

Vue第一天

今日内容介绍

- Vue介绍
- MVVM模式介绍
- Vue开发流程
- Vue常用系统指令
- 实现品牌管理案例
- Vue调试工具

今日内容学习目标

- 了解Vue的版本和常用学习资源
- 理解MVVM模式要解决的问题
- 能够分清楚MMVM中的M，V，VM在Vue的编码结构中是对应哪些代码块的
- 能够记住Vue的3个编码步骤
- 知道使用v-的相关的指令，例如v-text,v-html,v-model,v-for,v-on等
- 理解并记住Vue对象中的 el,data,methods三个属性的作用
- 能够利用vue的知识实现品牌管理的增，删除，列表展示
- 能够利用filterBy过滤器实现数据过滤
- 知道利用Vue-DevTools调试工具监控Vue运行时数据

详细内容

1.1 课程介绍

1.1.1 时间安排

- 总课时：8天
 - Vue基础 3.5天
 - Vue项目 4.5天

1.1.2 课程的特点和目的

在公司的项目开发中常常为了开发效率和可扩展性考虑，为了方便在后期维护软件，降低维护成本，公司常常会选择业内已经成熟的框架来架构自己的产品，其中Vue作为一个国产的前端开发框架，拥有完整的中文文档和学习门槛低而被越来越多的公司青睐，因此作为前端开发人员要想能够更快的看懂公司框架更好的融入到公司业务中，也必须学习Vue。

这套Vue项目课程从设计上是以常见开发需求作为依托，学习完毕以后，使同学们能够感受到公司通常开发的功能是什么样的，该如何去和后台配合。整个课程在教学中是由浅入深，从0到1带领学生一步步的将项目功能做出来，主要特点有：

- 首先会带领同学们学习Vue的基础知识
- 其次带领大家从0到1搭建一个项目框架，使学生拥有项目的架构能力

- 最后带领大家在自己搭建好的项目框架上实现一个个项目功能，对前面知识的一个综合运用，整个过程中让同学们感受一个完整的项目是如何产生的

1.2 项目截图

- 首页



- 新闻资讯

[< 返回](#)

图文资讯



1季度多家房企利润跌幅超50% 去库存促销战打响

发表时间：2015-04-16

点击：1



买房还是炒股，2015年买房无法拒绝的5大理由

发表时间：2015-04-16

点击：2



抢先实拍猎豹CS10 霸气时尚2.0T涡轮增压

发表时间：2016-12-07

点击：1



长城汽车不走高端很快就会死 哈弗H8卷土重来

发表时间：2015-04-16

点击：0



三星S6造价更高 卖得却比iPhone 6便宜

发表时间：2015-04-16

点击：0



乐视手机遭疯狂吐槽 忽不忽悠看了再说

发表时间：2015-04-16

点击：0



假宽带频现 什么是真正的光纤宽带？

发表时间：2015-04-16

点击：0



58同城与赶集网最快今日将宣布合并

发表时间：2015-04-16

点击：0



首页



会员



购物车

211



搜索

[< 返回](#)

资讯详情

1季度多家房企利润跌幅超50% 去库存促销战打响

2015-04-16 1次浏览 分类：经济民生

房企一季度销售业绩已经陆续公布，克而瑞研究中心统计数据显示，今年一季度，TOP20的房企仅6家实现业绩同比增长。

多家企业销售下滑

记者了解到，虽然恒大、融创一季度交出了不错的答卷，但是，万科等排名靠前的房企均出现销售下滑。其中万科公布前三个月销售金额为460.6亿元，同比去年的542.3亿元减少15%，而碧桂园公布，截至2015年3月31日止三个月，集团共实现合同销售金额约177.3亿元，同比下跌44.32%。

此外，一季度多家企业的利润跌幅超过50%。数据显示，万科预计2015年1月至3月归属于上市公司股东的净利润为6亿~7亿元，比上年同期下降54%至61%。

趋势

地价占比上升及去库存压力下 房企利润回归

中原地产首席分析师张大伟称，一季度房企销售及利润普遍下滑，主要是因为企业销售节奏的变化，“大部分企业在2013年完成情况非常好，预留了大量签约结转至2014年。但2014年整体企业完成情况较差，大部分企业反而透支了



