### **在组件中分发 Action**

### 你在组件中使用 this.$store.dispatch('xxx') 分发 action，或者使用 mapActions 辅助函数将组件的 methods 映射为 store.dispatch 调用（需要先在根节点入 store）：

**import** { mapActions } **from** 'vuex'  
**export default** {

// ...  
 methods: {  
 ...mapActions([  
'increment', // 将 `this.increment()` 映射为`this.$store.dispatch('increment')`   
 // `mapActions` 也支持载荷：  
 'incrementBy' // 将 `this.incrementBy(amount)` 映射为 `this.$store.dispatch('incrementBy', amount)`  
 ]),  
 ...mapActions({  
 add: 'increment' // 将 `this.add()` 映射为 `this.$store.dispatch('increment')`  
 })  
 }}

**...mapActions([   '你要获取的actions名字'   ])**，你传入什么，就会返回一个包含对应的actions的对象，通过...对象扩展运算符，将返回值解析就成为methods的属性，和平常的方法一样调用即可

**mapActions、mapMutations、必须先引用且要放在methods中。**

**mapState、mapGetters必须引用并且放在computed中**

Vue组件必须有<div> 包着？ 要不会报错

（产生一个数组）二维数组 这是一个二维数组里面还有一个一维数组

categorysArr () {  
 // 1.先从当前组件中得到所有食品分类的一维数组  
 **const** {categorys} = **this** // 2.准备一个空的二维数组--categorysArr  
 **const** arr = []  
 // for (let i = 0, len = categorys.length; i < len; i += 8) {  
 // arr.push(categorys.slice(i, i + 8))  
 // }  
 // 3.准备一个小数组--pages(最大长度为8)  
 **let** minArr = []  
 // 4.遍历categorys得到处理后的二维数组catagorysArr  
 categorys.forEach(data => {  
 // 如果当前小数组(pages)已经满了, 创建一个新的  
 **if** (minArr.length === 8) {  
 minArr = []  
 }  
 // 如果minArr是空的, 将小数组(pages)保存到大数组(categorysArr)中  
 **if** (minArr.length === 0) {  
 arr.push(minArr)  
 }  
 // 将当前分类信息保存到小数组(pages)中  
 minArr.push(data)  
 })  
 **return** arr  
}

 $mount 这里也是类似的,就是你写了一个 Vue 类,但是不能控制使用场景。所以就要有一个 $mount 方法,由使用者控制,绑到目标 DOM 上。有些时候你创建 vm 的时候 DOM 可能还没准备好,或者写 vm 的时候不确定它要挂在哪个 DOM 上,这个时候就有用了。

vm.$mount() 手动地挂载一个未挂载的实例