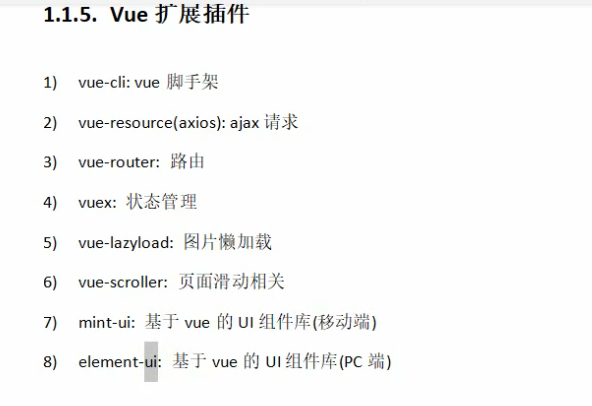
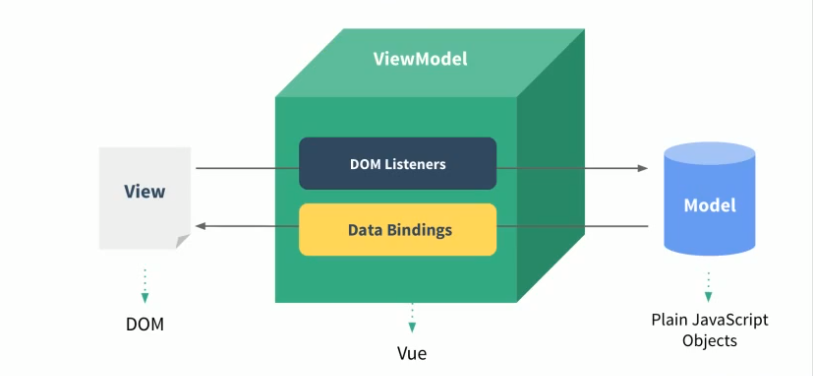
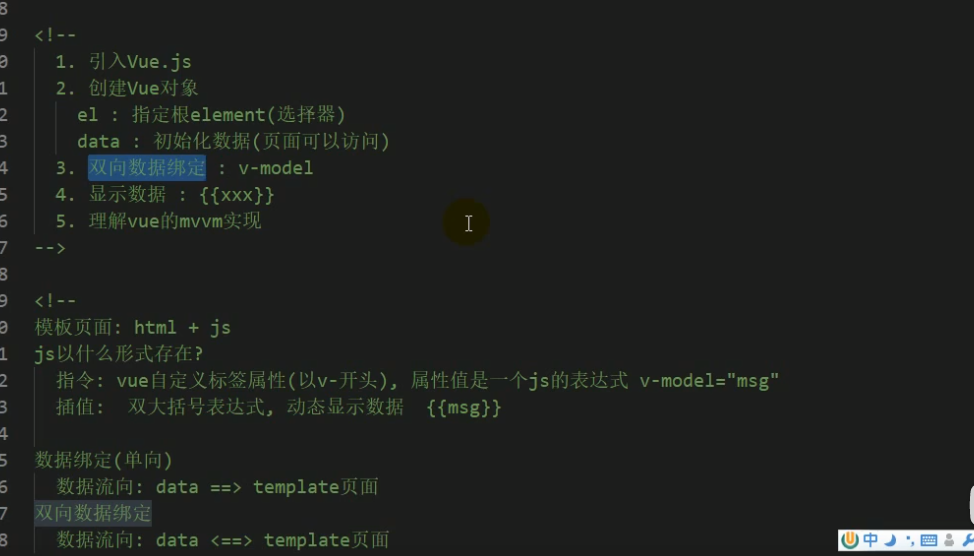
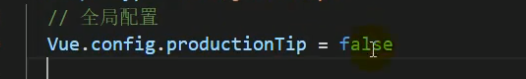
# Vue扩展插件：



