的地方
public:一般放置一些共用的静态资源，打包上线的时候，public文件夹里面资源原封不动打包到dist文件夹里面
src：程序员源代码文件夹
-----assets文件夹：经常放置一些静态资源（图片），assets文件夹里面资源webpack会进行打包为一个模块（js文件夹里面）
-----components文件夹:一般放置非路由组件（或者项目共用的组件）（全局组件）
App.vue 唯一的根组件
main.js 入口文件【程序最先执行的文件】
babel.config.js:babel配置文件（如es6转es5）
package.json：看到项目描述、项目依赖、项目运行指令，
README.md:项目说明文件

3)脚手架下载下来的项目稍微配置一下
3.1:浏览器自动打开
在package.json文件中
"scripts": {
"serve": "vue-cli-service serve --open",
"build": "vue-cli-service build",
"lint": "vue-cli-service lint"
},

3.2关闭eslint校验工具
根目录下创建vue.config.js文件：需要对外暴露（如声明变量但是没有使用eslint校验工具就报错）
```
module.exports = {
  lintOnSave:false,
}
```
3.3 src文件夹的别名的设置（@代表src文件下）
因为项目大的时候src（源代码文件夹）：里面目录会很多，找文件不方便，设置src文件夹的别名的好处，找文件会方便一些
根目录下创建jsconfig.json文件
```
{
  "compilerOptions": {
    "baseUrl": "./",
    "paths": {
      "@/*": [
        "src/*"
      ]
    }
  },
  "exclude": [
    "node_modules",
    "dist"
```

]
}

4:项目上传GIT
微信小程序实战课的时候，会带着大家玩耍的
注意:前面基础课程当中，创建分支、处理冲突等等
https://gitee.com/jch1011/shangpinhui_0607.git

5）路由的配置
vue-router
路由分为KV

node平台（并非语言）
对于后台而言:K即为URL地址 V即为相应的中间件
http://localhost:8080/0607
```
app.get("/0607",(res,req)=>{
  res.send('我是祖国的老花骨朵');
});
```

前端路由:
K即为URL（网络资源定位符）
V即为相应的路由组件

5.1路由的一个分析
确定项目结构顺序:上中下 -----只有中间部分的V在发生变化，中间部分应该使用的是路由组件
2个非路由组件|四个路由组件
两个非路由组件：Header 、Footer
路由组件:Home、Search、Login（没有底部的Footer组件，带有二维码的）、Register（没有底部的Footer组件，带二维码的）

5.2安装路由
```
cnpm install --save vue-router
```
--save:可以让你安装的依赖，在package.json文件当中进行记录
5.3创建路由组件【一般放在views|（或）pages文件夹】
5.4配置路由，配置完四个路由组件 项目当汇总配置的路由一般放置在router文件夹中（src下） 总结：
路由组件与非路由组件的区别？ 1.路由组件一般放置在pages或views文件夹，非路由组件一般放置components文件夹中
2 路由组件一般需要在router文件夹中进行注册（使用的即为组件的名字），非路由组件在使用的时候，一般都是以标签的形式使用， 3.注册完路由，不管是路由組件还是非路由组件身上都是$route、$routers属性

$route:一般获取路由信息【路径、query、params等等】 $router:一般进行编程式路由跳转【push|replace】

5.5路由跳转 路由的跳转有两种形式： 声明式导航router-link，可以进行路由的跳转 编程式导航push|replace，可以进行路由跳转

编程式导航：声明导航能做的，编程式导航都能在，但是编程式导航除了可以进行路由跳转，还可以做一些其他的业务逻辑

5.6Footer组件显示与隐藏

显示或者隐藏组件：（v-if|v-show,v-if需要频繁切换，v-show会好点，但是如果有很多个，用v-show不太方便，不是最终的解决方案，可以用路由的属性meta ）
Footer组件：在Home、Search显示Footer组件　　Footer组件：在登录、注册时候隐藏的

1.我们可以根据组件身上的$router获取路由信息，通过路由路径判断Footer显示与隐藏。有meta属性可以控制，route里面的配置属性都是定的，不能随便乱写
2.配置路由的时候，可以给路由添加路由元信息【meta】,路由需要配置对象，它的key不能瞎写，乱写。

5.7路由传参

1.路由的跳转有几种方式？　　比如a->b
声明式导航：router-link（务必要有to属性），可以实现路由的跳转　编程式导航：利用的是组件实例的$router.push|replace方法，可以实现路由的跳转。（可以书写一些自己业务）
2.路由传参

params参数：路由需要占位，程序就崩了，属于URL当中一部分
query参数：路由不需要占位，写法类似于ajax当中query参数

```
//路由的跳转,采用的是编程式导航.　　//路由传递参数
  //第一种传递query参数
  // this.$router.push({path:'/search',query:{keyword:this.keyword}});

  //第二种传递params参数 [一定要注意,面试的时候经常问]
  // this.$router.push({name:'search',params:{keyword:this.keyword}})

  //第三种传递query+params
  // this.$router.push({
  //  name: "search",
  //  params: { keyword: this.keyword },
  //  query: { keyword: "ABC" },
  // });

  //验证Vue-Router引入Promise技术,最笨的方法,给push传递第二个、第三个参数【回调函数】

  //下面这种写法：治标不治本！！！！
  // let result = this.$router.push({name: "search",params: { keyword: this.keyword|| undefined}},()=>{},()=>{});
```

路由传递参数先关面试题
　1:路由传递参数（对象写法）path是否可以结合params参数一起使用?
　不可以：不能这样书写，程序会崩掉
　2:如何指定params参数可传可不传? 在后面加问号
　3:params参数可以传递也可以不传递，但是如果传递是空串，如何解决？加上undefined
　4:如果指定name与params配置, 但params中数据是一个"", 无法跳转，路径会出问题
　5: 路由组件能不能传递props数据?可以
　　//新增配置项:props,给路由组件传递props参数
　　//第一种布尔模式,相当于把params参数，作为props属性值传递给这个路由组件
　　// props:true,

```
    //第二种:对象形式
    // props:{a:1,b:'我爱你'}

    //第三种写法:函数写法.一般是把query参数与params参数当中props传递给路由组
件!!!
    //route就是当前路由
    // props:(route)=>{
    //      //是将当前箭头函数返回结果，作为props传递给search路由组件!!!
    //      return {a:route.params.keyword,b:'可以传递参数'};
    // }
```

6)创建非路由组件（2个：Header、Footer）

在咱们项目当中，v也在以html+css为主，主要搞业务、逻辑　　在开发项目时：书写静态页面（html+css）、拆分组件、获取服务器的数据动态展示、完成相应的动态业务逻辑
注意：创建组件的时候，组建的结构+组建的样式+图片资源

