

全栈工程师 系列丛书

Vue.js

前端开发技术

王凤丽 豆连军 编著



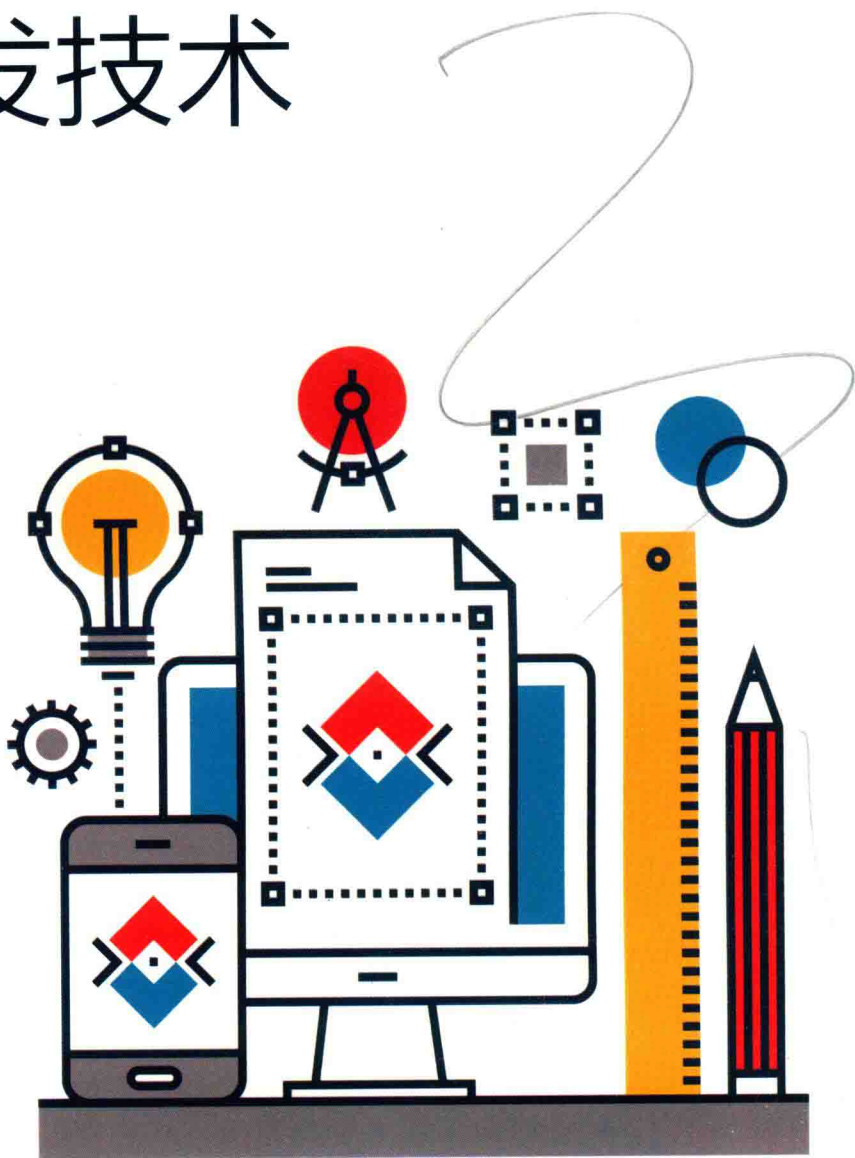
案例实战：全书以案例应用、工程化项目实战为目标，提供 136 个实例和案例代码分析



