# Vue

## 一、Vue基础

### 1.vue入门

#### 1.1Vue简介

MVVM：Model-View-View-Model，MVVM模式采用双向绑定，Model的变动，自动反映在View;View的变动，自动反映在Model。

#### 1.2入门案例

**步骤一**：导入**vue.js**。

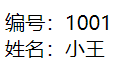
**步骤二**：创建Vue的实例对象，设置el属性和data属性。

**步骤三**：使用模板语法把数据渲染到页面上。



**例2**：



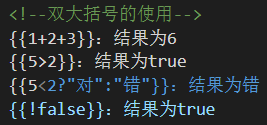


#### 21.3基础语法

el是用来设置Vue实例挂载(管理)的元素，通过css中的选择器设置vue实例管理的元素。

**Vue实例的作用范围**：Vue会管理el选项命中的元素及其内部的后代元素，即只要是在el中通过选择器选择的元素内部都可以使用{{}}，且都能获取值。

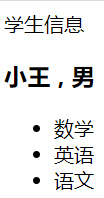
**注意{{}}只能使用在双标签中，单标签中不能使用，且不能使用在<html>、<body>标签上**。



Vue中用到的数据定义在data中。其中data属性有两种写法：







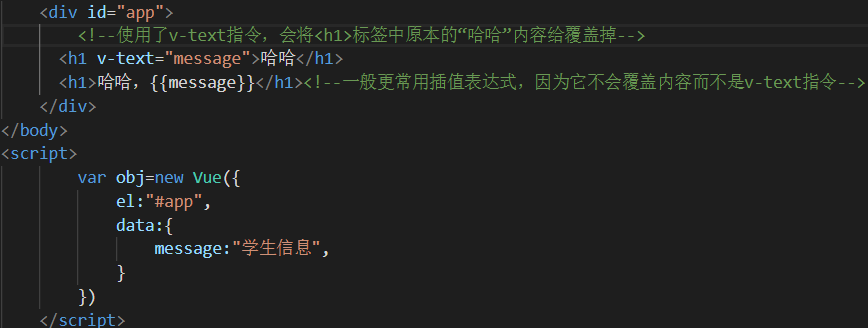
### 2.Vue指令

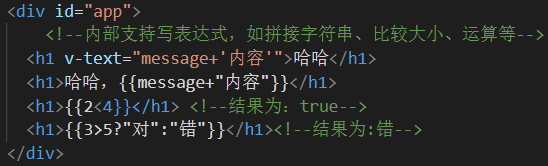
Vue指令是以v-开头的一组语法。这些指令都是作为标签的属性来使用。

#### 2.1内容绑定、事件绑定

**①v-text**

该指令用于设置标签的文本值，相当于jquery的$().text()。**注意**使用这个指令会将标签中原本有的内容完全覆盖掉。而使用插值表达式{{}}则不会。v-text指令是作为标签的属性来使用。且内部支持写表达式。





**②v-html**

该指令用来设置标签的innerHTML。相当于jquery的$().html()。

v-text指令和v-html指令的**区别**：当都只是纯文本内容时两者的效果是一样的；当内容中包含html的标签时，v-html指令会将标签解析，而v-text则是直接将标签也当作文本内容处理。**解析文本使用v-text指令，解析html结构使用v-html指令**。





**③v-on基础**

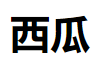
为元素绑定事件。使用语法：**v-on:事件名=”方法名”**，**v-on指令可以简写成@**，省略写法直接**@事件名=”方法名”**。这里的事件名不需要写on，比如点击事件onclick直接写成click等，**需要绑定的方法写在vue对象的methods属性中**。对于methods中定义的方法，也可以使用{{}}调用，如{{test1()}}。

事件绑定的方法写成函数调用的形式，可以传入自定义的参数，定义方法时则要定义形参来接收传入的实参。事件的后面跟上.修饰符则可以对事件进行限制，如.enter表示限制触发的按键为回车键。