# Vue：

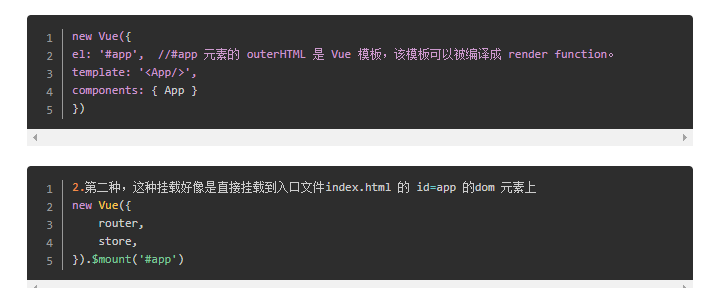


## 全局配置：



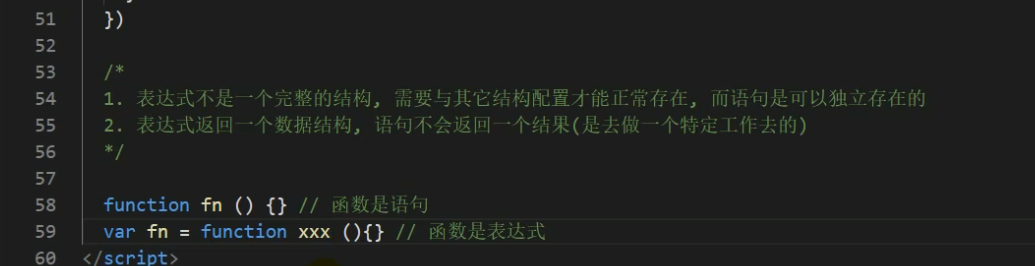
## 模板语法：

### 挂载：



### 插值（双大括号表达式）：

#### 表达式和语句的区别：

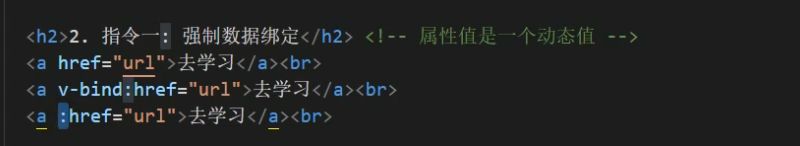


函数表达式是fn后面的function（）{}



### 指令：

#### 强制数据绑定v-bind：

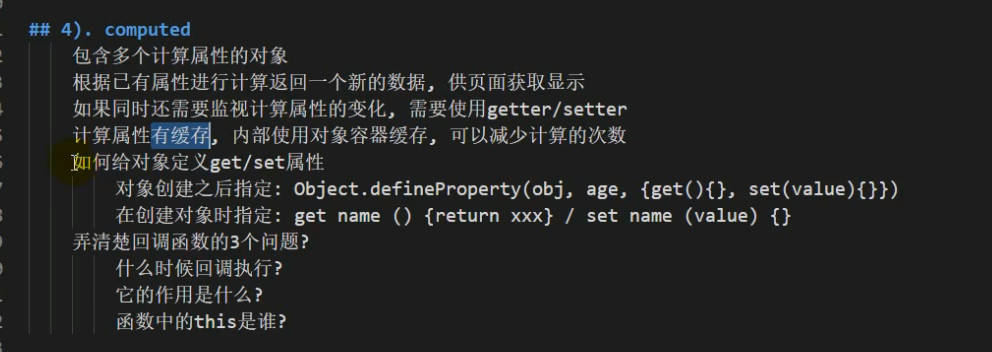


#### 绑定事件监听v-on:

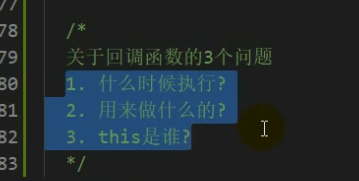


Test是回调函数