首页



会员



购物车

211



搜索

- 图片分享



• 现代简约 明亮的外表卧室卧室背景墙、吊顶黄色

不要简朴不要素雅洋气卧室我做主，高低床榻榻米式靓丽卧室装扮，现代油画壁画帆布画卧室餐厅仟象映画，现代中式



首页



会员



购物车

211



搜索

< 返回

图片详情

现代简约 明亮的外表卧室卧室背景墙、吊顶黄色

2015-04-18 0 次浏览 分类：经济民生





不要简朴不要素雅洋气卧室我做主，高低床榻榻米式靓丽卧室装扮，现代油画壁画帆布画卧室餐厅仟象映画，现代中式卧室装修效果图欣赏，温馨浪漫，而且很时尚的卧室设计，欧式卧室飘窗装修效果图。



首页



会员



购物车

211



搜索

2/5



- 商品购买



华为 (HUAWEI) 荣耀6Plus
16G双4G版

¥2195 ~~¥2499~~

热卖中

剩60件



苹果Apple iPhone 6 Plus 16G
4G手机 (联通三网版)

¥5780 ~~¥6388~~

热卖中

剩200件



小米 (Mi) 小米Note 16G双网
通版

¥2199 ~~¥2699~~



佳能 (Canon) EOS 700D 单
反套机

¥4799 ~~¥5099~~



首页



会员



购物车

211



搜索



苹果Apple iPhone 6 Plus 16G 4G手机
(联通三网版)

市场价: ¥6388 销售价: **¥5780**

购买数量:

-	1	+
---	---	---

立即购买

加入购物车

商品参数

商品货号: SD7159810321


首页


会员

 **211**
购物车


搜索

提交评论

[返回](#)

请输入评论内容...