通过this关键字可以在Vue对象实例中的其它属性中获取到data属性中封装的数据，同时修改的数据会同步到data中。



页面直接显示：

点击一次之后显示：

点击第二次之后显示：

例子：点击下面的加号则数量加1，减号则数量减1。











#### 2.2显示切换、属性绑定

**①v-show**

该指令是根据表达式的真假切换元素的显示和隐藏。v-show指令中的值为true时，元素会显示，为false时则元素不显示。v-show指令的值可以是结果为boolean类型的表达式。该指令的本质是修改元素的display的值来实现显示或隐藏元素。





**②v-if**

该指令是根据表达式值的真假切换元素的显示和隐藏，它操纵的是dom元素，当表达式的值为true时该dom元素存在于dom树中，为false时，从dom树中移除了，而v-show指令操纵的是元素的display属性。即**当值为false时，v-show所在的标签还存在于源码中，只不过该标签的display属性值为none，而v-if所在的标签不存在源码中，相当于将该dom元素删除了**。操作dom元素对性能消耗比较大，因此频繁的切换显示效果使用v-show更好。

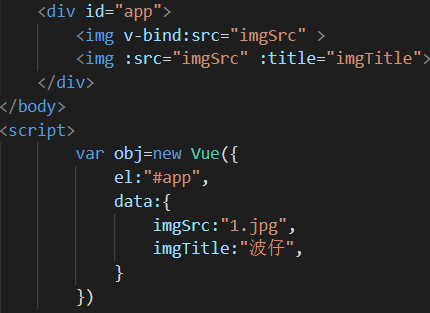


还有v-else-if、v-else指令，效果同if…else if…else语句的效果一样。



**③v-bind**

该指令用来设置元素的属性，比如src、title、class属性等。语法为：**v-bind:属性名=表达式**。可以简写成**:属性名=表达式**。需要动态的增删class可以采用三目运算符或者对象的方式，使用对象的方式更简单。



**例子：实现图片轮播**



**例2**：

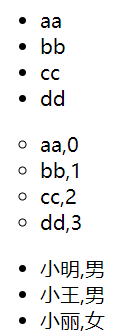


#### 2.3列表循环、表单元素绑定

**①v-for**

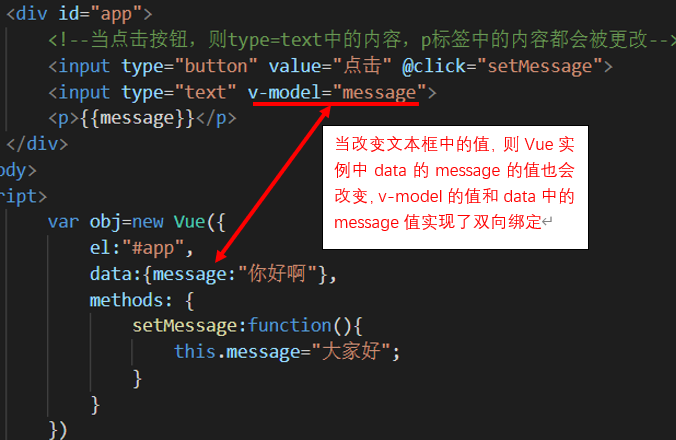
该指令根据数据生成列表结构。数组经常和v-for结合使用，**语法为**：**(item,index) in 数据**，其中item为数据中的每一项，index则为索引。

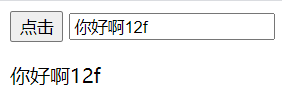




**②v-model**

该指令用来获取和设置表单元素的值(双向数据绑定)。**绑定的数据和表单元素的值相关联**。



当直接修改文本框中的值，则下面对应段落中的值也会被一起修改。

#### 2.4Vue指令总结

**v-text**：用于设置标签的文本值。

**v-html**：用于设置标签的innerHTML。

**v-on**：为元素绑定事件，用法如：v-on:click=”function”，简写@click=”function”。

**v-show**、**v-if**：根据表达式的值真假切换显示或隐藏元素。

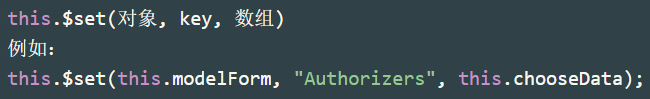
**v-bind**：用来设置元素的属性，用法如：v-bind:src=”imgSrc”，简写:src=”imgSrc”。