非路由组件使用分为几步:【一般放在components文件夹】
第一步：定义
第二步：引入
第三步：注册
第四步:使用

非路由组件的结构的搭建：
前台项目的结构与样式不需要自己写的，老师准备好了
辉洪老师静态页面：
结构 + 样式 +图片资源

项目采用的less样式,浏览器不识别less语法，需要一些loader进行处理，把less语法转换为CSS语法

npm install --save less less-loader@5

1：安装less less-loader@5
切记less-loader安装5版本的，不要安装在最新版本，安装最新版本less-loader会报错，报的错误setOption函数未定义

2:需要在style标签的身上加上lang="less",不添加样式不生效

刚开始打开样式有些不对，所以需要清除默认样式reset.css，放入public，在index.html中引入

7)路由的跳转
路由的跳转就两种形式：声明式导航（router-link：务必要有to属性）
　　编程式导航push||replace
编程式导航更好用：因为可以书写自己的业务逻辑

面试题：v-show与v-if区别?
v-show:通过样式display控制
v-if：通过元素上树与下树进行操作
面试题:开发项目的时候，优化手段有哪些?
1:v-show|v-if
2:按需加载
8)首页|搜索底部是有Footer组件，而登录注册是没有Footer组件
Footer组件显示|隐藏，选择v-show|v-if
路由元信息

9）路由传参
params参数：路由需要占位，程序就崩了，属于URL当中一部分
query参数：路由不需要占位，写法类似于ajax当中query参数
路由传递参数先关面试题
1:路由传递参数（对象写法）path是否可以结合params参数一起使用?
不可以：不能这样书写，程序会崩掉
2:如何指定params参数可传可不传?
3:params参数可以传递也可以不传递，但是如果传递是空串，如何解决？
4:如果指定name与params配置, 但params中数据是一个"", 无法跳转，路径会出问题
5: 路由组件能不能传递props数据?

犯的错误:
1:项目阶段，左侧菜单目录，只能有项目文件夹
2:联想电脑安装node_modules依赖包的时候，经常丢包。npm install --save axios --force
3：单词错误
4：路由理解
KV：K--->URL V---->相应的组件
配置路由：
------路由组件
-----router--->index.js

```
import Vue  from 'vue';
import VueRouter from 'vue-router';
//使用插件
Vue.use(VueRouter);
//对外暴露VueRouter类的实例
export default new VueRouter({
    routes:[
        {
            path:'/home',
            component:Home
        }
    ]
})
```

------main.js 配置项不能瞎写

$router:进行编程式导航的路由跳转
this.$router.push|this.$router.replace
$route:可以获取路由的信息|参数
this.$route.path
this.$route.params|query
this.$route.meta

1)编程式导航路由跳转到当前路由(参数不变), 多次执行会抛出NavigationDuplicated的警告错误?
注意:编程式导航（push|replace）才会有这种情况的异常，声明式导航是没有这种问题，因为声明式导航内部已经解决这种问题。
这种异常，对于程序没有任何影响的。
为什么会出现这种现象:
由于vue-router最新版本3.5.2，引入了promise，当传递参数多次且重复，会抛出异常，因此出现上面现象,
第一种解决方案：是给push函数，传入相应的成功的回调与失败的回调
第一种解决方案可以暂时解决当前问题，但是以后再用push|replace还是会出现类似现象，因此我们需要从‘根’治病；

2)将Home组件的静态组件拆分
2.1静态页面（样式）
2.2拆分静态组件
2.3发请求获取服务器数据进行展示
2.4开发动态业务
拆分组件：结构+样式+图片资源
一共要拆分为七个组件

3)axios二次封装
AJAX:客户端可以'敲敲的'向服务器端发请求，在页面没有刷新的情况下，实现页面的局部更新。
XMLHttpRequest、$、fetch、axios
跨域:如果多次请求协议、域名、端口号有不同的地方，称之为跨域
JSONP、CROS、代理
2.1:工作的时候src目录下的API文件夹，一般关于axios二次封装的文件
2.2进度条：nprogress模块实现进度条功能
工作的时候，修改进度条的颜色，修改源码样式.bar类名的

4)vuex:今晚务必vuex复习一下
vuex:Vue官方提供的一个插件，插件可以管理项目共用数据。
vuex：书写任何项目都需要vuex?
项目大的时候，需要有一个地方‘统一管理数据’即为仓库store
Vuex基本使用:

1:重写push与replace方法
工作的时候想处理掉，不想处理对于你的程序没有任何影响

```
function push(){
  return new Promise(resolve,reject){

    }

}
```

2:axios二次封装
-----API：关于请求相关的（axios）
请求、响应拦截器----nprogress进度条

3:vuex
当项目比较大，组件通信数据比较复杂，这种情况在使用vuex

Vuex是插件，通过vuex仓库进行存储项目的数据

1)vuex模块式开发【modules】
由于项目体积比较大，你向服务器发请求的接口过多，服务器返回的数据也会很多，如果还用以前的方式存储数据，导致vuex中的state数据格式比较复杂。采用vuex模块式管理数据。
Vuex核心概念:state、actions、mutations、getters、modules

```
{
  state：{
     a:1,
     b:2,
     c:[],
     d:{}
  }
}

//模块式开发
{
  state:{
    home:{a:1},
    search:{},
    detail:{}
  }
}
```

2)商品分类三级联动展示动态数据

以前基础课程的时候，发请求操作如下：在组件的mounted中书写axios.get||post,获取到数据存储到组件的data当中进行使用

你们写项目的时候发请求在哪里发呀？
mounted|created:都可以

mounted：模板已经变为真是DOM【只不过没有数据，显示空白】，因为ajax是异步，需要时间的。
created：稍微好那么一丢丢（不算啥）

3)商品分类数据猜想？

```
[
  {
    id:1,categoryName:'图书',
    child:[
       {id:3.14,
        categoryName:'影像',
```

```
            child:[
                {id:4,categoryName:'华为'}
            ]
            }
        ]
    }
]
```

4)完成动态展示商品分类的数据

5)完成一级分类的背景效果
第一种解决方案：CSS hover 怎么接单怎么来

6)完成动态展示2|3联动结构

7)演示卡顿现象

8)函数防抖与节流***面试题

正常：事件触发非常频繁，而且每一次的触发，回调函数都要去执行（如果时间很短，而回调函数内部有计算，那么很可能出现浏览器卡顿）

防抖：前面的所有的触发都被取消，最后一次执行在规定的时间之后才会触发，也就是说如果连续快速的触发,只会执行最后一次

节流：在规定的间隔时间范围内不会重复触发回调，只有大于这个时间间隔才会触发回调，把频繁触发变为少量触发

今晚需要把防抖与节流的原理，通过JS实现【闭包 + 延迟器】

9)优化项目。
v-if|v-show
按需加载
函数防抖与节流
按需加载:对于loadsh插件，它里面封装的函数功能很多
import _ from lodash 相当于把全部功能引入进来，但是我们只是需要节流。