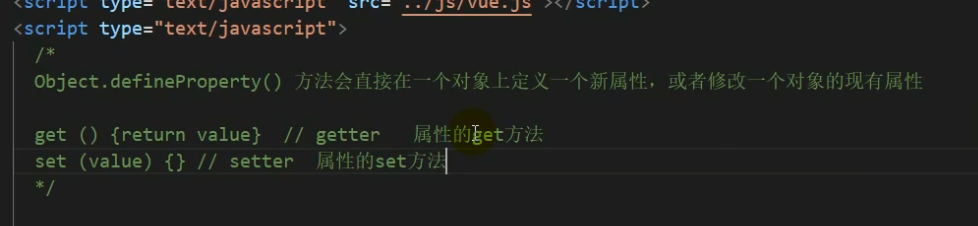
## 计算（computed）属性和监视（watcher）：

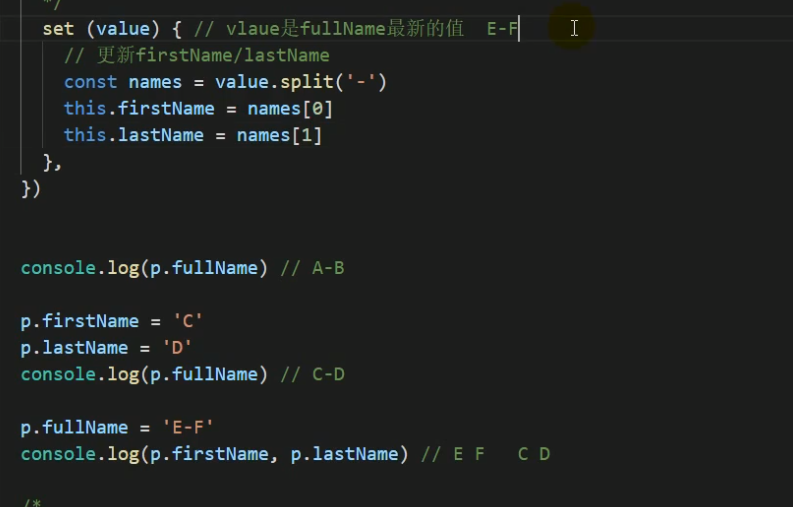
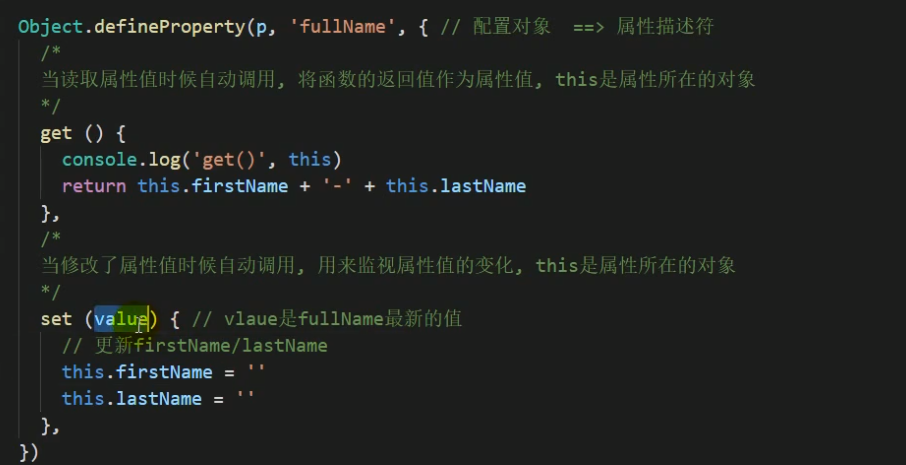
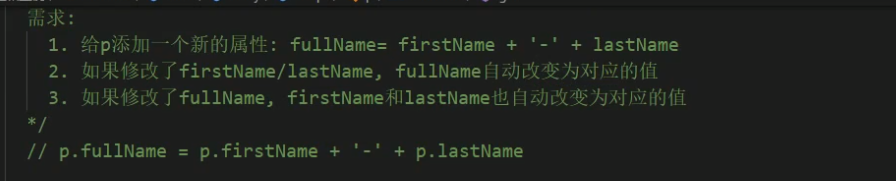


### 关于回调函数的三个问题：

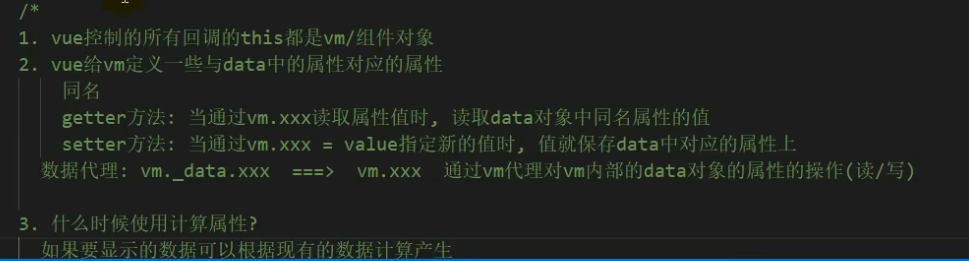


### 案例：





### computed属性：



                fullName3: {

                    get(){//监视依赖数据

                        console.log('fullName3()')

