

# 一、自我介绍

---

李树根 （根儿哥）

13年毕业于重庆大学

从事于源码时代 产品总监

北京博能 机场数字孪生 研发主管 江北机场、天府机场、大兴机场、宁波机场。。。。。

重庆海特 公路工程质量检测系统 项目经理 HIPMS

BIM+GIS+Web3D

spring mybatis springboot springcloud springsecurity vue react uniapp  
.....

# 二、课堂纪律

---

考勤（迟到的惩罚：表演节目，旷课是原则性问题）

课上：坚决不允许玩游戏、刷视频

请假问题：不允许霸王假

# 三、3周的计划安排

---

第一周：技术的铺垫、巩固、提升

前后端分离

前端（Vue2）：

小程序 uniapp

WEB ElementUI

后端：

springboot(spring+springmvc)

mybatis/mybatisplus

springsecurity/shiro

数据存储:

mysql

redis

分布式文件系统 (fastdfs、阿里云、七牛云)

第二周+第三周: 项目为主

给大家需求

设计 (原型、概要设计、数据库设计)

开发

WBS

(Project、禅道、腾讯coding) 项目的进度、项目的质量

测试

测试用例 --- bug --- 改bug

上线

devops

项目演示: (PPT + 项目 小组选一个代表进行演示)

## 四、小调研

---

技术调研:

0、html css js

1、vue2 elementUI (没学过)