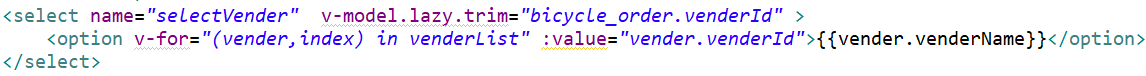
**v-for**：根据数据生成列表结构，常用于数组类型的数据。

**v-model**：用来实现表单元素的值和Vue实例的data中的数据相互绑定。

**v-model.trim**会自动过滤输入的首尾空白字符。

**给Vue实例中的对象添加数组**：不仅可以添加数组还可以添加其他类型的数据，方法都一样，只是将下面的数组替换掉。this.$set()方式给Vue中data属性的对象添加内容时，比如若对象modelForm中原本有Authorizers存在，则会被替换掉。



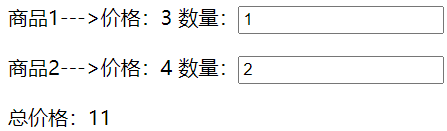


### 3.Vue的其他属性

#### 3.1computed属性

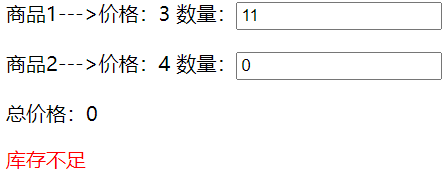
Vue的computed属性可以定义计算的方法。





**不使用watch属性实现监听效果**：





#### 3.2watch属性

watch属性是用来当作监听器的。下面这个例子的代码有点不太好，以后再改进一下，把msg拿出来，而不是放到goods对象中。



#### 3.3过滤器

使用**Vue.filter(filterName,function)**定义过滤器，其中第一个参数为定义的过滤器名称，第二个参数为函数。**注意这个过滤器必须定义在Vue实例之前，否则不起作用**。



## 二、发送异步请求

vue本身不支持发送AJAX请求，需要使用vue-resource、axios等插件实现。axios是一个基于Promise的HTTP请求客户端，用来发送请求，也是vue2.0官方推荐的，同时不再对vue-resource进行更新和维护。

### 1.使用vue-resource插件



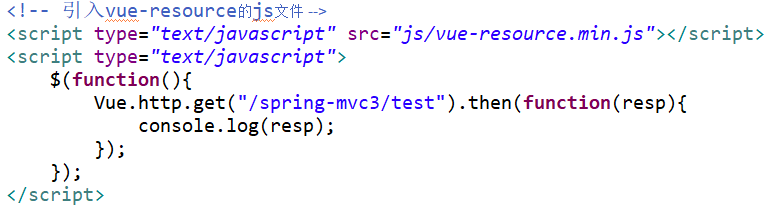
引入vue-resource后，可以基于全局的Vue对象使用http，也可以基于某个Vue实例使用http。

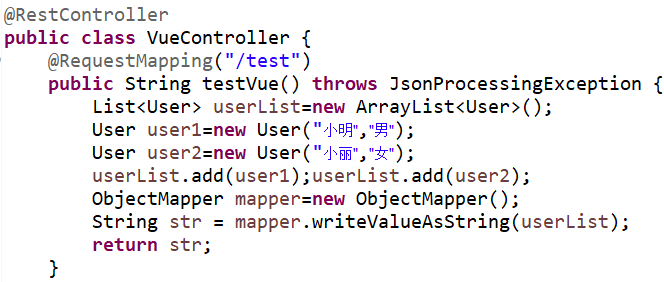
**①基于全局的Vue对象使用Http**：

Vue.http.get(url,其他参数).then(successCallback,errorCallback);

Vue.http.post(url,其他参数).then(successCallback,errorCallback);