10)路由的跳转与传参

第一种声明式导航:为什么使用router-link组件的时候，会出现卡顿那？
router-link是一个组件：相当于VueComponent类的实例对象，一瞬间
new VueComponent很多实例（1000+），很消耗内存，因此导致卡顿。

第二种编程式导航:push|replace

三级分类由于使用router-link的时候，会出现卡顿现象，因此采用编程式导航。

路由跳转的时候【home->search】：需要进行路由传递参数【分类的名字、一、二、三级分类的id】

this.$router.push()

```
{
 name:'search',
 query:{
   categoryName:'电子书',
   category2Id:4
 }
}
```

作业:利用过渡动画transiton，封装一个抽屉的效果。

1:课堂注意
出现一些项目中的提问的问题：比如 OK　1
　　比如：NO　2

2：线上授课
上课的时候如果收到社区人员通知做核酸，和老师说一声！
有任何困难和老说说一声
微信群+添加老师微信+学习问题及时与老师沟通
每一天的上课视频，老师会上传百度网盘

3)复习一下
三级联动业务:
3.1前面基础课程当中v-for很少使用index，以后在写项目的时候，index索引值切记加上
3.2防抖与节流【面试经常出现】
3.3vuex可以模块式开发
vuex经常用的套路是state、mutations、actions、getters、modules

1)搜索模块中的三级联动与过渡动画

1.1在home模块当中，使用了一个功能三级联动功能---->[typeNav]
1.2在search模块当中，也使用三级联动的功能------->[typeNav]
1.3注意的事项
注意1：以后在开发项目的时候，如果发现某一个组件在项目当中多个地方出现频繁的使用咱们经常把这类的组件注册为全局组件。
注册全局组件的好处是什么那：只需要注册一次，可以在程序任意地方使用

注意2:咱们经常把项目中共用的全局组件放置于components里面，以后需要注意，
项目当中全局组件（共用的组件）一般放置于components文件夹中

注意3：全局组件只需要注册一次，就可以在项目当中任意的地方使用，注册全局组件一般是在入口文件注册。

2)组件name属性的作用?
2.1开发者工具中可以看见组件的名字
2.2注册全局组件的时候，可以通过组件实例获取相应组件的名字

3)TypeNav组件业务分析?
3.1三级联动在home模块正常显示
3.2三级联动在search一会显示、一会隐藏 ---解决方案：通过一个响应式属性控制三级联动显示与隐藏
3.3开发的时候的出现问题：在home模块下不应该出现显示与隐藏的效果
3.4现在这个问题【三级联动：本身在search模块应该有显示与隐藏的业务】，但是在home模块下不应该出现显示与隐藏的业务
说白了：你需要让三级联动组件知道谁在用它。
3.5:通过$route让组件区分在那个模块下
以后在功的时候，如果出现某一个组件要区分当前在哪一个模块中【home、search】，通过$route路由信息区分
3.6路由跳转的时候，相应的组件会把重新销毁与创建----【kepp-alive】

4)过渡效果
最早接触的时候:CSS3
Vue当中也有过渡动画效果---transition内置组件完成
4.1:注意1,在Vue当中，你可以给 （某一个节点）|（某一个组件）添加过渡动画效果
但是需要注意，节点|组件务必出现v-if|v-show指令才可以使用。

5)TypeNav三级联动性能优化?
项目：home切换到search或者search切换到home，你会发现一件事情，组件在频繁的向服务器发请求，
获取三级联动的数据进行展示。

项目中如果频繁的向服务器发请求，很好性能的，因此咱们需要进行优化。

5.1为什么会频繁的向服务器发请求获取三级联动的数据那?

因为路由跳转的时候，组件会进行销毁的【home组件的created：在向vuex派发action，因此频繁的获取三级联动的数据】
只需要发一次请求，获取到三级联动的数据即可，不需要多次。
最终解决方案：在App.

5.2:项目性能优化手段有哪些？
v-if|v-show选择
按需加载　　lodash、ant
防抖与节流
请求次数优化

6)开发listContainer|Floor组件业务？
接下来需要开发listContainer与floor组件
场景:开发项目，产品经理画出原型，前端与后端人员需要介入（开发项目），
leader（老大）刚开完会，前端与后端负责哪些模块，后端人员（....开发服务器），
前端人员【项目起步、开发静态页面、查分静态组件】，回首一看回台‘哥哥’，接口没有写好，
向这种情况，前端人员可以mock一些数据【前端程序员自己模拟的一些假的接口】，当中工作中项目上线，需要把mock
数据变为后台哥哥给的接口数据替换。

6.1mock数据。
注意：因为后台老师没有给我们写好其他接口【老师们特意的：因为就是想练习mock数据】
但是项目中mock数据，你就把他当中真实接口数据用就行。

注意：mock（模拟数据）数据需要使用到mockjs模块，可以帮助我们模拟数据。
注意：mockjs【并非mock.js mock-js】
http://mockjs.com/ 官方地址

mock官网一句话：晚上练习的时候，如果网速可以，看看mock的官网，看看语法规则；
生成随机数据，拦截 Ajax 请求
mock官网当中这句话的理解：
模拟的数据一般：对象、数组
```
{
  'a|1-10':'我爱你'
}
```
拦截ajax请求：请求发布出去【浏览器会进行拦截：笨想，因为服务器】，只是项目当中本地自己玩耍数据。

第一步:安装依赖包mockjs

第二部：在src文件夹下创建一个文件夹，文件夹mock文件夹。

第三步:准备模拟的数据
把mock数据需要的图片放置于public文件夹中!
比如:listContainer中的轮播图的数据
[
  {id:1,imgUrl:'xxxxxxxxx'},
  {id:2,imgUrl:'xxxxxxxxx'},
  {id:3,imgUrl:'xxxxxxxxx'},
]

第四步：在mock文件夹中创建一个server.js文件
注意：在server.js文件当中对于banner.json||floor.json的数据没有暴露，但是可以在server模块中使用。
对于webpack当中一些模块：图片、json，不需要对外暴露，因为默认就是对外暴露。