## 2、JavaSE JDBC

3、开发工具 idea vscode

4、项目构建 maven npm

# 五、Vue

Vue React Angular 前端

Vue 鸿蒙 App android IOS

鸿蒙：思想还是android的思想 语言成了ArkTS ----> TypeScript -----  
-> JavaScript

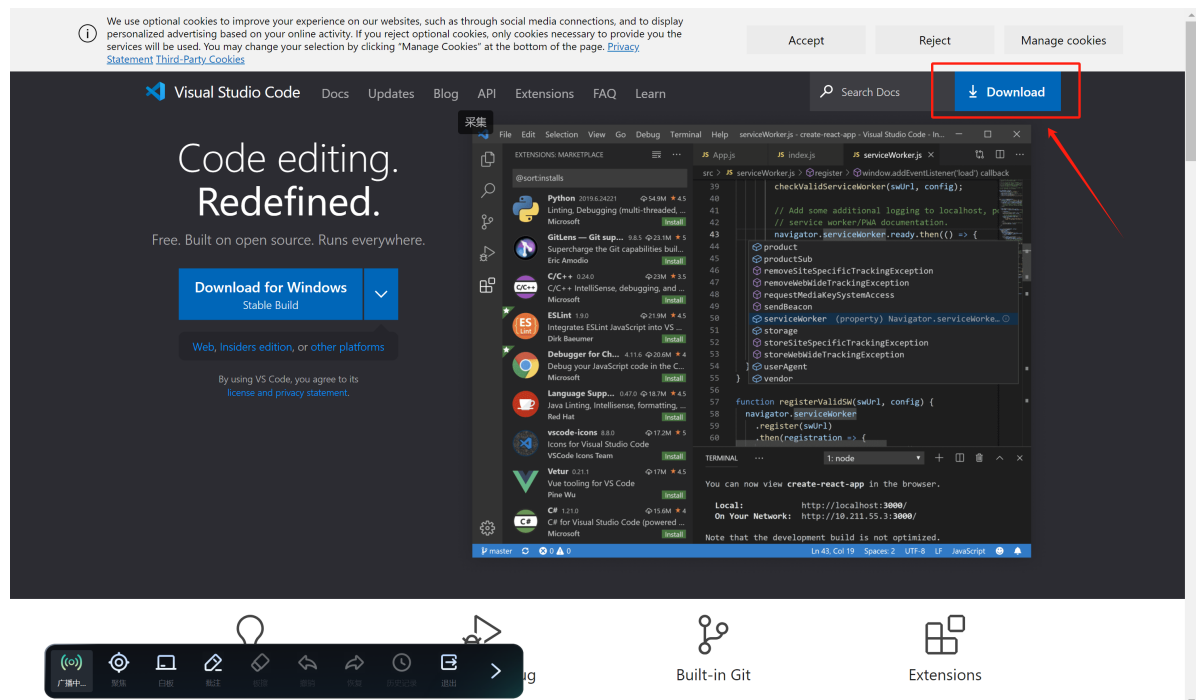