在发送请求后，使用then方法来处理响应结果，then方法有两个参数，第一个参数是响应成功时的回调函数，第二个参数是响应失败时的回调函数。



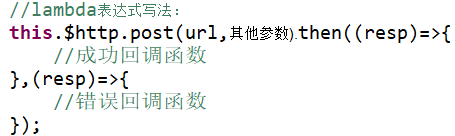


**②在一个Vue实例内使用$http**：

this.$http.get(url,其他参数).then(successCallback, errorCallback);

this.$http.post(url,其他参数).then(successCallback, errorCallback);

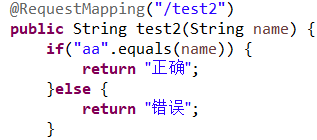
then方法的回调函数也有两种写法，第一种是传统的函数写法，第二种是更为简洁的ES 6的Lambda写法。



如果Web服务器无法处理编码为application/json的请求，可以启用emulateJSON选项。启用该选项后，请求会以application/x-www-form-urlencoded作为MIME type，就像普通的HTML表单一样。注意设置emulateJSON的值为true时，使用的是{emulateJSON:true}，是用冒号，而不是用等号，否则会报Uncaught SyntaxError: Invalid shorthand property initializer。

**对于get请求**，需要将发送给后端的数据封装到params对象中，且发送的数据形式为key-value形式，如{params:{“name”:“小明”}}，这样就会把数据以key-value形式拼接的路径后面。

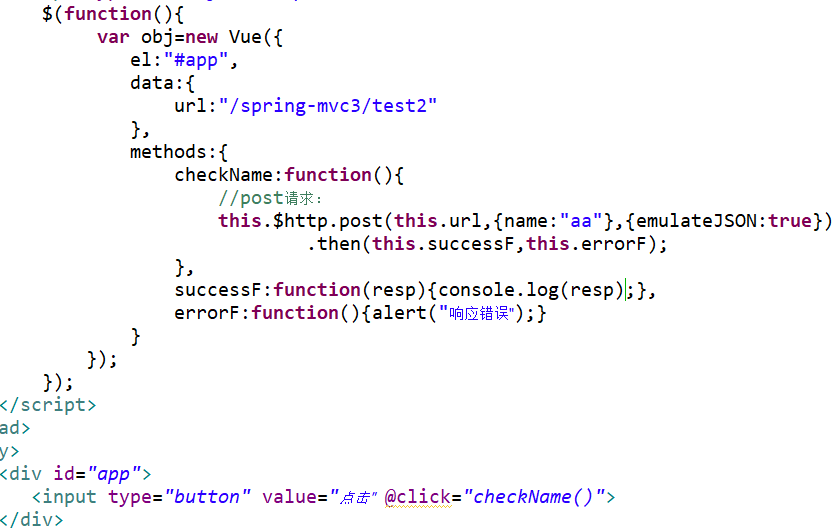
**对于post请求**，要发送给后端的数据则直接以{key,value}的形式发送即可，不需要封装到params对象中。若 Web服务器无法处理编码为application/json的请求，则需要设置{emulateJSON:true}。



**get请求**：



**post请求**：



**Lambda表达式的写法**：



### 2.使用axios插件

有空再补充。

## 三、Vue的声明周期

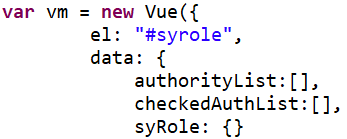
老师讲过，还没学。

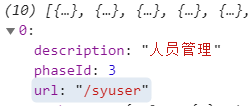
## 四、Vue的具体应用

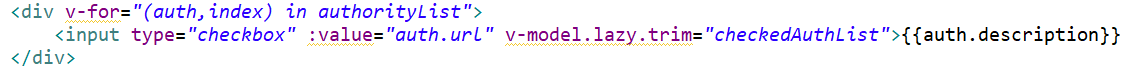
### 例1：v-for动态生成复选框

**使用v-for动态生成复选框，且使用v-model动态绑定数据**。

authorityList是一个集合，该集合中存放的每个元素都是一个对象，这个对象包含如下的属性：description、phaseId、url。采用v-for遍历authorityList集合来动态生成复选框。复选框的value属性值为集合中的对象的url属性，文本内容则是对象的description属性的值。使用v-model指令实现勾选复选框时动态绑定数据到checkedAuthList集合中。注意要想实现选中复选框的内容就动态绑定数据，则复选框的value属性值必须设置，同时必须绑定到一个集合中，不是集合则不能绑定，绑定到集合中的数据是复选框的value属性值。