第五步:通过mock模块模拟出数据

通过Mock.mock方法进行模拟数据

第六步:回到入口文件，引入serve.js
mock需要的数据|相关mock代码页书写完毕，关于mock当中serve.js需要执行一次，
如果不执行，和你没有书写一样的。

第七步:在API文件夹中创建mockRequest【axios实例：baseURL:'/mock'】
专门获取模拟数据用的axios实例。

在开发项目的时候：切记，单元测试，某一个功能完毕，一定要测试是否OK

7）存储数据，存储于vuex
7.1:书写静态页面
7.2：拆分组件
7.3：获取服务器数据
7.4：展示数据
7.5：开发动态业务

8)swiper基本的使用？

8.1:swiper可以在移动端使用？还是PC端使用？
答：swiper移动端可以使用，pc端也可以使用。

8.2:swiper常用于哪些场景？
常用的场景即为轮播图----【carousel:轮播图】
swiper最新版本为7版本的，项目当中使用的是5版本

https://www.swiper.com.cn/ 官网地址

------作业：
晚上翻看一下swiperAPI

项目第四天:重要的事情
1:mockjs模块实现模拟数据
---对于将来实际工作的时候，后台没有准备好接口（服务器没有开发出来），前端工程师可以利用mock技术，
实现模拟数据，将来项目上线（后台真实接口）写好了，替换为真实接口即可。
---对于咱们而言，后台老师确实没有给首页中轮播这部分的接口，mock数据，你可以当中一个真实接口就行了。
上线的时候，对于mock数据对于项目而言没有任何影响。

对于项目而言:真实的接口 /api/xxxx 模拟的数据/mock/xxxx
模拟数据JSON：没有空格，最好使用格式化插件进行格式化。

2:swiper插件。
提醒:当年学习过移动端视口、rem、高清图等等。
swiper插件：可以在移动端、PC端都可以使用，这个插件经常可以快速开发轮播图。
前端开发:轮播图、分页器、日历。

Swiper使用步骤：
第一步：引入依赖包【swiper.js|swiper.css】
第二步:静态页面中结构必须完整【container、wrap、slider】，类名不能瞎写
第三步:初始化swiper实例

1:swiper在Vue项目中使用 （开发ListContainer组件【首页banner图片】）
提示：卸载插件，你可以不用删除node_modules文件夹，可以使用npm uninstall xxxx进行卸载
1.1swiper安装到项目当中

1.2在相应的组件引入swiper.js|swiper.css

但是需要注意，home模块很多组件都使用到swiper.css,没必要在每一个组件内部都引入样式一次，
只需要在入口文件引入一次即可。

1.3:初始化swiper实例在哪里书写?
初始化swiper实例之前，页面中的节点（结构）务必要有，
对于Vue一个组件而言，mounted[组件挂载完毕：相应的结构不就有了吗]
mounted-->组件挂载完毕

1.4动态效果为什么没有出来?
Swiper需要获取到轮播图的节点DOM，才能给swiper轮播添加动态效果，
因为没有获取到节点。

1.5第一种解决方案，延迟器（不是完美的解决方案）
同学的想法：都不是完美的【错误的想法】
created里面：created执行与mounted执行，时间差可能2ms，白扯
updated里面：如果组件有很多响应式（data），只要有一个属性值发生变化updated还会再次执行，再次初始化实例。

总结：第一种解决方案可以通过延迟器（异步）去解决问题，
但是这种解决方案存在风险（无法确定用户请求到底需要多长时间），因此没办法确定延迟器时间。

2:Swiper在Vue项目中使用完美解决方案
第一种解决方案问题出现在哪里：v-for,在遍历来自于Vuex（数据:通过ajax向服务器发请求，存在异步）

watch:监听属性，watch可以检测到属性值的变化，当属性值发生变化的时候，可以出发一次。

Vuex当中的仓库数据bannerList（组件在使用）：
bannerList仓库数据有没有发生过变化？
一定是有的：bannerList初始值空数组，当服务器的数据返回以后，它的bannerList存储的属性值会发生变化【变为服务器返回的数据】
组件实例在使用仓库中的bannerList，组件的这个属性bannerList一定是发生过变化，watch可以监听到。

组件实例的一个方法:$nextTick
this.$nextTick(()=>{

})
nextTick官网解释:
在下次DOM更新, 循环结束之后,执行延迟回调。在 修改数据之后 立即使用这个方法，获取更新后的DOM。
注意：组件实例的$nextTick方法，在工作当中经常使用，经常结合第三方插件使用，获取更新后的DOM节点

总结:
1:Swiper插件工作的是很常用（今晚把API、基本使用方法）看看
2:组件实例的$nextTick方法。
在下次DOM更新, 循环结束之后,执行延迟回调。在 修改数据之后 立即使用这个方法，获取更新后的DOM

3)开发Floor组件
开发Floor组件：Floor组件它被复用的（重复使用两次）

3.1:Floor组件获取mock数据，发请求的action书写在哪里?
派发action应该是在父组件的组件挂载完毕生命周期函数中书写，因为父组件需要通知Vuex发请求，父组件
获取到mock数据，通过v-for遍历 生成多个floor组件，因此达到复用作用。

3.2:父组件派发action，通知Vuex发请求，Home父组件获取仓库的数据，通过v-for遍历出多个Floor组件

3.3v-for|v-show|v-if|这些指令可以在自定义标签（组件）的身上使用

3.4组件间通信**面试必问的东西
props:父子
插槽:父子
自定义事件:子父
全局事件总线$bus:万能
pubsub:万能
Vuex:万能
$ref:父子通信

3.5为什么在Floor组件的mounted中初始化SWiper实例轮播图可以使用.
因为父组件的mounted发请求获取Floor组件，当父组件的mounted执行的时候。
Floor组件结构可能没有完整，但是服务器的数据回来以后Floor组件结构就一定是完成的了，因此v-for在遍历来自于服务器的数据，如果服务器的数据有了，Floor结构一定的完整的。
否则，你都看不见Floor组件

4)carousel全局组件
如果项目当中出现类似的功能，且重复利用，封装为全局组件----【不封装也可以】

为了封装全局的轮播图组件:让Floor与listContainer组件中的代码一样，如果一样完全可以独立出来
封装为一个全局组件。

最终:今天项目当中那部分业务有问题（没明白的） ----1

```
项目业务逻辑OK的     ------------------------2
```

复习:重点
1:swiper|lodash|moment插件工作的时候经常使用----【API：有时间翻看一下】
2:$nextTick,组件实例的方法。
在下次DOM更新循环结束之后执行延迟回调。在修改数据之后立即使用这个方法，获取更新后的 DOM。