<https://cn.vuejs.org/>



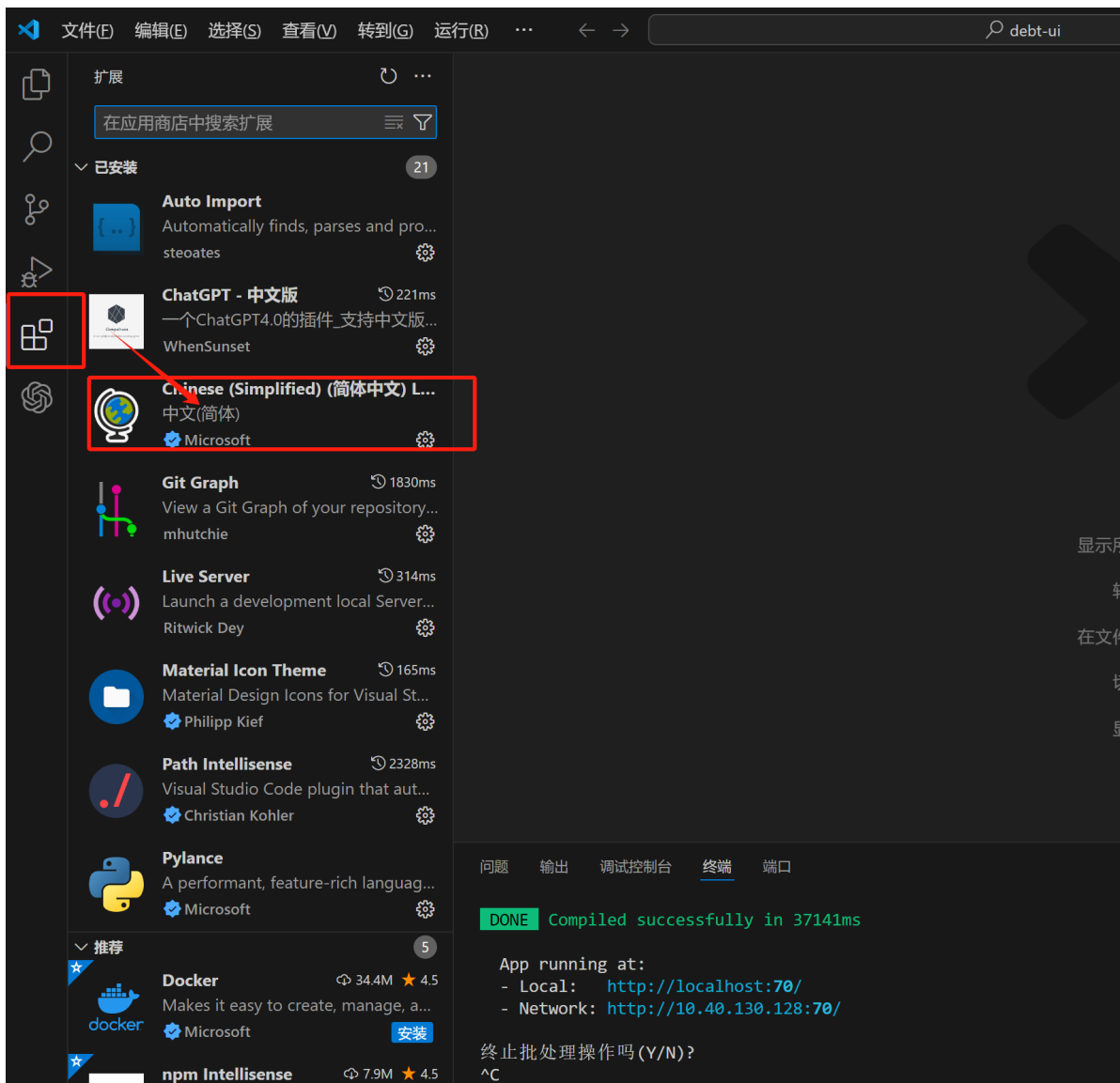
## 5.1 前端环境搭建

- 开发工具：vscode
- 运行环境：node (jre)
  - js 运行在（浏览器）客户端的脚本语言

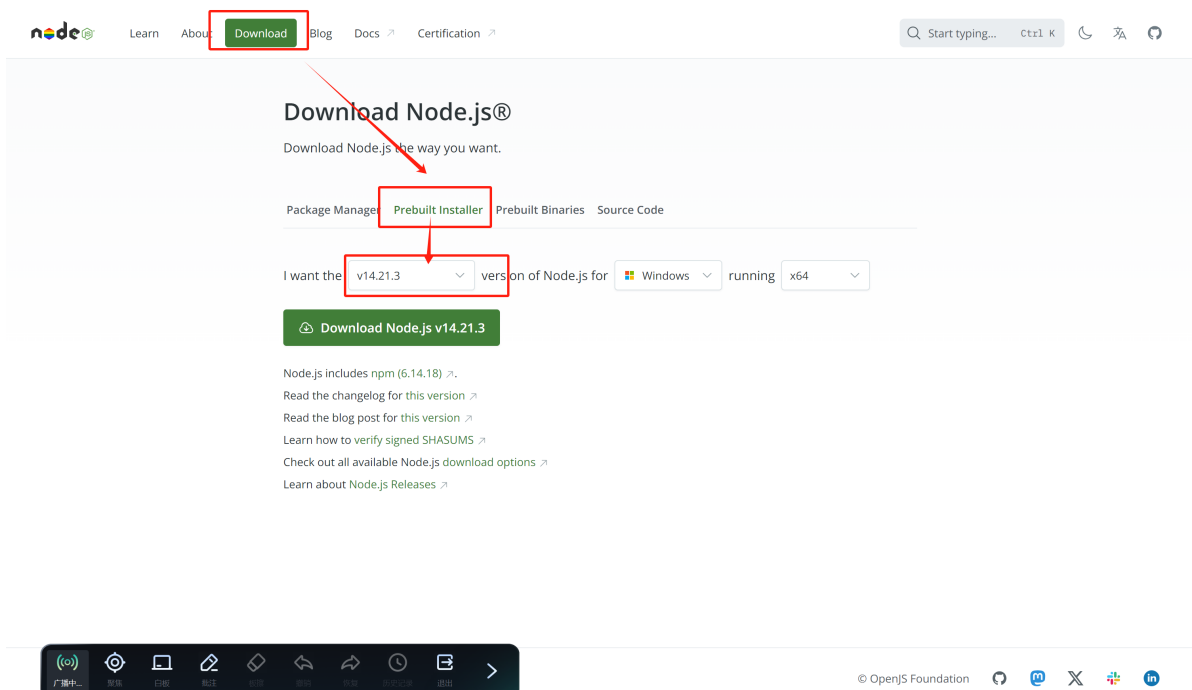
<https://code.visualstudio.com/>