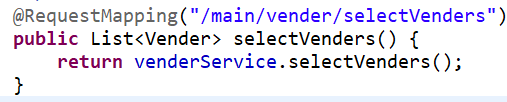


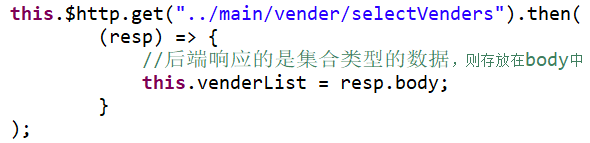
在将checkAuthList集合中的数据添加到syRole对象中：

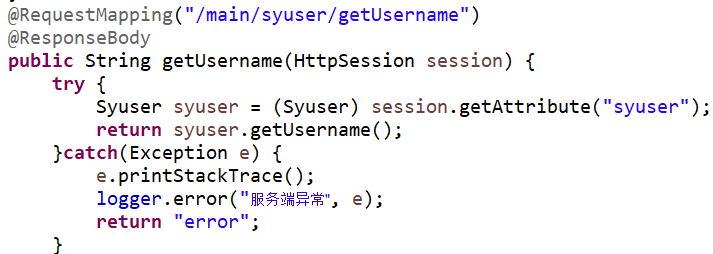


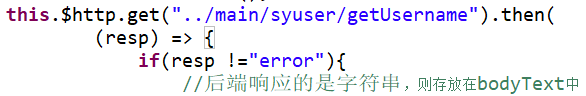
### 例2：接收后端不同类型的数据

**在Vue中发送ajax请求时，针对后端响应的不同内容，采用不同的方式接收数据**：**当后端响应的是集合类型的数据，则ajax中直接用resp.body接收，若响应的是字符串，则用resp.bodyText接收**。





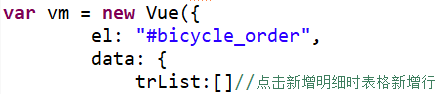




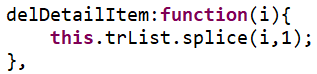
### 例3：v-for动态生成表格中的行

使用Vue的v-for指令动态生成表格的行，以及删除表格的行。需要在Vue的实例中添加一个数组。为按钮绑定添加和删除事件，添加行就是往数组中push一个值，数组中有几个值就会循环产生几行，即数组长度为行的个数。删除事件则删除数组中指定的内容即可以。

下面这个代码实现的效果，在删除时，每次都只能删除尾行，有点问题，因为每次删除时index都会变。



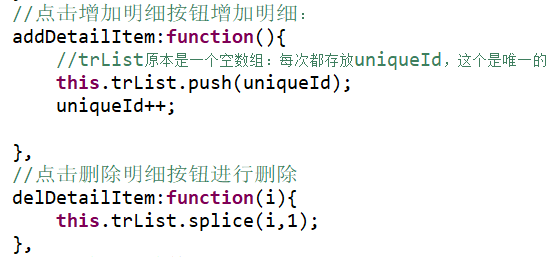






解决办法，每行都使用一个唯一的固定的id，这样删除时就可以解决上面的问题：**在使用v-for的标签中使用key属性绑定唯一的id，同时在删除按钮的方法中传入的参数为index**。







### **例4**：

**采用post方式提交的数据需要序列化**。采用post的方式，从前端向后端发送对象时，在后端的Controller方法中可以采用对象直接映射接收，但必须保证前端传送的对象名和后端的对象参数名称一致，且后端的该model对象必须实现序列化，否则接收不到值，都为null。