                        return this.firstName+'-'+this.lastName

                    },

                    set(value){//监视当前数据

                        const names = value.split('-');

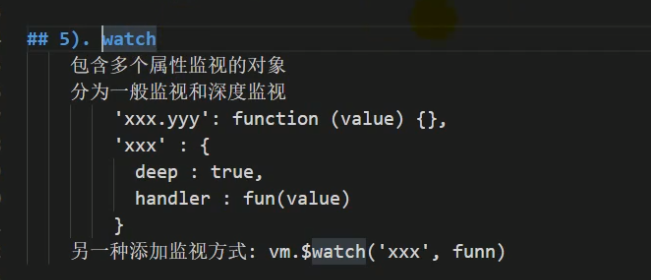
                        this.firstName = names[0];

                        this.lastName = names[1];

                    }

                }

### watch属性：



            watch: {//当fisrtName的值发生改变时调用

                firstName:function(value){  //firstName(value)也可以

                    this.fullName2 = `${value}-${this.lastName}`

                }

            },

        })

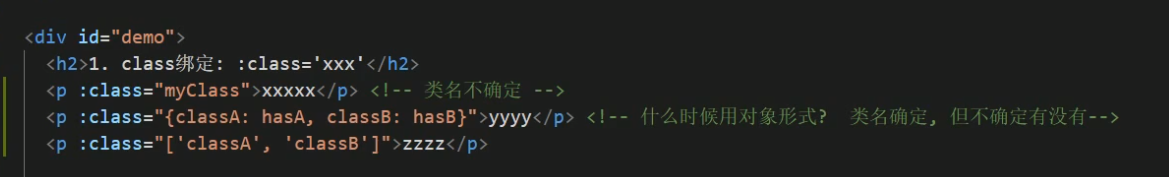
        vm.$watch('lastName',function(value){

            this.fullName2 = `${this.firstName}-${value}`

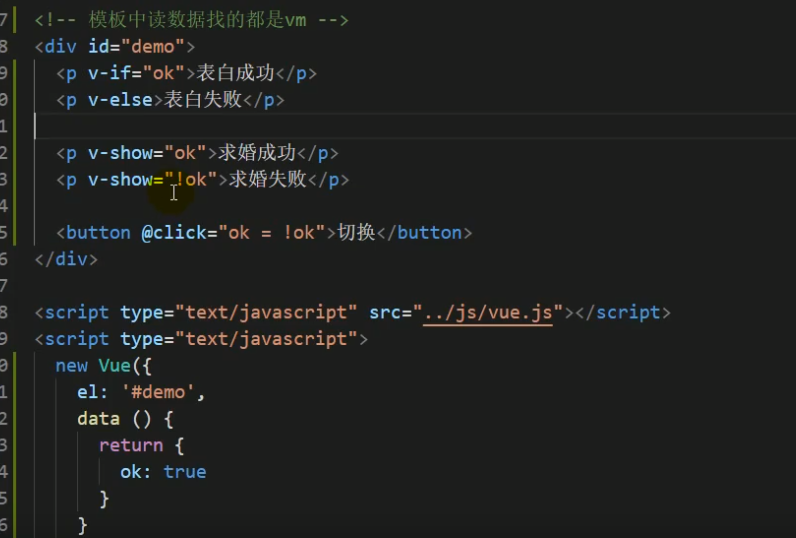
        })

## class和style绑定：





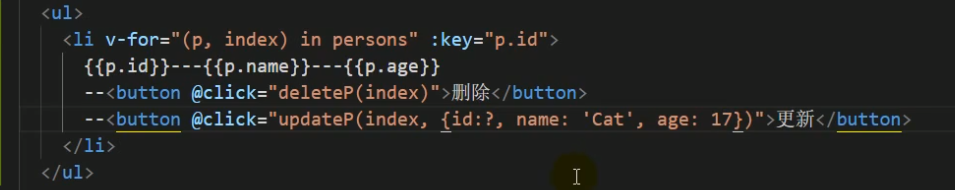
## 条件渲染 v-if v-else v-show:

