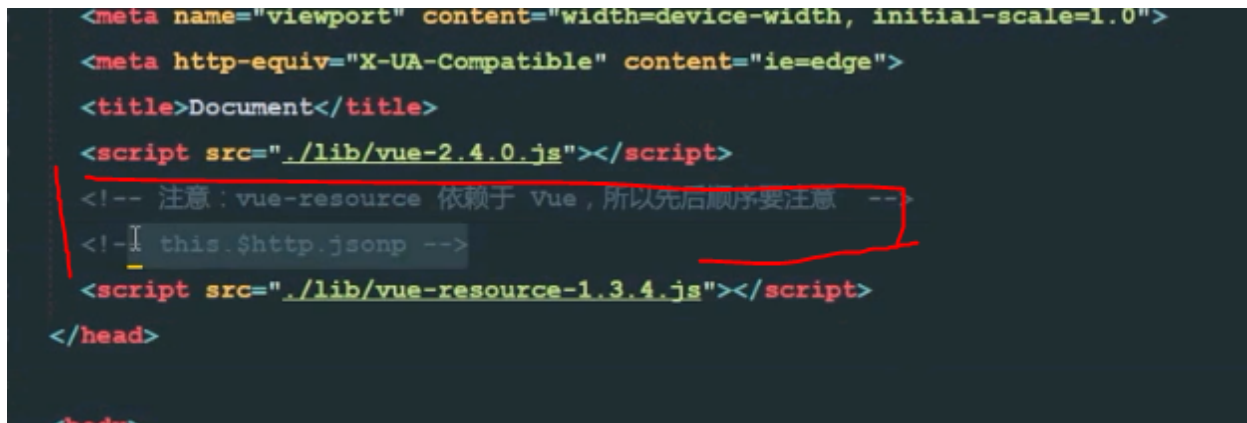


1: 导包顺序的先后，一定要先把vue的包导入
再导入resource的包



1. vue-resource 的配置步骤:

- 直接在页面中，通过script标签，引入 vue-resource 的脚本文件;
- 注意：引用的先后顺序是：先引用 Vue 的脚本文件，再引用 vue-resource 的脚本文件;

发送get请求:

```
getInfo() { // get 方式获取数据
  this.$http.get('http://127.0.0.1:8899/api/getlunbo').then(res => {
    console.log(res.body);
  })
}
```

发送post请求:

```
postInfo() {
  var url = 'http://127.0.0.1:8899/api/post';
  // post 方法接收三个参数:
  // 参数1: 要请求的URL地址
  // 参数2: 要发送的数据对象
  // 参数3: 指定post提交的编码类型为 application/x-www-form-urlencoded
  this.$http.post(url, { name: 'zs' }, { emulateJSON: true }).then(res => {
    console.log(res.body);
  });
}
```

发送JSONP请求获取数据:

```
jsonpInfo() { // JSONP形式从服务器获取数据
  var url = 'http://127.0.0.1:8899/api/jsonp';
  this.$http.jsonp(url).then(res => {
    console.log(res.body);
  });
}
```

全局配置根路径的跟

```
<script>
// 如果我们通过全局配置了, 请求的数据接口 根域名, 则 , 在每次单独发起 http 请求的时候, 请求
的 url 路径, 应该以相对路径开头, 前面不能带 / , 否则 不会启用根路径做拼接;
Vue.http.options.root = 'http://vue.studyit.io/';_ I

// 创建 Vue 实例, 得到 ViewModel
var vm = new Vue({
  el: '#app',
  data: {
    name: '',
    list: [ // 存放所有品牌列表的数组
    ]
  },
  created() { // 当 vm 实例 的 data 和 methods 初始化完毕后, vm实例会自动执行
    created 这个生命周期函数
    this.getAllList()
  },
  methods: {
    getAllList() { // 获取所有的品牌列表
      // 分析:
```