1)合并参数*
为什么需要合并参数（query|params）:因为这些参数，对于search是有用的，因为search通过这些参数
向服务器发请求，需要把这些参数携带给服务器，服务器就会返回相应的用户的搜索的数据，search就可以进行展示。

1.1:开发的三级联动业务，当你点击a标签的时候，会进行路由的跳转，将产品的名字与id传递给search模块----（query）
1.2:点击搜索按钮的时候，用户输入进来的关键字，点击按钮的时候会通过params参数传递给search模块-----（params）
1.3路由跳转（home->search）,两个地方，三级联动（typeNav）、Header组件（搜索按钮）

2)完成search静态组件
项目节点:学习的并不是基础的语法，'套路'
再次提醒：组件通信很重要-----【七种组件通信：务必要会】
接下来开发search搜索模块：注意在老师给你们的文件夹中有search静态组件，复制过来即可。
2.1书写静态页面【布局、样式】
2.2拆分组件
2.3获取服务器数据展示数据
2.4玩业务

//是搜索模块需要携带给接口的参数
{
  "category1Id": "61",//一级分类的id
  "category2Id": "61",//二级分类的id
  "category3Id": "61",//三级分类的id
  "categoryName": "手机",//产品的名字
  "keyword": "小米",//关键字
  "order": "1:desc",//排序
  "pageNo": 1,//当前第几页
  "pageSize": 10,//每一页需要展示多少条数据
  "props": ["1:1700-2799:价格", "2:6.65-6.74英寸:屏幕尺寸"],//平台属性的选择参数
  "trademark": "4:小米"//品牌参数
}

注意：搜索的接口，需要传递参数，至少是一个空对象（KV没有，但是至少给服务器一个对象）
3)获取search模块数据
4)展示商品列表数据

5)根据用户的搜索条件展示不同的数据。

根据前台传递参数决定的
根据不同条件，展示不同的数据。----->取决于后台返回的数据

1:发请求，获取搜索模块的数据
2:根据用户搜索的条件携带参数给服务器，展示用户搜索的内筒

开发遇见问题:用户条件可以发生多次变化，但是咱们的请求，只是会发一次【mounted中书写的】

请求的性能优化:
发一个请求，需要向服务器携带参数：带100个参数 带1参数 【消耗宽带】
对于给服务器携带的参数：如果数值为undefind，向服务器发请求的时候，参数步携带给服务器的

6)面包屑的业务完成
-----务必把今天的套路---书写三遍
'作业':什么是富文本，插件有哪些，一些如何使用【在Vue当中使用】

7）另外一件事情？

豪哥就是带你们到前台项目结束。

项目第七天:
1:获取到search模块的数据
2：对于商品的展示
3:对于关键字、三级联动的面包屑展示业务

学会套路最重要:
套路1:路由自己跳自己----修改路由
套路2：watch监听路由的变化发请求

切记：自己写项目的时候，切记进行单元测试---【完成一个功能：要验证是否OK】
　　在你写项目的时候，锻炼自己解决bug问题--【别遇见红色就怕】

1)品牌与平台属性的数据进行动态展示
tradeMark---品牌
举例子:理解平台属性 【用户购买一个手机】
颜色【平台属性】:红色、白色、紫色【平台属性值】
价格【平台属性】：1299,6999,899【平台属性值】
操作系统【平台属性】：window、linux【平台属性值】
//看见页面结构，大概能知道数据结构什么样子的
尺寸: 中、短
材料：塑料、涤纶
```
[
  {attrId:1,attrName:尺寸,attrValueList:['中','短']},
  {attrId:2,attrName:材料,attrValueList:['塑料','涤纶']},
]
```

2)完成品牌 与 平台属性的业务
2.1刚刚我们只是把服务器的数据动态展示，但是需要注意，对于品牌|平台属性、用户可以点击的【小米、苹果】|平台属性
2.2我们还是需要收集用户选择的数据，把用户选择的数据信息，给服务器传递获取，获取相应的数据进行展示
2.3组件通信-----（工作使用频率非常高、面试的时候经常出现）
父子:props、插槽、ref
子父：自定义事件
万能：vuex、$bus、pubsub

经典面试题:数组去重[1,2,2,3,4,2];

平台属性携带参数格式:
props Array N 商品属性的数组: ["属性ID:属性值:属性名"] 示例: ["2:6.0～6.24英寸:屏幕尺寸"]

props:['属性的ID:属性值:属性名']

3)完成排序业务
num1:在基础课程当中曾经写过排序业务。

num2:综合与价格按钮，点击谁，谁的背景颜色变为红色。（类名：active）
谁有类这件事情，区分开综合与价格

num3：将来点击综合||价格，还是需要给服务器发请求
【价格升序：把这个信息给服务器传递过去，服务器接收到信息，数据库自动把排序这件事情做了，把排序做好的数据返回给你，你展示即可】

order:服务器需要字段，代表的是排序方式
order这个字段需要的是字符串（可以传递也可以不传递）
1:代表综合
2:代表价格
3:asc代表升序
4:desc代表降序
告诉服务器排序方式有几种情况?
"1:asc" "1:desc" "2:asc" "2:desc"

num4:综合与价格箭头

4.1箭头用什么去做【可以选用阿里图标库】 https://www.iconfont.cn/

4.2对于综合|价格旁边的箭头【动态显示：时而又，时而没有】，带有类名active，拥有箭头

4.3:根据1、2区分谁有类名（背景）、谁有箭头
根据asc|desc区分它用哪一个箭头【上、下】

5)分页功能。
第三方插件:elementUI实现超级简单
但是咱们需要自己封装。也属于前台项目当中比较重要的一部分。

1)复习
1.1品牌业务的完成
1.2完成平台属性业务
1.3完成排序业务
总结：
注意1:在你书写项目的时候,需要注释!
注意2:单元测试，完成一个功能测试是否OK【打印、vue开发者工具】
注意3：每一次书写的时候，都有小bug【解决bug能力】

2)分页器业务
前端三大件:轮播图、分页、日历。这属于前端开发常见三种业务

2.1:为什么很多项目中都采用分页功能?
比如电商平台：搜索一个奶粉，奶粉的产品有10000+，一次渲染10000+条数据，可能慢。
数据多的时候，可以选择分页，比如每一次只是展示10

2.2拆分分页组件（静态组件），注册为全局组件，因为其他模块也在使用分页功能。

面试当中:你自己封装过一个通用的组件吗?-----分页组件 **

分页器封装业务分析:
封装分页器组件的时候：需要知道哪些条件？
假如你知道条件1、条件2：知道一共多少页 100/3
1:分页器组件需要知道我一共展示多少条数据 ----total【100条数据】