发表

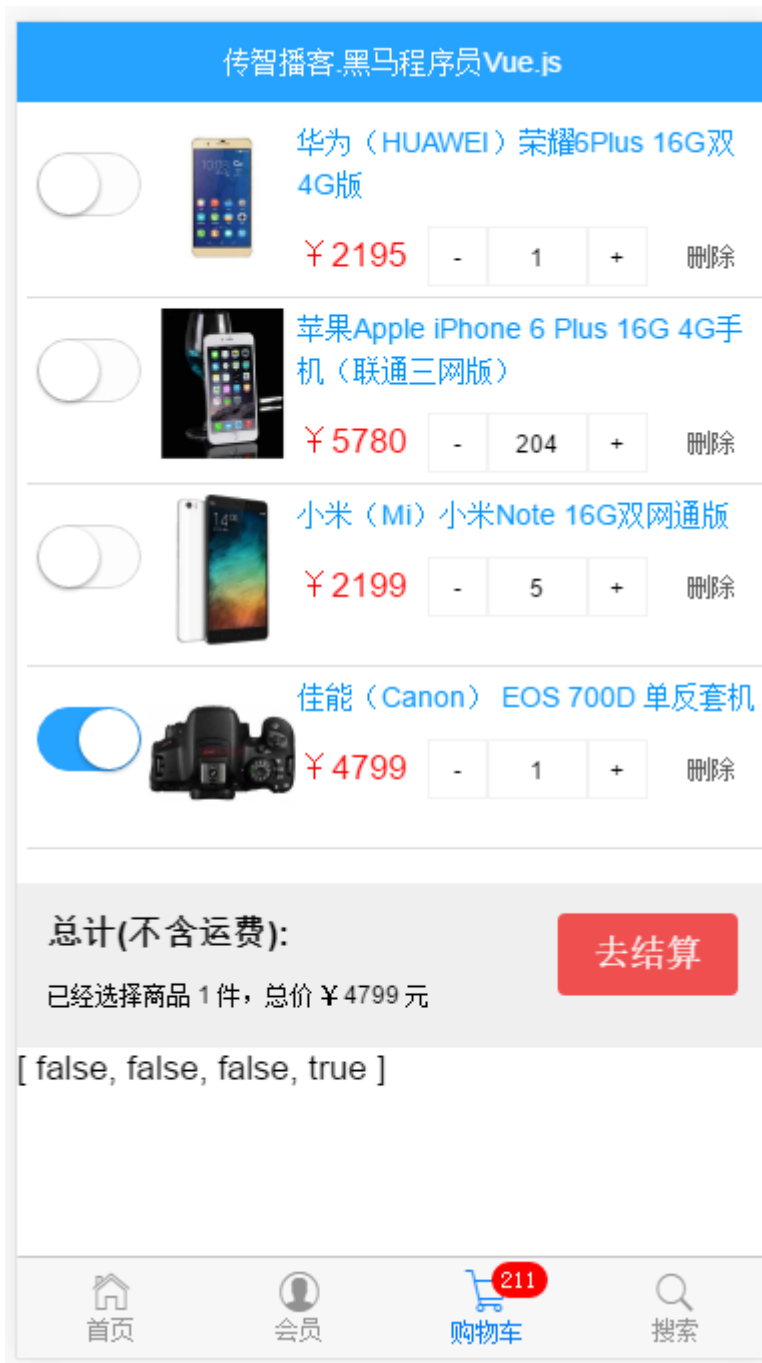
评论列表

44条评论

加载更多



- 购物车



1.3 Vue前置知识

1.3.1 Vue学习资源

- Vue官网: <https://cn.vuejs.org/>
- Vue GitHub: <https://github.com/vuejs/vue>
- Vue1.0 在线文档: <http://v1-cn.vuejs.org/guide/>
- Vue2.0 在线文档: <https://cn.vuejs.org/v2/guide/>
- Vue1下载地址: <http://v1-cn.vuejs.org/js/vue.js>
- Vue2下载地址: <https://unpkg.com/vue@2.2.1/dist/vue.js>

1.3.2 Vue使用现状

- Vue项目：饿了么招聘：<https://jobs-mobile.ele.me/#/>
- Vue 1.0.26开发的移动版项目：<https://www.uhouzz.com/>
- Vue.js v1.0.28开发的移动版项目：<http://m.jianguanoa.com/#!/index>（开启chrome调试工具中的移动版浏览）
- Vue.js v1.0.28开发的移动版项目：<http://www.fangdd.com/group/shanghai?zhihu>
- Vue开发的PC版本项目：<http://lushi.zhujunwu.cn/#!/builds>
- 知乎上的一个专帖：[国内有哪些公司在用Vue.js，有什么心得](#)

1.4 开始课程

1.4.1 MVVM模式

- MVVM拆分解释为：
 - Model:负责数据存储
 - View:负责页面展示
 - View Model:负责业务逻辑处理（比如Ajax请求等），对数据进行加工后交给视图展示
- MVVM要解决的问题是将业务逻辑代码与视图代码进行完全分离，使各自的职责更加清晰，后期代码维护更加简单
- 用图解的形式分析Ajax请求回来数据后直接操作Dom来达到视图的更新的缺点，以及使用MVVM模式是如何来解决这个缺点的
- Vue中的 MVVM

