Vue学习笔记

### MVVM简介

MVVM是前端的一种思想,其中vm起调度作用,第一个M可以理解为保存数据的,V可以理解为正常的html页面,如果页面要填充数据,则V通过VM层向M层拿数据,然后渲染(填充)到页面,如果要V层数据改变,则也是通过VM进行向M层保存,不过这种模式的优点就是,如果V或M层的某一方数据改变了,则M或V层都能感知到,这种机制成为数据的双向绑定,VUE就支持这种机制。

### Vue简介

Vue，是一个前台的框架，这个框架只需要开发者将注意力放在业务逻辑上，不需要关心如何将数据渲染到页面，并且该框架改变了Jquery通过操作dom来进行js的动态变化。

### 常用的指令

⑴为标签绑定属性(只能从m绑定到v)

v-bind：属性名称 绑定属性，缩写为:

⑵为标签绑定事件

v-on: 事件名称 绑定事件，缩写为@

⑶为标签绑定文本

msg为vue对象中data对象下的一个属性

{{msg}}：例如<p>---{{msg}}+++</p>，可以显示msg和p中的所有字符串

v-text=”msg”: msg对应的字符串原封不动的赋值给指定标签

v-html=”msg”:该指令可以将msg中含有的html通过页面显示出来

v-bloak=”msg”:msg的信息完全加载出来的时候，才会渲染到页面（{{msg}}会先显示{{msg}}，然后等加载完成再显示msg的真实内容）

⑷v-model(数据的双向绑定,只能运用到表单中)

Eg:<input type=”text” v-model=”msg”/>

⑸v-for指令

I.迭代数组

<p v-for=”(item,index) in arr”>{{item}}</p>

item:元素值

index：索引值

II.迭代对象中的属性

<div v-for=(val,key,index) in userInfo></div>

val:属性值

key：属性名称

index：在数组中的索引

III.迭代数字

<p v-for=”i in 10”>{{i}}</p>

i从1开始，到10结束

当用v-for更新已经渲染过的元素列表时，他默认就地复用策略，如果数据项的顺序改

变，Vue不是通过移动dom元素来匹配数据项的顺序，而是简单的复用此处每个元素，并

且保证他在特定索引下显示已被渲染过的每个元素

⑹v-if和v-show

两个指令都可以将其控制的区域隐藏，但是略微有些不同。

v-if 的特点：每次都会重新删除或创建元素

v-show 的特点： 每次不会重新进行DOM的删除和创建操作，只是切换了元素的

display:none 样式

### 事件修饰符

用法：<div @click.修饰符名称=”show”></div> or <div @click.修饰符名称></div>

⑴stop

阻止冒泡和事件的传播（冒泡：比如div中包含一个btn,给两个元素都定义事件，则点击btn则会触发div的点击事件的执行）（执行顺序：从最里层到最外层）

⑵prevent

阻止默认事件的发生

⑶once

只触发一次

⑷self

只用触发当前元素时才会执行事件函数

⑸capture

捕获，（从最外层到最里层）

### Vue中使用class样式

⑴数组

Eg：<h1 :class=”[‘red’,’thin’]”>111</h1>

⑵数组中使用三元表达式

Eg：<h1 :class=”[‘red’,’thin’,flag?active:’’]”>11</h1>

如果data中flag为true，则绑定属性active，否则不绑定

⑶数组中嵌套对象(使用对象代替三元表达式，提高可读性)

Eg；<h1 :class=”[‘red’,’thin’,{‘active’:flag}]”>11</h1>

如果data中flag为true，则绑定属性active，否则不绑定

⑷直接使用对象

Eg：<h1 :class=”{red:flag1,thin:flag2}”>11</h1>

对象的属性是样式上类名称，值是data中的boolean的值

⑸使用内联样式

Eg：

写法一：直接绑定在元素上(包含其他字符，要加双引号)

<h1 :style=”{color:’red’,”font-size”:’12px’}”>111</h1>

写法二：将样式放在data中

<h1 :style=”styleObj”>11</h1>

data:{

styleObj: {color:’red’,”font-size”:’12px’}

}

写法三：直接绑定多个样式对象(标签中写数组)