2:每一个需要展示几条数据------pageSize【每一页3条数据】

3:需要知道当前在第几页-------pageNo[当前在第几页]

4:需要知道连续页码数【起始数字、结束数字：连续页码数市场当中一般5、7、9】奇数，对称好看 continues

已经条件: total=【99】 pageSize =【3】 pageNo=6 continues 5

4 5 6 7 8

已经条件: total=【99】 pageSize =【3】 pageNo= 1 continues 5

错误:-1 0 1 2 3
正确: 1 2 3 4 5

已经条件: total=【99】 pageSize =【3】 pageNo= 2 continues 5

错误: 0 1 2 3 4
正确：1 2 3 4 5

已经条件: total=【99】 pageSize =【3】 pageNo= 33 continues 5

错误: 31 32 33 34 35
正确：29 30 31 32 33

已经条件: total=【99】 pageSize =【3】 pageNo= 32 continues 5

错误：30 31 32 33 34
正确: 29 30 31 32 33

3)分页器封装
3.1进行单元测试

连续页码5: 8 [6,7,8,9,10]
连续页码7: 8 [5,6,7,8,9,10,11]

连续页码5: 8 [6,7,8,9,10]
连续页码7: 8 [5,6,7,8,9,10,11]

经典面试题：v-for与v-if优先级？ v-for优先级更高

//正常情况：再回来因该还是第一页【遇见脑袋xxxx产品可能有这种操作】
4)需求：最后这个需求可以书写、可以不书写【正常说：没有这个需求的】
比如:2021年10月30日11:47:44 点击分页器 第四页 ->网站关闭了
但是2021年11月11日11:48:12 打开这个项目 第四页 -->本地存储

总结:
面试问题：v-for与v-if优先级?
    工作当中是否自己封装过一些通用的组件？
对于一个分页器:
1)需要知道数据总条数
2)每一个需要展示数据条数
3)知道当前是第几页
4)连续页码数字
5)自定义事件【子给父通信的】

5)push与replace区别?
编程式导航：push 与 replace
能不能记录历史记录：push（能记住历史记录） replace（不能记住历史记录）
目前项目当中：进行路由跳转（编程式导航）基础都是push
push与replace是有区别的

6)开发详情业务
6.1:熟悉静态页面、书写样式
6.2：拆分组件
6.3:获取服务器动态展示
6.4：完成动态业务

7)滚动行为的设置。

8)项目当中控制台

vue-warn:警告（不影响的你程序），对于你的代码提出一个警告。
对于程序没有任何影响，俗称假报错。

切记:
1:学套路------遇见一些业务，怎么去解决
2:及时复习----将前面套路，自己梳理一遍
3:前台 + 后台项目-----解决bug能力

复习:
前端开发三大件:轮播 + 分页 + 日历
1)分页器功能
需要知道一共展示多少条数据
需要知道每一页展示几条数据
需要知道当前是第几页
需要知道连续页码数

2)详情模块注意事项
vuex:时不时会有假报错现象

3)详情模块开发之商品属性的开发
排他操作：在工作中经常使用-----千万别忘记

```
{
  attr:'颜色',
  attrValue:['红色','黑色','白色']
}
```

4)注意
在工作中假报错现象很常见，因为什么导致的，尽可能解决掉-----【不解决掉对于你的程序没有任何影响】

5)放大镜的功能
----插件:插件解决可以【巧劲】

5.1遮罩层为什么能动。
获取节点（DOM：必须要定位），通过JS动态修改left|top、定位元素才有left、top属性

6)产品个数业务
以后项目当中：凡是出现文本框【用户输入：一定有'幺蛾子',思考情况一定要多思考】

7)加入购物车的业务？购物车项目第二个重要地方
购物车：每一个人都有属于自己的购物车，那为什么不同用户登录自己账号，可以看见属于自己产品
一定是用户点击加入购物车，把你的产品信息提交给服务器进行保存，当你下次在进入购物车的时候，
需要向服务器发请求，获取你购物车里面的信息展示

项目：点击加入购物车按钮的时候，以前经常进行路由跳转【调到另外一个路由】，
但是你要注意，点击加入购物车这个按钮的时候，将用户选择产品，提交给服务器进行存储，如果服务器存储成功，
之后在进行路由跳转

面试题：GET与POST
相同点：都是HTTP协议。
不同点:
1:GET请求携带参数是有上限的 post请求携带的参数是没有'上限的'
2:GET请求相对而言不安全，POST安全

面试题:H5新增那些特性？
CSS4、本地存储、多媒体、canvas

面试题：本地存储与会话存储区别？

本天:项目的第十天，今天课程重点，完成购物车业务。*
前台项目三个重要地方：分页器、购物车、登录注册。

6)面试题:token相关的面试

切记:Vue项目,React尽可能别扔JSX。
情况:外包公司【Vue、React】

1)复习
问题1:什么时候用router-link、什么时候用编程式导航。
1.1如果一个按钮点击过后，如果只有路由跳转的业务【声明式导航、编程式导航】
1.2如果点击一个按钮除了路由跳转，还有其他的业务，可以选择编程式导航。

问题2：promise问题。【基础问题】

2)加入购物车成功组件的业务?
路由跳转

3)获取购物车的数据进行展示?

举例子:用户是淘宝平台的用户。
为什么目前我们获取不到自己购物车的数据，你没有给我分配一个用户id
张三:奶粉、鞋子、手机
李四:羽绒服
3.1问题1：用哪个技术可以生成用户id【身份】----uuid
3.2问题2:用户身份如何给后台专递过去？
3.3临时身份只需要执行一次
3.4临时身份数据持家化的
3.5工作的时候不这么玩
会创建一个utils（工具）文件：把常用的代码片段放到这个文件夹里面
3.6配置一些文件[JS]，不能操作仓库
配置文件不限执行，没办法运行项目【配置文件很少碰仓库】

4)设计购物车的数据？
注意：获取购物车的数据的时候，读取的时候切记小心。后台老师写的数据格式有问题的。
张三:衣服、裤子、鞋子

```
[

]

[
   {
      cart:[   {name:'衣服'},{name:'裤子'},{name:'鞋子'}]
   }
]
```

5）购物车静态结构需要注意
头部:6
身体:7
静态页面需要删除一些：把每一个产品的第三个li删除
需要修改每一个li的百分比:
[con1 2 3 4 5 6 7]的百分比
15 35 10 17 10 13

6)购物车数量的操作?
修改购物车产品数量的时候，需要发请求的，通知服务器产品最新的个数【服务器需要保存】，
当你组件在获取购物车的数据时候，不就是最新的数值【用户刷新刷不掉】