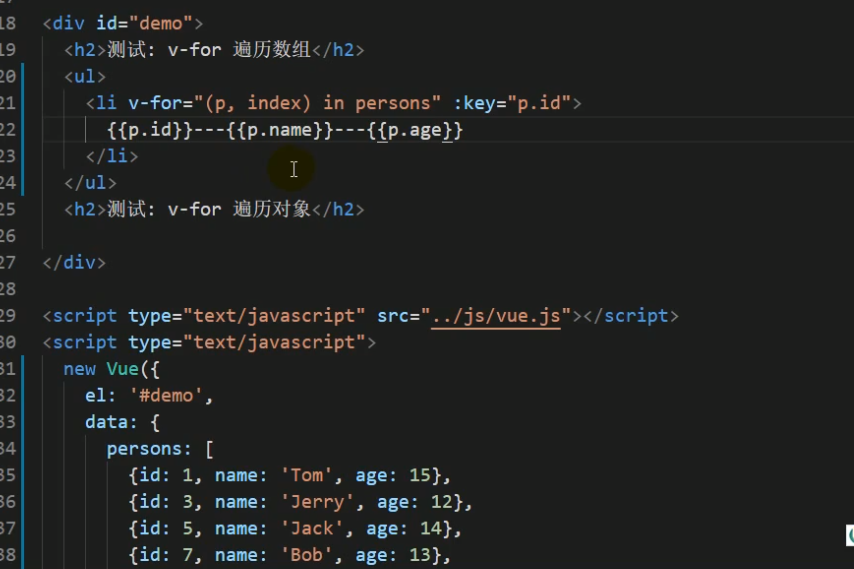


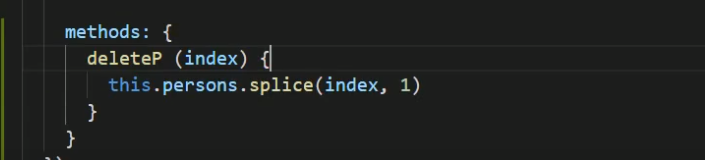
v-if v-else通过删除标签隐藏，v-show通过display：none 隐藏。

