关于lodash使用，补充常用数组，对象、字符串的方法，

对象的浅克隆object.assign();

对象的深度克隆

this.$router.push('manage')

子组件传递父组件数据this.$emit(‘on-change’,this.selections[index])

父组件函数接受传过来的参数，并触发事件

数据引起页面变化

对对象或者数组数据的变化

this.array.push()

vue.set(target,key,value)

可以添加数据，也可以删除数据

Slice concat不会引起数组的更新

引起页面的更新变化

:class=”{active:index===nowIndex}”

绑定 class  
：class=“{‘line-last’:index%2===0}”

v-for=”item in product”循环里面哪一个要去掉某一个样式，可以添加状态数据对象里面添加属性last=true；

v-if=”item.last”就可以专门选出这一项，添加或者取消class

写页面时可以先写一些假数据，然后再渲染

**关于路由 v-route**

<router-link v-for="item in list" :to="{ path:item.toKey }" tag="li" active-class="active">{{ item.title }}</router-link>

路由访问，浏览器通过path访问，route-link标签访问，js里面也可以访问

在组件中：  {{$route.params.color}}  
在js里： this.$route.params.color获取参数

:to="{path: 'shop', query:{geohash, id: item.id}}"

router.push({path: 'apple', query: {color: 'red' }})

ref 就是指获取节点

This.$ref 返回组件实例

用在DOM上，返回DOM，可以使用原生DOM API

nextTick 解决DOM内容刷新的

better-scroll

vue-charts

通常是给元素order-bottom: 1px solid #ddd;，然后通过transform: scaleY(.5)缩小0.5倍来达到0.5px的效果

Promise/ajax/async/await

屏幕适配/移动端布局

Vue双向绑定原理

webpack配置会自己修改

五星好评点几颗星亮几颗，用css

查询字符串中出现最多次数的字符

Object.assign({},)主要的用途是用来合并多个JavaScript的对象

Object.values()

new Promise((resolve,reject)=>{

})`

created (){获取资源}

alias 设置读取文件路径

keep-alive

slot 插槽

关于数据加载加载的先后很重要

@load=”loadImage”

懒加载，只请求一部分图片，其他在划动时在就在家

Vue-lazyload插件 图片做懒加载，看一下源码，

其实前端vue组件和用jquery插件一样

看到需求自己也想想怎么做，再看作者怎么做的？

不使用造好的 轮播组件，自己做一个轮播组件

watch 里面如何设置变化的参数

如果数据生成的慢，可以先写在方法methods里面，在watch里面监听这个方法

Vue里面计时器 使用箭头函数

增加一些内容测试试试