产品个数变化接口参数：
skuID string Y 商品ID

skuNum：在修改产品个数的时候,需要给服务器传递的是【变化的数值】

比如：佩奇 起始数量 4 用户在表单元素中输入 6 ----->给服务器参数是2
佩奇 起始数量4 用户在表单元素中输入1 ------>给服务器的参数-3
佩奇 起始的数量4 用户在表单元素中输入4 ------>给服务器的参数0

blur:失去焦点--->点击空白的地方
change:文本需要有变化，而且还需要点击空白的地方
input:只要文本发生变化立马执行【不许点击空白的地方】

day11:重点*
1:购物车完成
2:登录注册
注意:有的时候node_modules会丢包,删除重新安装就行了。
注意:函数防抖与节流【面试经常出现】
注意:用户有两个身份【临时游客、用户】

1)删除购物车中的产品的操作
面试题:
HTTP请求GET|POST
你常用的HTTP请求有哪些:GET|POST|DELETE等等

2)购物车产品选中与未选中业务
购物车中产品的数据：isChecked属性，1：代表这个产品勾选中 0:代表这个产品没有被选中

注意:以后工作的时候，最基本的基本功看API文档（在线文档），在开发后台管理系统项目的时候
一定要培养看在线文档能力

3)全选的业务
3.1以后在工作当中出现了一些，想不明白、或者没有思路。【前一个月：有问题，尽可能别问同事】
3.2全选的复选框业务
目前而言：是没有这个接口，一次修改全部产品的选中状态接口【正常工作当中一定是有这样的接口：一次全部修改选中状态】
全选复选框：如果它勾上，顶上全部的产品的选中状态，勾上
全选复选框：如果它没勾上，顶上的全部产品的选装中台，没勾上

vuex:minStore[小仓库state、getters、dispatch、commit]

3.3promise:毕业的时候，手写promise
将来毕业：手写Promise【能记住，毕业就忘掉】

前台项目:分页器、购物车、登录与注册业务

4)登录注册*
对于企业当中，一般项目都有登录注册功能【这个业务很重要】
当然有一些项目不需要注册，后台管理系统项目，一般不需要注册。

4.1登录与注册的静态组件（图片问题会报错）

4.2assets【放置静态资源文件的地方】
一般放置所有组件共用的静态资源
在样式当中也可以使用@,在样式当中使用@，前面加上~

4.3注册的业务
4.3.1:今天在做注册、登录业务的时候，先不处理表单的验证功能，在项目最后一天，在把表单如何验证，如果是那哪些插件解决【最后去处理】
正则
手机号:11
验证码:4-6
登录密码|确认密码:首字母大写、包含英文、数字、特殊字符等等。

4.3.2获取验证码
/api/user/passport/sendCode/{phone}

token面试题:项目当中token过期、失效如何处理？
答：清除本地token（本地存储），让用户回到登录页，获取最新的token

项目day12:今天课程重点【登录注册业务】

1.1:注册业务已经完成【手机号、验证码、登录密码】，点击注册按钮的时候，需要把这些信息给服务器传递过去
手机号:11位
验证码：4-6
登录密码：英文字母、数字、首字母大写等等

1.2登录业务
当你点击登录按钮的时候，需要把手机号、密码需要携带给服务器，服务器需要判断，你是不是我的用户【注册过的】
如果是用户登录成功，进行登录，如果用户登录失败给一个提示即可。

1.3token【令牌：字符串，服务器下发给用户的身份凭证】
举例子:古代大战，兵符

一、用户登录以后获取用户信息进行展示

企业项目:登录成功以后，服务器会返回token【存储于vuex当中】，如果想获取用户信息还需要再发请求【用户信息】，携带token给服务器。
api/user/passport/auth/getUserInfo 获取用户信息的接口

1.1:为什么刷新页面，用户信息就消失
用户刷新页面，用户信息消失没了获取不到，因为token没有携带给服务器。
Vuex存储数据是否持持久化的?并非持久化
1.2：本地存储持久化存储token

1.2为什么去别的模块【非home模块】获取用户信息失败？
因为你去别的模块根本没有发请求获取用户信息，没办法展示用户信了

怎么解决:
每一个组件都在mounted里面发起获取用户信息，进行展示（可以太麻烦）
残留的问题：用户在home模块刷新的时候，用户信息一直在展示（mounted执行的时候在向服务器发请求、获取用户信息展示）

home->search[用户信息刷新数据就没了，因为在search模块当中根本没有发请求获取用户信息]
search-detail[根本没有获取用户信息进行展示]

二、退出登录
2.1发请求，需要通知服务器，把现在用户身份token【销毁】
2.2清除仓库数据+本地存储数据【都需要清理】

三、演示一些操作，你看一下是否正常?
用户已经登录了，用户想从home路由跳转到login路由，不应该这么操作了。
现在用户登录以后，home路由不应该跳转到login路由当中【因为登陆了】，
但是咱们以现在认知【技术】，没办法约束从home调到login

四、导航守卫*
守卫条件判断*

导航:表示路由正在发生改变。
守卫:古代的守门的士兵'守卫'，守卫可以通过条件判断路由能不能进行跳转。

三大守卫:

全局守卫：
项目当中任何路由变化都可以检测到，通过条件判断可不可以进行路由跳转。
前置守卫：路由跳转之前可以做一些事情。
后置守卫：路由跳转已经完成在执行。

```
//全局守卫:[后置守卫:在路由跳转完毕之后才会执行一次]
router.afterEach(()=>{
    console.log('守卫:路由跳转完毕才会执行一次')
})
```

4.1用户已经登录了，不应该在访问login？【通过什么条件能判断用户登录、未登录】
路由独享守卫：
针对某一个路由的守卫

组件内守卫：
也是负责某一个路由守卫

5)身份凭证?
以后登录：
TOKEN身份为大
5.1UUID生成的临时省份
5.2用户（注册与登录）token【正式身份】

项目十三天:
发现:基础知识点-->回首复习
分页
购物车
登录、注册

1)交易业务
前面课程当中可能自己已经注册了一个账号【18666666661】，今天在做支付的时候，统一使用
账号:13700000000
密码:111111

1.1获取用户地址信息、获取用户购物车清单信息
//用户地址信息
/api/user/userAddress/auth/findUserAddressList
//商品清单接口
/api/order/auth/trade