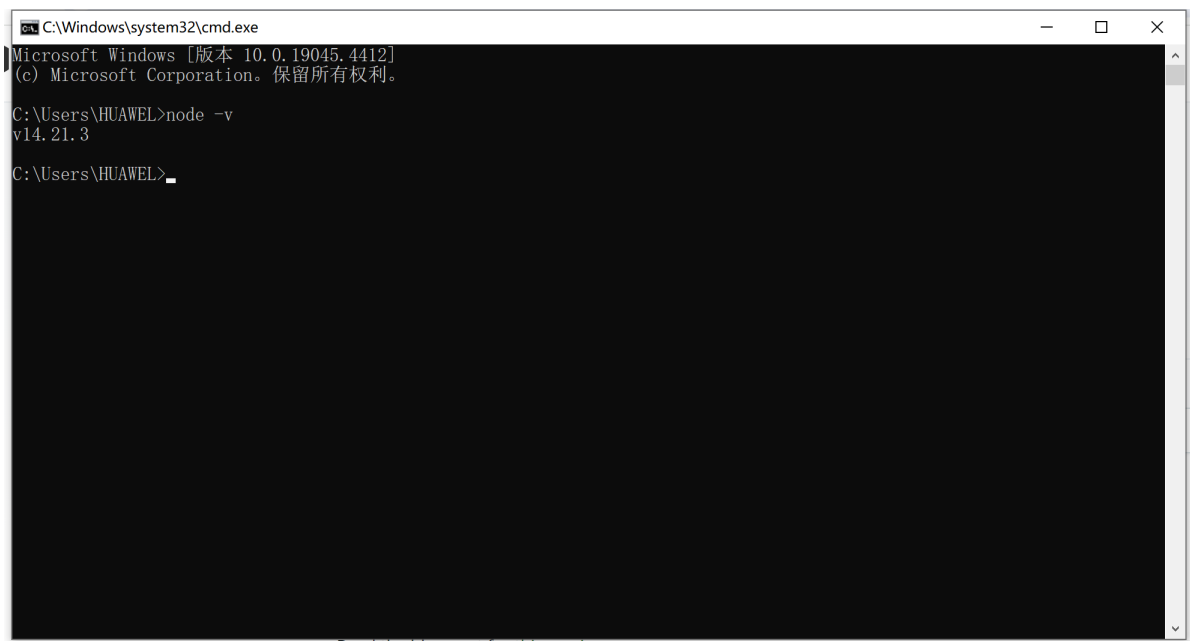
下载中文插件



node安装: <https://nodejs.org/en/download/prebuilt-installer>



测试是否安装成功：



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.4412]
(c) Microsoft Corporation。保留所有权利。