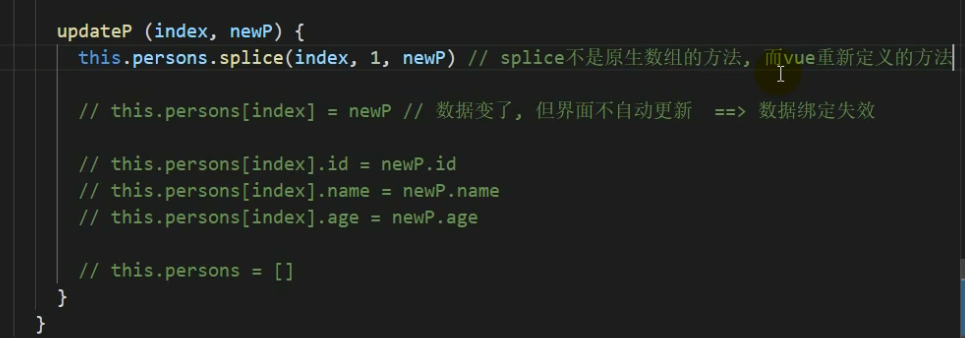


## 列表渲染 v-for：

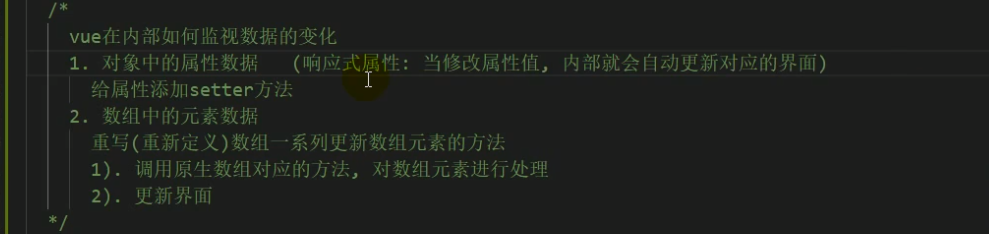


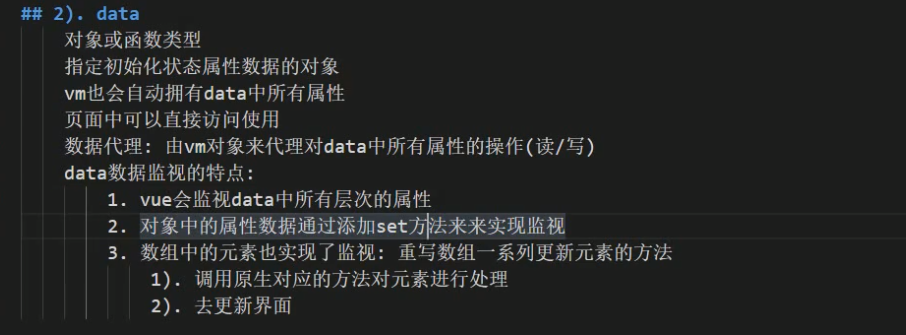






### Vm监视数据变化





### 案例：列表过滤列表排序

## 事件处理：

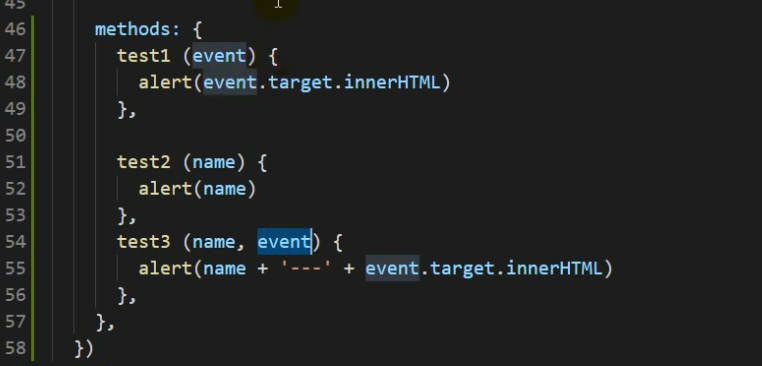
### 绑定监听：



第三条等价于

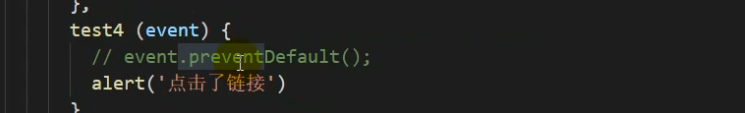


创建一个回调函数，回调函数的参数是$event



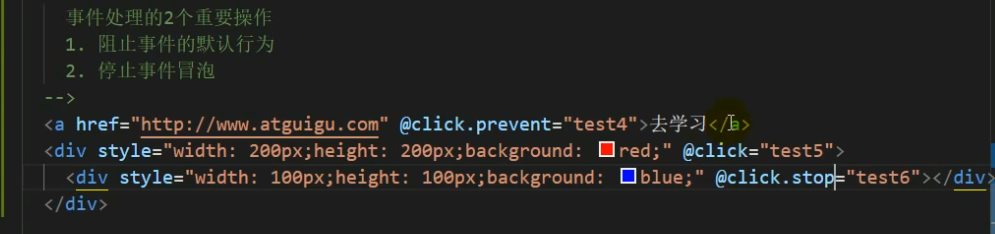
### 事件修饰符：

#### 1.阻止事件默认行为：



两者效果等价