1.2Vuex的action发请求，但是从今天开始，咱们要练习不用Vuex改如何开发？
请求配置,类似$bus使用

2)展示商品清单数据

3)提交订单业务

当用户点击提交订单按钮的时候，需要发请求的
提交订单的请求地址:/api/order/auth/submitOrder?tradeNo={tradeNo}

前台：需要告诉服务器：谁买的、买的啥、买几个、 支付多少钱、留言信息...
后台：订单号，这笔交易的一个标识符【支付的】

axios({url:'xxx',methods:'post',data:{a:1}})

3.1微信支付、支付宝支付等等
交易编码（服务器：字符串hash）
收件人名字
收件人手机号
收件的地址
买家留言信息
支付产品

4)获取支付信息进行展示

5)element-ui官方UI组件库（插件）？

react框架:
UI组件库antd【蚂蚁金服旗下PC端UI组件库】
antd-mobile【蚂蚁金服旗下的移动端UI组件库】

Vue框架:
element-UI【饿了吗旗下的UI组件库，官方承认的PC组件库插件】
vant【Vue官方提供移动端UI组件库】

官网地址:https://element.eleme.cn/#/zh-CN
官网地址：https://youzan.github.io/vant/#/zh-CN/

第一步：项目中安装element-ui组件库 [2.15.6版本：Vue2]

第二步：在入口文件引入elementUI组件库
第一种：全部引入【不采用：因为项目中只是用到一个组件，没必要全都引入进来】
第二种：按需引入【按照开发需求引入相应的组件，并非全部组件引入】

第三步：按需引入，安装相应的插件
cnpm install babel-plugin-component -D
文档中说的.babelrc文件，即为babel.config.js文件
修改完babel.config.js配置文件以后，项目重启

第四部：按照需求引入相应的组件即可

Vue.component();
Vue.prototype.$xxx = xxx;

6)支付业务【微信支付】
this.$alert('这是 ***HTML*** 片段', 'HTML 片段', {dangerouslyUseHTMLString: true});
6.1今晚稍微把elementUI的组件都稍微看看。
6.2使用messageBox显示弹框
6.3展示二维码----qrcode插件
通过qrCode.toDataUrl方法，将字符串转换为加密的在线二维码链接，通过图片进行展示。
moment.js
swiper.js
nprogress.js
qrcode.js

GET|POST：短轮询，请求发一次，服务器响应一次，完事。

第一种做法:前端开启定时器，一直找服务器要用户支付信息【定时器】

第二种做法:项目务必要上线 + 和后台紧密配合
当用户支付成功以后，需要后台重定向到项目某一个路由中，将支付情况通过URL参数形式传给前端，
前端获取到服务器返回的参数，就可以判断了。

项目day14:
分页、登录注册、购物车、支付->这几个业务有疑问的地方及时喊我
elementUI:今晚在稍微看看【后台管理系统项目：全都是用elementUI】

1)个人中心路由搭建
1.1当年学习路由的时候:一级路由、二级路由、三级路由 【二级路由搭建】
1.2完成个人中心数据的展示【分页】

2)未登录全局守卫的判断

在前面课程当中:导航守卫【导航:路由发生变化，守卫可以检测到，通过判断，确定这次路由跳转】

前置守卫：在路由跳转之前，进行判断
后置守卫:路由都已经跳转完毕才执行。

未登录的情况:
全局守卫:只要的项目当中任何某一个路由发生变化，就会出发。
项目守卫使用:一般有用前置全局守卫

用户登录:

用户未登录：点击购物车的结算按钮->交易页面【没有登录:去不了】
未登录不能调到支付页面
未登录不能调到支付成功页面
未登录不能去个人中心【都不知道你是谁：展示谁的个人中心啊】

3)路由独享守卫
路由独享守卫：需要在配置路由的地方使用
导航守卫:全局守卫->项目当中有任何路由变化【a->b,b->d】触发。
路由独享守卫：专门负责某一个路由

用户登陆了:
去交易页面:从购物车才能跳转到交易页面。

next():你本来想去哪里，我就放行，你就去完事了。

next('/login'):执行守卫放行到执行的路由。

next(false):路由跳转的时候，从哪里来回那里去。

4)组件内守卫---->一般很少用【全局 + 路由独享守卫】
组件内守卫：也是专门负责某一个路由【并非负责全部路由】，写法和路由独享守卫有区别?
组件内守卫需要书写在组件内部

beforeRouteEnter
beforeRouteUpdate (2.2 新增)
beforeRouteLeave

6)路由懒加载
面试【高频的面试】:项目的性能优化手段有哪些?
v-if|v-show:尽可能采用v-show
按需引入【lodash、elementUI】
防抖与节流
路由懒加载：当用户访问的时候，加载对应组件进行展示。

7)图片懒加载
vue-lazyload:图片懒加载
图片：比用用户网络不好，服务器的数据没有回来，
总不可能让用户看白色，至少有一个默认图片在展示。

8)表单验证【后台管理系统：大量使用elementUI】
以后工作的时候经常会进行表单验证【element-ui】进行表单验证，so 简单。
项目当中表单验证功能比较常见的。

8.1vee-validate插件：Vue官方提供的一个表单验证的插件【老师接下来的操作能大概看懂即可】
这个插件很难用：如果你翻看它的文档（看一个月：不保证能看懂），依赖文件很多（文档书写的很难理解）
花大量时间学习，很难搞懂。

8.2哪怕将来工作了，真的使用vee-valadiate【老师项目搞出来：改老师代码即可】

使用步骤：
1：安装vee-valadite，别安装最新版本@2
2：在plugins文件夹中创建一个validate.js[专门注册vee-valadite]
3:注册插件
4：注册插件的时候，用中文，以及需要验证的字段【用中文显示提示形式】
5：在入口文件需要引入执行一次
6:使用vee-valadiate插件

8)vee-validate 基本使用

第一步：插件安装与引入
cnpm i vee-validate@2 --save 安装的插件安装2版本的

```
import VeeValidate from 'vee-validate'
import zh_CN from 'vee-validate/dist/locale/zh_CN'  // 引入中文 message
Vue.use(VeeValidate)

第二步：提示信息
VeeValidate.Validator.localize('zh_CN', {
messages: {
...zh_CN.messages,
is: (field) => `${field}必须与密码相同` // 修改内置规则的 message，让确认密码和密码相同
},
attributes: { // 给校验的 field 属性名映射中文名称
phone: '手机号',
code: '验证码',
password:'密码',
password1:'确认密码',
isCheck:'协议'
}
})

第三步：基本使用
<input type="text" placeholder="请输入你的手机号" />
{{ errors.first("phone") }}

const success = await this.$validator.validateAll(); //全部表单验证
//自定义校验规则
//定义协议必须打勾同意
VeeValidate.Validator.extend('agree', {
validate: value => {
return value
},
getMessage: field => field + '必须同意'
})
```