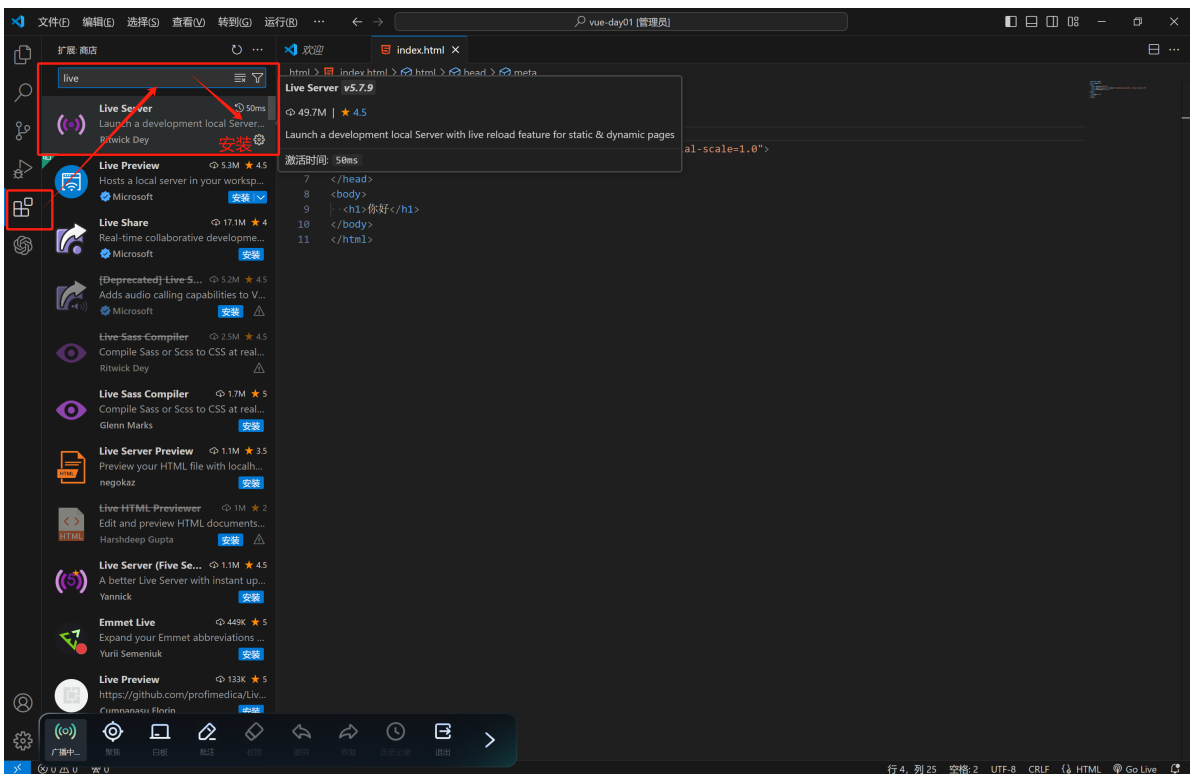
C:\Users\HUAWEI>node -v
v14.21.3

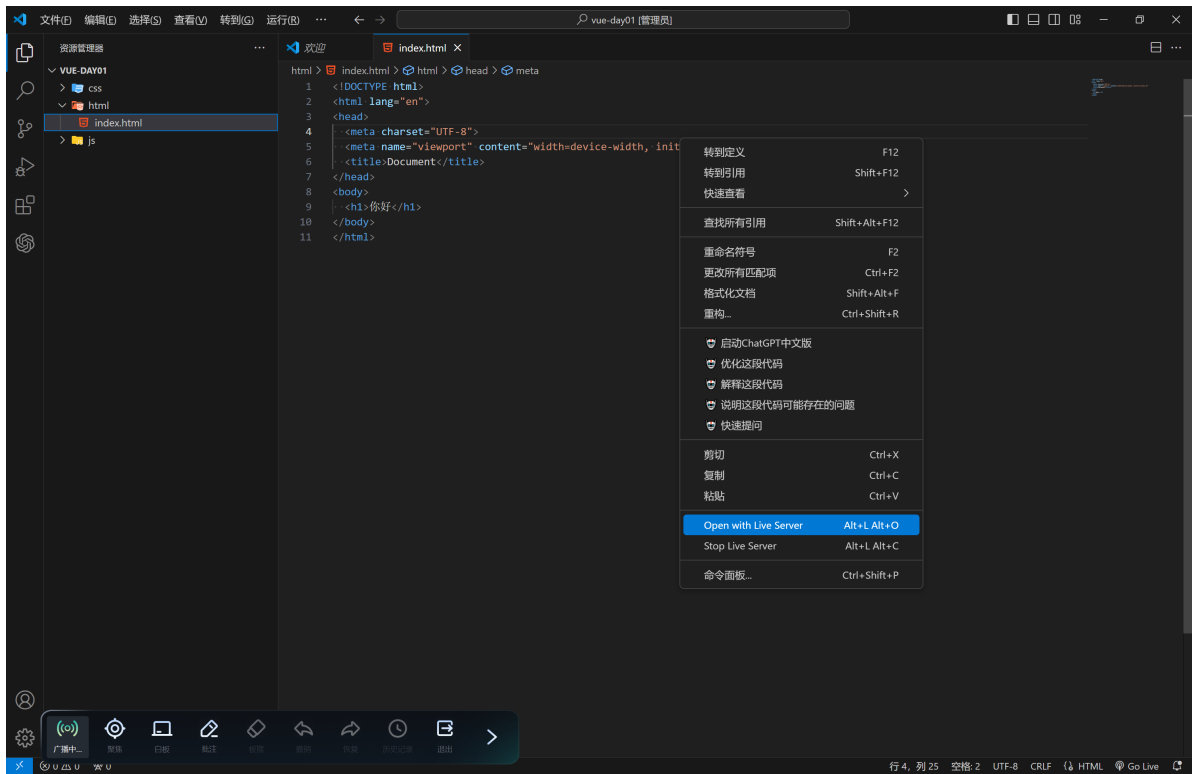
C:\Users\HUAWEI>
```

尝试使用vue

## 5.2 学习Vue的基础语法

写完前端项目，预览某个html页面，安装live-server轻量服务器软件





## 5.2.1 Vue快速入门

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <!-- 1、引入vue.js文件 -->
  <!-- 离线 -->
  <script src="../js/vue.js"></script>
  <!-- 在线 -->
  <!-- <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js">
</script> -->
  <title>Document</title>
</head>
<body>

  <div id="app">
    <div>
      <button @click="handleClick">{{ message }}</button>
```

```
<h1>{{ testMethod() }}</h1>

</div>
</div>

<script>

  new Vue({

    // 挂载：vue的代码在哪个dom标签下生效
    el: "#app",
    // vue通常的思维是将data中的数据绑定到页面上，更改data的值，
    页面自动渲染
    data(){
      return {
        message: '点我试试'
      }
    },
    // vue中定义函数，统一写在methods中，定义的函数在vue的范围内
    都可以使用了
    methods: {
      handleClick(){
        alert("44944")
      },
      testMethod(){
        return 123
      }
    }
  })

</script>

</body>
</html>
```



