1. 真实数据调用。

1>获取数据

一般都是用POST请求来做添加功能,且用服务器的数据请求的数据一般挂载到create钩子函数上请求数据(一进页面数据就请求到了)

2>添加数据

里面的status是人为在json数据几面设置的一个对象。

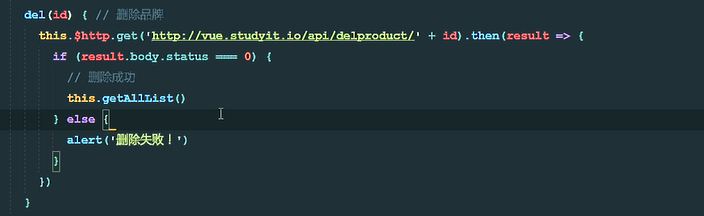
添加的POST请求最好配置下位表单格式数据发送不然可能会报错(emulateJSON:true)



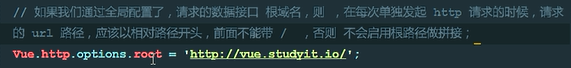
里面调用手动调用this.getAllList();主要是为了将最新添加的数据渲染到页面以达到最新的数据。



3>删除数据



二．配置全局服务器地址(为了方便修改服务器修改地址维护)



下面的获数据---》注意看url地址



三．全局启动emulate:true选项

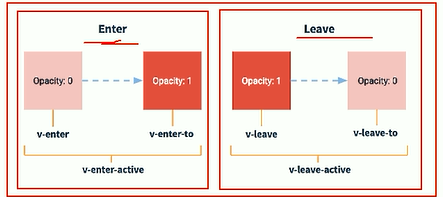
-🡪

Vue.http.options.emulateJSON = true;

四．动画在vue都分为两部分1>进入部分.2>离开部分。分为这两个半场动画

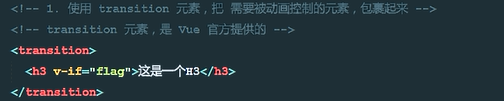
进入部分的结束状态和离开部分起始状态样式一样，

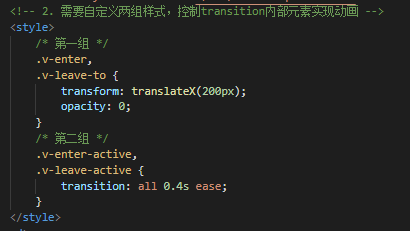
进入部分起始状态和离开部分结束状态样式一样



五．动画

1>第一种方法过渡类名实现动画

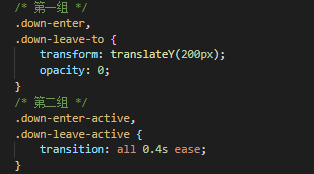




自定义前缀(默认是v，可以包裹动画元素的transition中定义name属性来实现)

用于实现一个实例实现不同动画-🡪





2>第二种方式，使用第三方库animation.css类库实现动画

第一步.下载nanimate.css。

第二引入(和正常的css引入一样link)

使用方法

直接在包裹要过渡元素的transition加上属性，只有两个属性

enter-active-class=”animated 实现的动画” leave-active-class=”animated实现的动画”

注意：一定要加上animated类名，实现的动画到animate官网找--🡪

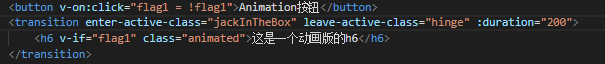


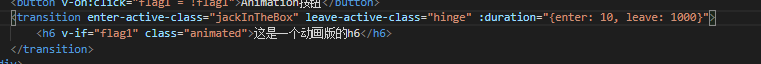
类名可以只写一遍不过得放到实现动画的元素身上-🡪



里面可以设置一个动画时间，属性为:duration

一个值时入场和出场时间一样，传一个对象时可以设置出场和入场的时间。





3>第三种方式，使用钩子函数实现动画(一般只需要实现半场动画使用如 加入到购物车)

常用是使用每场前面三个钩子函数(购物车半场动画不需要出场的动画所以不需要定义下面四个出场动画)