#### 2.停止事件冒泡：

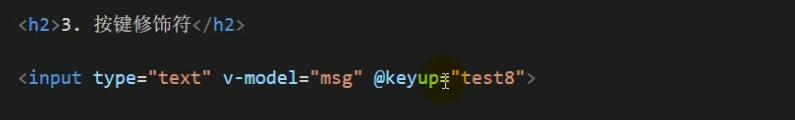


#### 其他：



事件只绑定一次

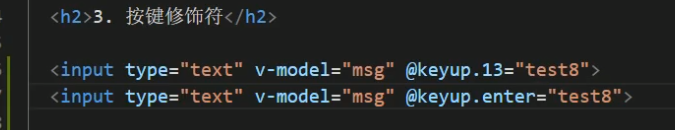
### 3.按键修饰符：



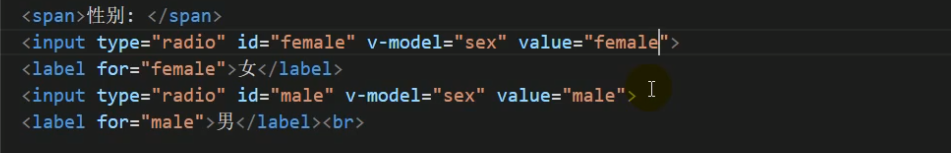
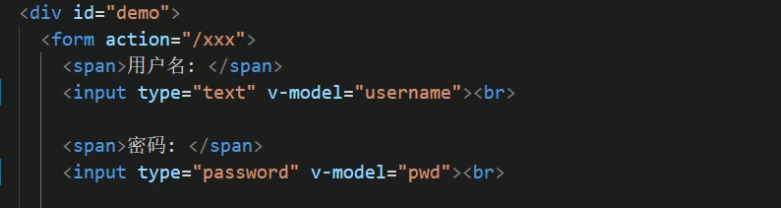
原生js：按enter键

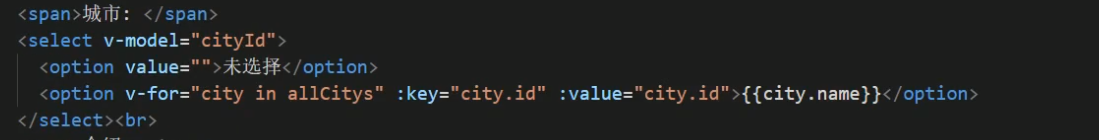


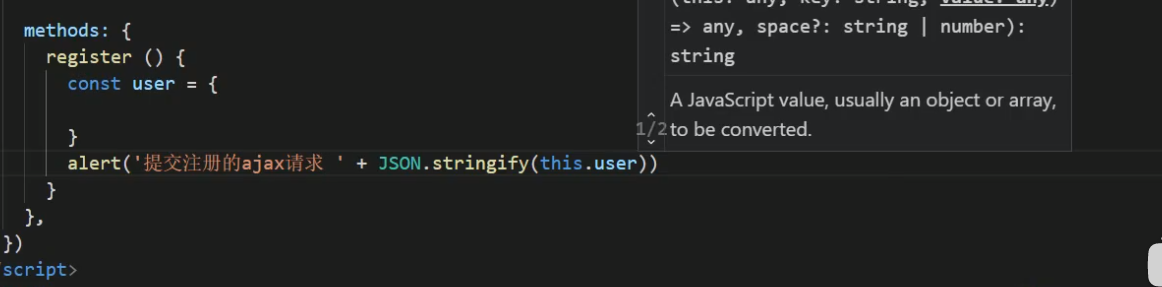
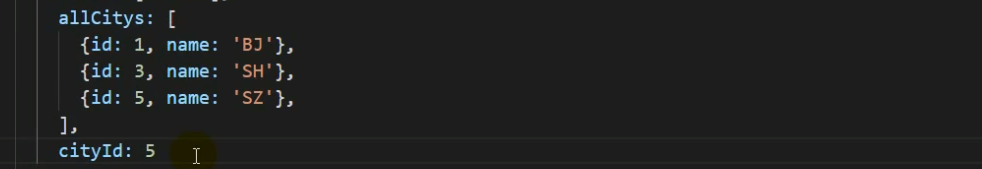
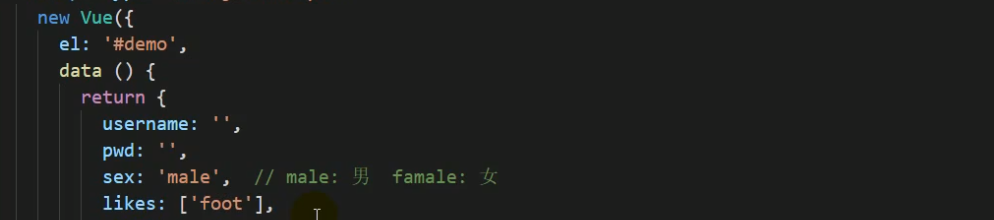
Vue：



## 表单输入绑定：







## Vue实例的生命周期：