当我点击-或者+，修改数量和总价和共计，如果-到0，则自动删除这条数据，如果所有的数据都没有了，则隐藏表格，显示：空空如也

序号	名称	单价	数量	总价
1	小米su7	200000	<input type="text" value="2"/>	400000
2	华为P70Pro	8000	<input type="text" value="2"/>	16000
3	问界M9	400000	<input type="text" value="1"/>	400000
共计	816000			

步骤：

- 搭建页面基本结构

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js">
</script>
  <title>Document</title>

</head>
<body>

  <div id="app">

</div>

<script>
```

```

    new Vue({

      el: '#app',
      data(){
        return {

        }
      },
      methods:{

      }
    })

  </script>

</body>
</html>

```

- 组织data, 渲染到页面

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js">
</script>
  <title>Document</title>
  <style type="text/css">
    table {
      width: 500px;
      margin: auto;
      border-collapse: collapse;
      text-align: center;
    }
    tr {
      height: 40px;
    }
    td {

```

```
border: 1px solid #000;
}
</style>
</head>
<body>
```

```
<div id="app">
```

```
<table>
  <tr>
    <td>序号</td>
    <td>名称</td>
    <td>单价</td>
    <td>数量</td>
    <td>总价</td>
```

```
</tr>
```

```
<!--
```

**v-for** vue的指令：循环

**a in b**

**b** 是要循环的对象或者数组

**a** 是变量，存放每一次循环出来的元素

**data methods** 指令中是不需要加{{{}}

在html的内容中需要加{{{}}

**v-for="(product,index) in products"**

**index** 就是每次循环的索引，从0开始

```
-->
```

```
<tr v-for="(product,index) in products">
```

```
<td>{{index + 1}}</td>
```

```
<td>{{product.name}}</td>
```

```
<td>{{product.price}}</td>
```

```
<td>
```

```
<!--
```

**v-on:事件="处理函数"**

**@事件="处理函数"**

```
-->
```

```

        <button v-on:click="handleDecrement()">-
</button>
        {{product.num}}
        <button @click="handleIncrement()">+</button>
    </td>
    <td>{{product.num * product.price}}</td>
</tr>

<tr>
    <td>总计: </td>
    <!--
        colspan="4" 表示这个单元格横向合并4个单元格
    -->
    <td colspan="4">{{ computeTotal() }}</td>
</tr>
</table>
</div>

<script>

```

```

new Vue({

    el: '#app',
    data(){
        return {
            products:[{
                name: '小米su7',
                price: 200000,
                num: 2
            },{
                name: '问界M9',
                price: 390000,
                num: 2
            },{
                name: '华为P70 Pro',
                price: 8000,
                num: 2
            },{
                name: '华为电视',
                price: 7000,
                num: 2
            }
        ]
    }
})

```

```

        }]
      }
    },
    methods: {
      // 计算总计
      computeTotal() {
        let total = 0;
        for (let i = 0; i < this.products.length; i++) {
          let product = this.products[i];
          total += product.price * product.num
        }
        return total;
      },
      // 加
      handleIncrement(index) {

      },
      // 减
      handleDecrement(index) {

      }
    }
  })
</script>

</body>
</html>

```

- 处理加和减

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js">
</script>
  <title>Document</title>

```

```
<style type="text/css">
  table {
    width: 500px;
    margin: auto;
    border-collapse: collapse;
    text-align: center;
  }
  tr {
    height: 40px;
  }
  td {
    border: 1px solid #000;
  }
</style>
</head>
<body>
```

```
<div id="app">
```

```
<table>
  <tr>
    <td>序号</td>
    <td>名称</td>
    <td>单价</td>
    <td>数量</td>
    <td>总价</td>
  </tr>
<!--
```

**v-for** vue的指令：循环

**a in b**

**b** 是要循环的对象或者数组

**a** 是变量，存放每一次循环出来的元素

**data methods** 指令中是不需要加{{{}}

在html的内容中需要加{{{}}

**v-for="(product,index) in products"**

**index** 就是每次循环的索引，从0开始

```

-->
<tr v-for="(product,index) in products">
  <td>{{index + 1}}</td>
  <td>{{product.name}}</td>
  <td>{{product.price}}</td>
  <td>
    <!--
      v-on:事件="处理函数"
      @事件="处理函数"
    -->
    <button v-on:click="handleDecrement(index)">-
  </button>
    {{product.num}}
    <button @click="handleIncrement(index)">+
  </button>
  </td>
  <td>{{product.num * product.price}}</td>
</tr>