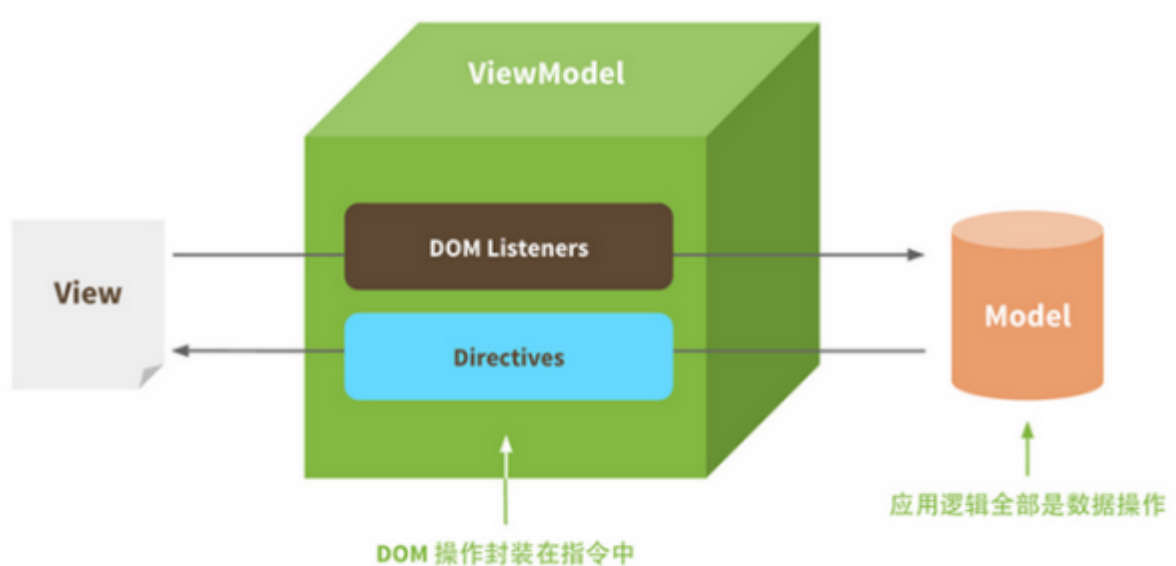


图1 Vue.js的MVVM架构

1.4.2 Vue初体验（记住Vue的编写步骤）

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <!-- 导入vue文件 (1.0) -->
  <script src="vue1028.js"></script>
</head>
<body>
  <!-- 2.0 一定要确定一个范围来表示此范围中的所有指令都是被vue来解析的 -->
  <div id="app">
    {{ msg }} <!-- 插值表达式：纯粹是展示一些字符串 -->
  </div>

  <script>
// 3.0 通过vue实例化一个vue对象,用这个对象来解析id=app中的所有指令
// vm对象就是MVVM中的vm这个角色
var vm = new Vue({
  el: '#app',    // 表示vm对象将来操作的view的区域是 app的这个div中的所有指令
  data: {
    msg: 'hello vue'    //称之为MVVM中的 Model
  }
});
  </script>
</body>
</html>

```

1.4.3 Vue常用系统指令

- 插值表达式{{}}:

- 数据绑定最常见的形式就是使用 “Mustache” 语法（双大括号）的文本插值
例如: `Message: {{ msg }}`
Mustache 标签将会被替代为对应数据对象上 `msg` 属性（`msg`定义在`data`对象中）的值。
无论何时，绑定的数据对象上 `msg` 属性发生了改变，插值处的内容都会更新。

{{}}对JavaScript 表达式支持，例如：

```
{{ number + 1 }}
```

```
{{ ok ? 'YES' : 'NO' }}
```

```
{{ message.split('').reverse().join('') }}
```

但是有个限制就是，每个绑定都只能包含单个表达式，如下表达式无效：

```
<!-- 这是语句，不是表达式 -->
{{ var a = 1 }}
```

```
{{ if (ok) { return message } }}
```

```
<!-- 流控制也不会生效，请使用三元表达式 -->
```

```
{{ if (ok) { return message } }}
```

- v-text

- **v-text**可以将一个变量的值渲染到指定的元素中,例如:

```
<div v-text="msg"></div>
new Vue({
  data:{
    msg:'hello ivan'
  }
});
```

输出结果:

```
<div>hello ivan</div>
```

- **v-html**

- 双大括号和**v-text**会将数据解释为纯文本,而非 HTML 。

为了输出真正的 HTML , 你需要使用 **v-html** 指令:

例如: `<div v-html="rawHtml"></div>`

```
new Vue({
  data:{
    rawHtml:'<h1>hello ivan</h1>'
  }
});
```

被插入的内容都会被当做 HTML,但是对于没有HTML标签的数据绑定时作用同**v-text**和`{{}}`

注意: 使用**v-html**渲染数据可能会非常危险,因为它很容易导致 XSS (跨站脚本) 攻击,使用的时候请谨慎,能够使用`{{}}`或者**v-text**实现的不要使用**v-html**

- **v-cloak**

- **v-cloak**指令保持在元素上直到关联实例结束编译后自动移除, **v-cloak**和 CSS 规则如 `[v-cloak] { display: none }` 一起用时,这个指令可以隐藏未编译的 Mustache 标签直到实例准备完毕。

通常用来防止`{{}}`表达式闪烁问题

例如:

```
<style>
  [v-cloak] { display: none }
</style>
```

`<!--` 在span上加上 **v-cloak**和css样式控制以后,浏览器在加载的时候会先把span隐藏起来,知道 Vue实例化完毕以后,才会将**v-cloak**从span上移除,那么css就会失去作用而将span中的内容呈现给用户 `-->`

```
<span v-cloak>{{msg}}</span>
```

```
new Vue({
  data:{
    msg:'hello ivan'
  }
});
```

- **v-model**以及双向数据绑定

- 1、在表单控件或者组件上创建双向绑定

- 2、v-model仅能在如下元素中使用：

input
select
textarea
components（Vue中的组件）

- 3、举例：

```
<input type="text" v-model="uname" />
```

```
new Vue({  
  data:{  
    uname:'' //这个属性值和input元素的值相互一一对应，二者任何一个的改变都会联动的改变对方  
  }  
})
```

- 4、修饰符（了解）：

.lazy - 取代 input 监听 change 事件
.number - 自动将输入的字符串转为数字
.trim - 自动将输入的内容首尾空格去掉

- v-bind

- 1、作用：可以给html元素或者组件动态地绑定一个或多个特性，例如动态绑定style和class

- 2、举例：

```
  
<div v-bind:class="{ red: isRed }"></div>  
<div v-bind:class="[classA, classB]"></div>  
<div v-bind:class="[classA, { classB: isB, classC: isC }]">  
<div v-bind:style="{ fontSize: size + 'px' }"></div>  
<div v-bind:style="[styleObjectA, styleObjectB]"></div>
```

- 3、缩写形式

```
  
<div :class="{ red: isRed }"></div>  
<div :class="[classA, classB]"></div>  
<div :class="[classA, { classB: isB, classC: isC }]">  
<div :style="{ fontSize: size + 'px' }"></div>  
<div :style="[styleObjectA, styleObjectB]"></div>
```

- v-for

1、作用：通常是遍历数组中的元素并指定模板内容生成内容

2、用法举例：

```
<div v-for="item in items">
  {{ item.text }}
</div>

new Vue({
  data:{
    items:[{text:'1'},{text:'2'}]
  }
});
```

3、可以为数组索引指定别名（或者用于对象的键）：

Vue1.0写法：

```
<div v-for="(index,item) in items"></div>
<div v-for="(key,val) in user"></div>

new Vue({
  data:{
    items:[{text:'1'},{text:'2'}],
    user:{uname:'ivan',age:32}
  }
});
```

Vue2.0写法：

```
<div v-for="(item, index) in items"></div>
<div v-for="(val, key) in user"></div>
<div v-for="(val, key, index) in user"></div>

new Vue({
  data:{
    items:[{text:'1'},{text:'2'}],
    user:{uname:'ivan',age:32}
  }
});
```

4、v-for 默认行为为不改变整体（为了性能考虑的设计），而是替换元素。迫使其重新排序的元素，在Vue2.0版本中需要提供一个 key 的特殊属性，在Vue1.0版本中需要提供一个 track-by="\$index"：

Vue2.0写法：

```
<div v-for="item in items" :key="item.id">
  {{ item.text }}
</div>
```

Vue1.0写法：

```
<div v-for="item in items" track-by="$index">
  {{ item.text }}
</div>
```

5、vue1.0与vue2.0对于v-for使用区别总结：

- 1、vue1.0中有\$index，而vue2.0中将\$index移除
- 2、vue1.0中 (index,item) in list 而 vue2.0 变成了 (item,index) in list的写法
- 3、vue1.0中使用 track-by来标记dom对象的唯一性，vue2.0中改成了 :key

- v-if

- 1、作用：根据表达式的值的真假条件来决定是否渲染元素，如果条件为**false**不渲染（达到隐藏元素的目的），为**true**则渲染。在切换时元素及它的数据绑定被销毁并重建

2、示例：