注意1:绑定的是事件和对应的处理函数

注意2：动画钩子函数的第一个参数:el,鸟屎要执行动画的那个DOM元素，是个原生的JS DOM对象可以认为， el是通过document.getElementById('');这样的方式获取的JS DOM对象

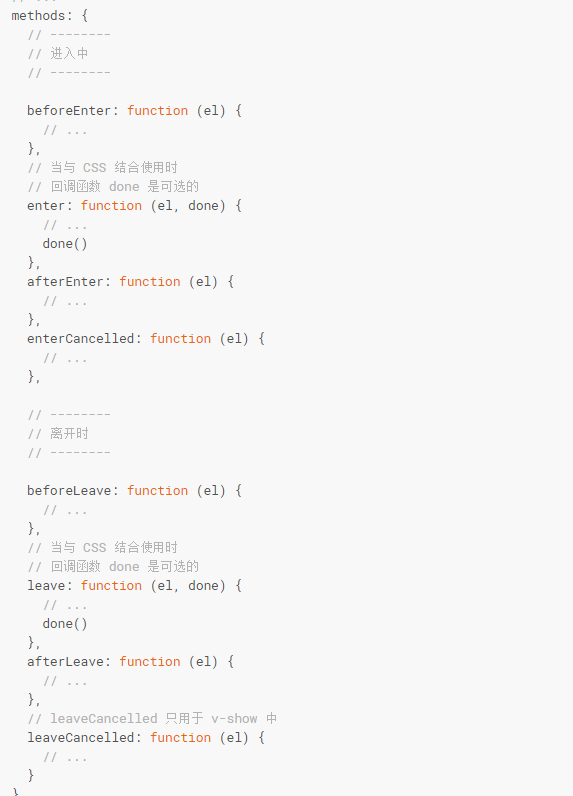
主要3：一定会回调这个函数 所以这个函数不可以不写，不然会出现延迟等问题

这里的done就是afterEnter这个函数，这个done是afterEnter函数的引用

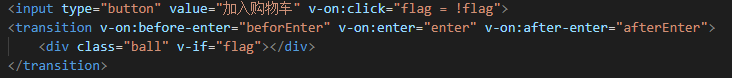
Html中🡪



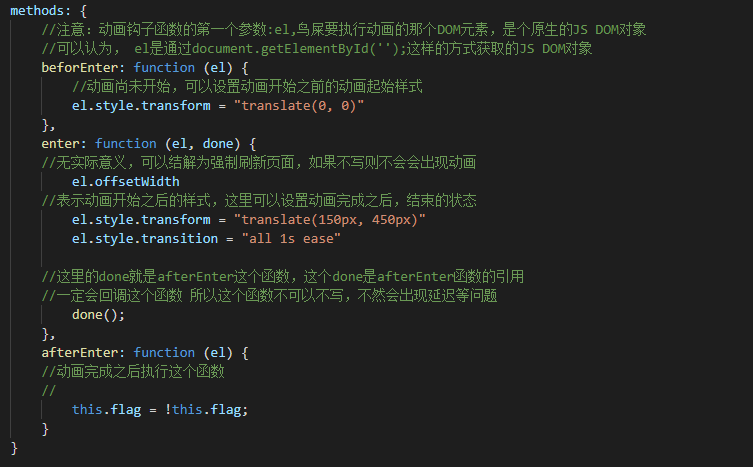
Vue实例中🡪



具体实现购物车功能html中🡪



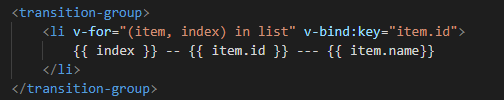
具体实现购物车的实例中🡪



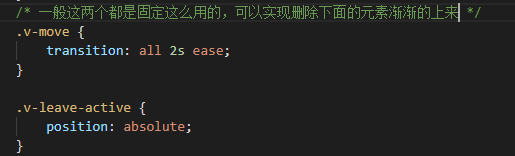
六．列表中实现动画

因为列表使用v-for渲染出来的所以需要用transition-grounp标签进行动画过渡设置

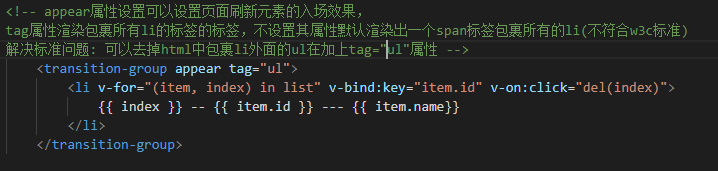
注意:实现动画一定需要列表绑定key值🡪



删除动画是需要删除的li渐渐的上来需设置🡪



apear和tag属性🡪



七．组件化和模块化的区别

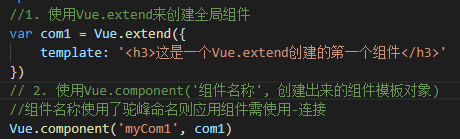
1>模块化是按代码逻辑的角度来进行划分的。方便分层开发，保证每个功能单一

2>组件化是按UI界面角度进行划分的。前端的组件化方便UI组件的重用

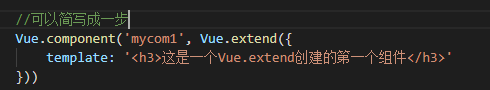
八．组件的创建的三种方式

注意：无论哪种方式创建的组件都有且只有一个根元素

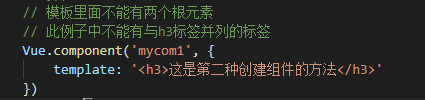
1>使用Vue.extend来创建全局组件