<tr>
  <td>总计: </td>
  <!--
    colspan="4" 表示这个单元格横向合并4个单元格
  -->
  <td colspan="4">{{ computeTotal() }}</td>
</tr>
</table>
</div>

<script>

new Vue({

  el: '#app',
  data(){
    return {
      products:[{
        name: '小米su7',
        price: 200000,
        num: 2

```

```

        }, {
            name: '问界M9',
            price: 390000,
            num: 2
        }, {
            name: '华为P70 Pro',
            price: 8000,
            num: 2
        }, {
            name: '华为电视',
            price: 7000,
            num: 2
        }
    ]
},
methods: {
    // 计算总计
    computeTotal() {
        let total = 0;
        for (let i = 0; i < this.products.length; i++) {
            let product = this.products[i];
            total += product.price * product.num
        }
        return total;
    },
    // 加
    handleIncrement(index) {
        this.products[index].num++
    },
    // 减
    handleDecrement(index) {
        this.products[index].num--
        if (this.products[index].num <= 0) {
            // 从products数组中删除当前元素 splice自己去学
            this.products.splice(index, 1)
        }
    }
}
})

```

习



```
</script>

</body>
</html>
```

- 处理空空如也

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js">
</script>
  <title>Document</title>
  <style type="text/css">
    table {
      width: 500px;
      margin: auto;
      border-collapse: collapse;
      text-align: center;
    }
    tr {
      height: 40px;
    }
    td {
      border: 1px solid #000;
    }
  </style>
</head>
<body>

  <div id="app">
    <!--
      v-if 指令 判断为true, 则渲染这个标签, 为false则不渲染
    -->
    <table v-if="products.length > 0">
```

```

<tr>
  <td>序号</td>
  <td>名称</td>
  <td>单价</td>
  <td>数量</td>
  <td>总价</td>

```

```

</tr>

```

```

<!--

```

**v-for** vue的指令：循环

**a in b**

**b** 是要循环的对象或者数组

**a** 是变量，存放每一次循环出来的元素

**data methods** 指令中是不需要加{{{}}

在html的内容中需要加{{{}}

**v-for="(product,index) in products"**

**index** 就是每次循环的索引，从0开始

```

-->

```

```

<tr v-for="(product,index) in products">

```

```

  <td>{{index + 1}}</td>

```

```

  <td>{{product.name}}</td>

```

```

  <td>{{product.price}}</td>

```

```

  <td>

```

```

    <!--

```

**v-on:事件="处理函数"**

**@事件="处理函数"**

```

-->

```

```

<button v-on:click="handleDecrement(index)">-

```

```

</button>

```

```

  {{product.num}}

```

```

  <button @click="handleIncrement(index)">+

```

```

</button>

```

```

  </td>

```

```

  <td>{{product.num * product.price}}</td>

```

```

</tr>

```

```

<tr>

```

```

  <td>总计: </td>

```

```
      <!--
      colspan="4" 表示这个单元格横向合并4个单元格
    -->
    <td colspan="4">{{ computeTotal() }}</td>
  </tr>
</table>
<div v-else>空空如也</div>
</div>

<script>
```

```
new Vue({

  el: '#app',
  data(){
    return {
      products:[{
        name: '小米su7',
        price: 200000,
        num: 2
      },{
        name: '问界M9',
        price: 390000,
        num: 2
      },{
        name: '华为P70 Pro',
        price: 8000,
        num: 2
      },{
        name: '华为电视',
        price: 7000,
        num: 2
      }]
    }
  },
  methods:{
    // 计算总计
    computeTotal(){
      let total = 0;
      for(let i=0; i<this.products.length; i++){
        let product = this.products[i];
```

```

        total += product.price * product.num
    }
    return total;
},
// 加
handleIncrement(index){
    this.products[index].num++
},
// 减
handleDecrement(index){
    this.products[index].num--
    if(this.products[index].num <= 0){
        // 从products数组中删除当前元素 splice自己去学
        this.products.splice(index,1)
    }
}
}
})

</script>

</body>
</html>

```

## 5.3 学习使用Vue搭建前端项目

