**数组的**[**变异方法**](https://cn.vuejs.org/v2/guide/list.html#%E5%8F%98%E5%BC%82%E6%96%B9%E6%B3%95)

Vue 包含一组观察数组的变异方法，所以它们也将会触发视图更新。这些方法如下：

* push()
* 往数组最后面添加一个元素，成功返回当前数组的长度
* pop()
* 删除数组的最后一个元素，成功返回删除元素的值
* shift()
* 删除数组的第一个元素，成功返回删除元素的值
* unshift()
* 往数组最前面添加一个元素，成功返回当前数组的长度
* splice()
* 有三个参数，第一个是想要删除的元素的下标（必选），第二个是想要删除的个数（必选），第三个是删除后想要在原位置替换的值（可选）
* Splice(startIndex,count,replaceVal)
* 常用：splice(index,1) 删除序号为index的元素
* sort()
* 使数组按照字符编码默认从小到大排序,成功返回排序后的数组
* reverse()
* 将数组倒序，成功返回倒序后的数组

你打开控制台，然后用前面例子的 items 数组调用变异方法：example1.items.push({ message: 'Baz' })

数组常用函数

foods.indexOf(food) 返回food在数组中的下标

1.map() 对每个元素的(key,value)进行处理，保留指定的,但元素个数不变

const newItems= items.map(item=>({

url:item.html\_url,

avatar\_url:item.avatar\_url,

name:item.login

}))

2.将一维数组按个数拆分为二维数组

categorysArr(){  
 //需要拆分的数组  
 **const** {categorys}=**this** //创建大的数组，用于存下的数组[][]  
 **const** arr=[]  
 //创建小的数组，用于存category的数组[]  
 **let** minArr=[] //此处不能用const,需要创建新的  
 categorys.forEach(c=>{  
 //判断小数组是否装满了n个，此处设置为8，满了之后创建新的小数组  
 **if**(minArr.length===8){  
 minArr=[]  
 }  
 //判断是否为空，为空则装入大数组  
 **if**(minArr.length===0){  
 arr.push(minArr)  
 }  
 //将元素存入小数组  
 minArr.push(c)  
 })  
 **return** arr  
}

注意：两个if判定不能颠调，否则只能存入一个小数组

3.reduce——统计数量，累加器