```
<!-- Handlebars 模板 -->
{{#if isShow}}
  <h1>Yes</h1>
{{/if}}
```

通常我们使用写法居多：

```
<h1 v-if="isShow">Yes</h1>
```

也可以用 **v-else** 添加一个 “else” 块：

```
<h1 v-if="isShow">Yes</h1>
<h1 v-else>No</h1>
```

注意：**v-else** 元素必须紧跟在 **v-if** 元素否则它不能被识别。

```
new Vue({
  data:{
    isShow:true
  }
});
```

- v-show

- 1、根据表达式的真假值，切换元素的 **display** CSS 属性，如果为**false**，则在元素上添加 **display:none**来隐藏元素，否则移除**display:none**实现显示元素

2、示例：

```
<h1 v-show="isShow">Yes</h1>
```

```
new Vue({
  data:{
    isShow:true
  }
});
```

3、**v-if**和**v-show**的总结：

v-if和**v-show** 都能够实现对一个元素的隐藏和显示操作,但是**v-if**是将这个元素添加或者移除到**dom**中,而**v-show**

是在这个元素上添加 **style="display:none"**和移除它来控制元素的显示和隐藏的

- v-on

- 1、作用：绑定事件监听，表达式可以是一个方法的名字或一个内联语句，如果没有修饰符也可以省略，用在普通的html元素上时，只能监听 原生 DOM 事件。用在自定义元素组件上时，也可以监听子组件触发的自定义事件。

2、常用事件：

```
v-on:click
v-on:keydown
v-on:keyup
v-on:mousedown
v-on:mouseover
v-on:submit
....
```

3、v-on提供了很多事件修饰符来辅助实现一些功能，例如阻止冒泡等事件修饰符有如下：

```
.stop - 调用 event.stopPropagation()。
.prevent - 调用 event.preventDefault()。
.capture - 添加事件侦听器时使用 capture 模式。
.self - 只当事件是从侦听器绑定的元素本身触发时才触发回调。
.{keyCode | keyAlias} - 只当事件是从侦听器绑定的元素本身触发时才触发回调。
.native - 监听组件根元素的原生事件。
```

4、示例：

```
<!-- 方法处理器 -->
<button v-on:click="doThis"></button>
<!-- 内联语句 -->
<button v-on:click="doThat('hello', $event)"></button>
<!-- 缩写 -->
<button @click="doThis"></button>
<!-- 停止冒泡 -->
<button @click.stop="doThis"></button>
<!-- 阻止默认行为 -->
<button @click.prevent="doThis"></button>
<!-- 阻止默认行为，没有表达式 -->
<form @submit.prevent></form>
<!-- 串联修饰符 -->
<button @click.stop.prevent="doThis"></button>
```

5、v-on的缩写形式：可以使用@替代 v-on：

```
<button @click="doThis"></button>
```

1.4.4 利用系统指令实现品牌管理案例

- 目的

通过案例熟悉Vue系统指令的用法和结构的写法练习，做到知识点的巩固和学会应用，并且在案例的扩展需求中，学习Vue新知识点，做到先有需求，再有知识点学习，最后到知识点的应用，让学员学习知识点的同时学会知识点的应用

- 实现品牌列表数据展示

- ○ 效果

编号	品牌名称	创建时间	操作
1	宝马	Thu Mar 09 2017 10:24:05 GMT+0800 (中国标准时间)	删除
2	奥迪	Thu Mar 09 2017 10:24:05 GMT+0800 (中国标准时间)	删除

- 代码

