**数组API**

**forEach遍历数组**

tabs.forEach((v,i) => i===index ? v.isActive=true:v.isActive=false)

这个方法会直接导致原有数组的内容发生修改

**every判断所有数组所有元素都满足条件**

cart.every(v=>v.checked)

对数组进行遍历 只有当元素的每一项都返回true时 方法的结果才是 true

**some判断至少某一元素满足条件**

let isCollect = collect.some(v => v.goods\_id===this.GoodsInfo.goods\_id)

这个遍历数组的方法 只要有一个元素满足条件 就会返回true

1. sort排序

arr.sort(function(a,b){rerurn a-b})

**map 对数组重新封装，形成新数组**

 const urls = this.GoodsInfo.pics.map(v=>v.pics\_mid)

map方法不会修改原来的数组，而是在遍历过程中 根据条件产生一个新的数组

**Filter 过滤符合条件的元素**

let checkedCart = cart.filter(v=>v.checked)

Filter()会根据条件返回一个新的数组

**findIndex找到元素处在数组中的下标**

let index = cart.findIndex(v=>v.goods\_id===this.GoodsInfo.goods\_id)

**reduce 给定初始值按条件遍历数组 得到最终的结果**

chatMsgs.reduce((preTotal, msg) => preTotal+(!msg.read&&msg.to===userid?1:0), 0)

preTotal的初始值为0 在对消息数组进行遍历时 判断消息是否已读并且消息的发送对象是登录用户 如果是的话 未读消息数量加1