可以简写成一步

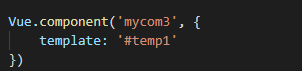


2>直接用{} 缺点：没有智能提示

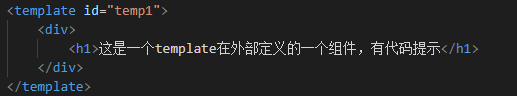


3>直接用{} 有点解决了代码提示问题

第一步



第二部，在html中



引用直接使用mycom3标签

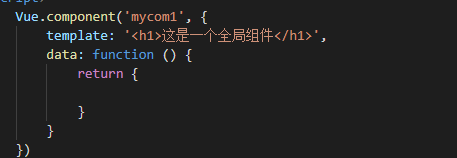
九．组件中的data

注意：

1>.组件的data必须是函数

2>组件的data必须return出一个对象

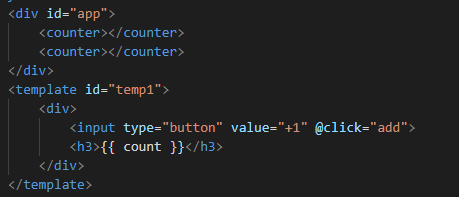
3>组件中的data数据和实例中的data数据使用方式完全一样

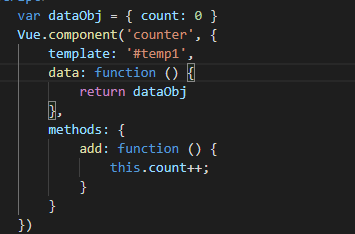


十．组件为什么一定要是一个函数切需要返回一个对象问题。

不return出一个对象组件重复使用会相互影响🡪

这样两个counter组件会相互影响

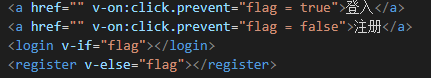




十一.不同组件之间相互切换

1. 方式1,使用v-if和v-else 缺点：只能两个组件之间进行切换

通过控制data里面的数据flag进行控制

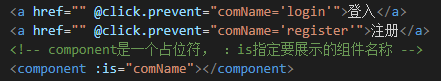


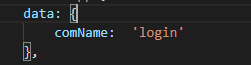
1. 方式2，通过Vue提供的component标签来切换

通过传组件名称给data自定义的comName来切换

注意:组件名称是一个字符串，属性绑定:is=”单做表达式来解析”

所以不借助comName需要这样写 :is=”’login’”





十二.组件切换设置动画

用transition标签包组件就可以实现， 和简单元素实现动画产不多。

可以设置mode属性来控制切换的模式的效果，

这里需要区分.v-move，.v-leave-active设置这两个类进行移除元素的效果