```
<script src="vue1028.js"></script>
</head>
<body>
  <div id="app">
    <!-- 品牌列表区域 -->
    <table class="list">
      <tr>
        <th>编号</th>
        <th>品牌名称</th>
        <th>创建时间</th>
        <th>操作</th>
      </tr>
      <tr v-for="item in list" track-by="$index">
        <td>{{item.id}}</td>
        <td>{{item.name}}</td>
        <td>{{item.ctime}}</td>
        <td>
          <a href="#" >删除</a>
        </td>
      </tr>
    </table>
  </div>
  <script>
    new Vue({
      el: '#app',
      data: {
        list: [
          {id: 1, name: '宝马', ctime: new Date()},
          {id: 2, name: '奥迪', ctime: new Date()}
        ]
      }
    });
  </script>
```

1、导入js

2、v-for实现列表

3、实例化Vue对象定义模型数据

- 实现品牌数据添加
-
- 效果

0

添加品牌

编号	品牌名称	创建时间	操作
1	宝马	Thu Mar 09 2017 13:59:49 GMT+0800 (中国标准时间)	删除
2	奥迪	Thu Mar 09 2017 13:59:49 GMT+0800 (中国标准时间)	删除

- 代码

```

<script src="vue1028.js"></script>
</head>
<body>
  <div id="app">
    <input type="text" v-model="productid" >
    <input type="text" v-model="productname" >
    <button @click="addProduct">添加品牌</button>
    <!-- 品牌列表区域 -->
    <table class="list">
      <tr>
        <th>编号</th>
        <th>品牌名称</th>
        <th>创建时间</th>
        <th>操作</th>
      </tr>
      <tr v-for="item in list" track-by="$index">
        <td>{{item.id}}</td>
        <td>{{item.name}}</td>
        <td>{{item.ctime}}</td>
        <td>
          <a href="#" @click="del(item.id)">删除</a>
        </td>
      </tr>
    </table>
  </div>

```

```

<script>
new Vue({
  el: '#app',
  data: {
    list: [
      {id: 1, name: '宝马', ctime: new Date()},
      {id: 2, name: '奥迪', ctime: new Date()}
    ],
    productid: 0,
    productname: ''
  },
  methods: {
    // 1.0 删除
    del: function(id) {
      // 将list数组中的对应的元素删除即可
      alert(id);
    },
    // 2.0 添加
    addProduct: function() {
      // alert(this.productid + " , "+this.productname);
      // 1.0 获取到页面上的数据
      var pobj = {id: this.productid, name: this.productname, ctime: new Date()};
      // 2.0 添加数据
      this.list.push(pobj);

      // 3.0 清空模型productid和productname的值
      this.productname = '';
      this.productid = 0;
    }
  }
});
</script>

```

- 实现品牌数据删除
- 效果

编号	品牌名称	创建时间	操作
1	宝马	Thu Mar 09 2017 15:55:00 GMT+0800 (中国标准时间)	删除
2	奥迪	Thu Mar 09 2017 15:55:00 GMT+0800 (中国标准时间)	删除

此网页显示：

您真的要删除数据吗？

- 代码


```

<script src="vue1028.js"></script>
</head>
<body>
  <div id="app">
    <input type="text" v-model="productid" >
    <input type="text" v-model="productname" >
    <button @click="addProduct">添加品牌</button>

    <!-- 品牌列表区域 -->
    <table class="list">
      <tr>
        <th>编号</th>
        <th>品牌名称</th>
        <th>创建时间</th>
        <th>操作</th>
      </tr>
      <tr v-for="item in list" track-by="$index">
        <td>{{item.id}}</td>
        <td>{{item.name}}</td>
        <td>{{item.ctime}}</td>
        <td>
          <a href="javascript:void(0)" @click="del(item.id)">删除</a>
        </td>
      </tr>
    </table>
  </div>

```



```

<script>

```

```

<script>

```

```

new Vue({
  el: '#app',
  data: {
    list: [
      {id: 1, name: '宝马', ctime: new Date()},
      {id: 2, name: '奥迪', ctime: new Date()}
    ],
    productid: 0,
    productname: ''
  },
  methods: {
    //1.0 删除
    del: function(id) {
      if(!confirm('您真的要删除数据吗?')) {
        return;
      }
      //将list数组中的对应的元素删除即可
      var index = this.list.findIndex(function(obj) {
        return obj.id == id;
      });