<h1 :style=”[styleObj1,styleObj2]”>111</h1>

data:{

styleObj1:{style….},

styleObj2:{style…}

}

### Vue中的过滤器

作用: 可被用作一些常见的文本格式化

过滤器可以用在两个地方：mustache 插值{{msg}}和 v-bind 表达式

使用步骤：

创建过滤器，并将过滤器和vue对象绑定。

Vue.filter('dataFormat',function(date){

var year=date.getFullYear();

var month=date.getMonth()+1;

var days=date.getDay();

return year+"-"+month+"-"+days;

});

使用：(将date的值传给后边的函数做处理，后边的函数也可以带参数，可以多个过滤器连用)

<p>{{date|dataFormat()}}</p>

字符串补全位数：padStart(最大长度，用什么字符补齐); （头部补齐）

padend(最大长度，用什么字符补齐); (尾部补齐)

### Vue生命周期函数（钩子函数）

在vue生命周期每个阶段都让开发者插入自己的代码。

创建时期：

beforeCreate()：表示实例完全被创建出来之前，会执行它。data 和 methods 中的 数据都还没有没初始化

created():在 created 中，data 和 methods 都已经被初始化好了！

beforeMount():表示 模板已经在内存中编辑完成了，但是尚未把模板渲染到页面中,换言之，在 beforeMount 执行的时候，页面中的元素，还没有被真正替换过来，只是之前写的一些模板字符串

mounted()：表示，内存中的模板，已经真实的挂载到了页面中，用户已经可以看到渲

染好的页面了。当执行完 mounted 就表示，实例已经被完全创建好了

运行时期：（data中的数据改变时触发）

beforeUpdate()：这时候，表示 我们的界面还没有被更新，数据肯定被更新了

updated()：updated 事件执行的时候，页面和 data 数据已经保持同步了，都是最新的

对象销毁：

beforeDestory():data，methods，指令，过滤器还没被销毁，

destroyed():已经被完全销毁，不能使用。

### 发送get，post请求

引入js：<script src="./lib/vue-resource-1.3.4.js"></script>

this.$http.get('/someUrl', [options]).then(successCallback, errorCallback);

this.$http.post('/someUrl', [body], [options]).then(successCallback, errorCallback);

数据使用返回值的body属性获取。

### 组件

组件创建方式：

使用 Vue.extend 来创建全局的Vue组件：

var com1 = {

template: '<h3>这是使用 Vue.extend 创建的组件</h3>'

// 通过 template 属性，指定了组件要展示的HTML结构

};

使用 Vue.component('组件的名称', 创建出来的组件模板对象)：

Vue.component('myCom1', {

template:’’,

data:{},

methods:{}

}) //注册组件

引入到页面中：(使用注册组件的名称引入)

如果使用 Vue.component 定义全局组件的时候，组件名称使用了 驼峰命名，则在引用组件的时候，需要把 大写的驼峰改为小写的字母，同时，两个单词之前，使用 - 链接；

如果不使用驼峰,则直接拿名称来使用即可

<mycom1></mycom1>

通过template标签也可以创建标签

Vue.component('mycom3', {

template: '#tmpl2'

}),

在html中：

<template id="tmpl2">

<h1>这是私有的 login 组件</h1>

</template>

引用方法相同。

也可以定义私有化组件：

var vm=new Vue({

el:’#app’,

data:{},

methods:{},

components:{

login: {

template: '#tmpl2'

}

}

});

组件的data必须是一个方法，必须返回一个对象。

### 组件切换方式

通过v-if标签和v-else标签指向同一个boolean值。

<login v-show="flag"></login>

<register v-show="!flag"></register>

使用vue提供的标签,绑定is属性，该属性指定要显示那个组件：

<a href="" @click.prevent='con="login"'>登录<a><hr/>

<a href="" @click.prevent='con="register"'>注册<a>

<component :is="con"></component>（con为data中定义的一个变量）

### 