      //根据索引将list数组的数据删除了，这个时候会自动调用v-for进行重新生成数据
      this.list.splice(index, 1);
    },
  },

```

- 实现根据品牌名称过滤符合条件的数据
- 效果

编号	品牌名称	创建时间	操作
1	宝马	Thu Mar 09 2017 16:05:09 GMT+0800 (中国标准时间)	删除

- 代码

```

<div>
  <input type="text" placeholder="请输入搜索内容" v-model="searchValue">
</div>

<!-- 品牌列表区域 -->
<table class="list">
  <tr>
    <th>编号</th>
    <th>品牌名称</th>
    <th>创建时间</th>
    <th>操作</th>
  </tr>
  <tr v-for="item in list | filterBy searchValue in 'name'" track-by="$index">
    <td>{{item.id}}</td>
    <td>{{item.name}}</td>
    <td>{{item.ctime}}</td>
    <td>

```

1.4.5 Vue在Chrome浏览器的调试工具Vue-Devtools

- 作用

Vue-Devtools是Chrome浏览器的一个扩展，通过Vue-Devtools可以实现在Chrome浏览器的调试工具栏中看到Vue开发页面的相关数据对象，方法，事件，状态信息，方便程序员监控和调试解决问题

- 地址

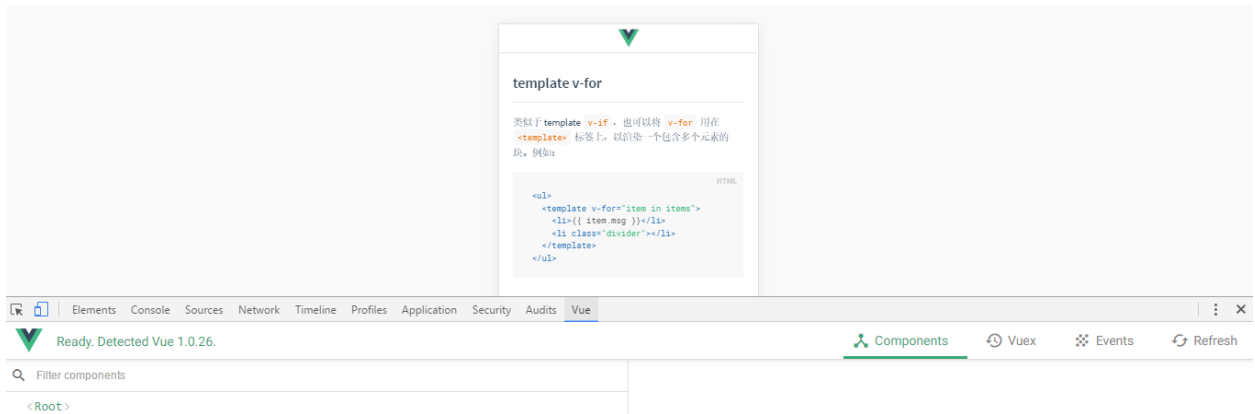
GitHub地址: <https://github.com/vuejs/vue-devtools>

Chrome插件地址: <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnanhbledajbpd?hl=zh-CN>

- 通过Chrome插件地址安装插件(注意：这种方式需要翻墙)
 - 1、在Chrome浏览器中打开地址: <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnanhbledajbpd?hl=zh-CN>
 - 2、点击里面的“+ 添加至CHROME”按钮即可安装插件



- 3、安装以后，在Chrome浏览器中打开使用Vue开发的站点后按F12打开调试工具即可看到Vue调试工具



- 通过Vue-DevTools源码安装（需要先安装node.exe）
 - 1、<https://nodejs.org/en/> 下载node.exe安装
 - 2、去<https://github.com/vuejs/vue-devtools> 下载到文件
 - 3、进入vue-devtools-master工程 先执行npm install再执行npm run build
 - 4、进入vue-devtools-master\shells\chrome文件夹中修改manifest.json 中的persistant为true
 - 5、打开谷歌浏览器设置--->扩展程序-->勾选开发者模式--->加载已解压的扩展程序--->选择“vue-devtools-master\shells下的chrome”文件夹，至此恭喜已经安装成功！