名师编著：由具有 18 年软件开发和培训经验，并申报多项横向科技项目的实战派讲师王凤丽、“斤斗云学堂”教学总监豆连军共同编著



资源丰富：本书提供了配套视频、电子教案、教学课件、通关测试、案例源代码等大量学习资源

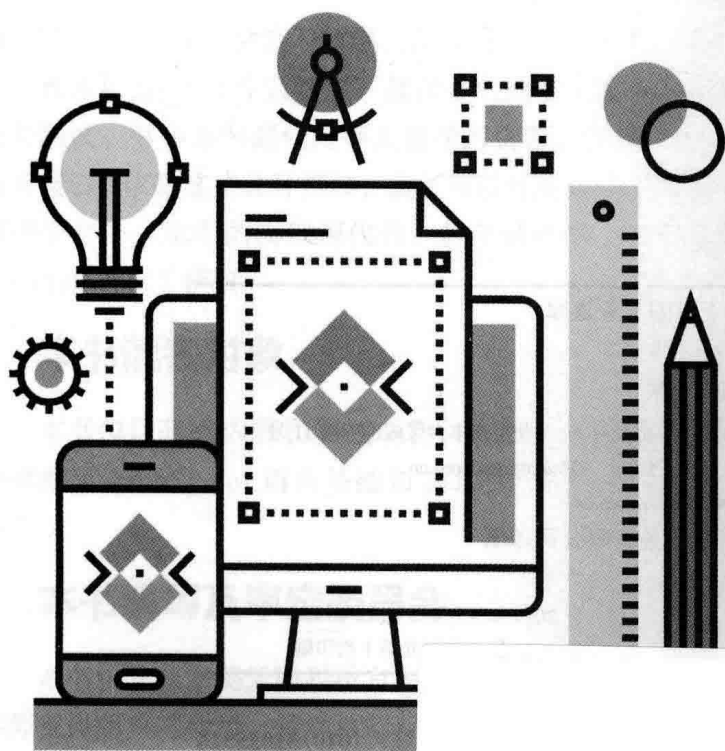


全栈工程师 系列丛书

Vue.js

前端开发技术

王凤丽 豆连军 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Vue.js 前端开发技术 / 王凤丽, 豆连军编著. — 北京: 人民邮电出版社, 2019.6
ISBN 978-7-115-50805-8

I. ①V… II. ①王… ②豆… III. ①网页制作工具—程序设计 IV. ①TP392.092.2

中国版本图书馆CIP数据核字(2019)第028072号

内 容 提 要

本书首先介绍了一个“Hello World”案例,帮助读者入门;然后分章节介绍了数据绑定、指令、事件等 Vue 带给我们的非常好用的功能;接下来介绍了组件编程,这是 Vue 富有魅力的部分,掌握了组件编程,基本上可以实现大多数 UI 交互设计;紧接着介绍了自定义指令、过渡与动画、渲染函数等高级概念,这些可以帮助我们编写更简洁的代码,实现更绚丽的交互效果;最后,介绍了路由、webpack 等前端工程自动化工具和脚手架模式编程,并辅以一个综合案例与项目实战应用。

本书内容安排循序渐进,讲解通俗易懂,把基本概念与应用场景通过案例教学紧密联系在一起。学生可以用本书自学,教师也可以将本书作为教材或参考工具书,其中的案例代码可用于课堂教学。

本书可作为高等院校、职业院校前端开发课程的教材或参考书,也可作为社会岗前培训班的培训教材,同时也可作为 Vue 自学者入门指南用书。

-
- ◆ 编 著 王凤丽 豆连军
责任编辑 刘 博
责任印制 陈 犇
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市中晟雅豪印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 16.25 2019年6月第1版
字数: 426千字 2019年6月河北第1次印刷
-

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

前言

用心做教育，专心做教育，编写一本能帮助初学者快速学习 Vue 的教程，一直是“斤斗云学堂”教师们的愿望，现在终于得以实现。本书从入门开发到工程化项目实战，每个案例都经过了精心设计。本书以当前发布的 Vue 新版本为介绍对象，内容包括基础知识、综合案例和项目实战应用，力求体现“易学”“实用”的特色。为方便更多的 JavaScript 初学者掌握本书内容，书中的实例代码尽量采用 ES5 语法规范，但为了更好地表达 Vue 的优势特性，也有少数案例使用了 ES6、ES7 的语法。

本书亮点

案例教学：全书有 136 个实例和案例代码分析，便于快速入门。

实战应用：以实战应用、工程化项目实战为目标，有效避免了“一叶障目不见森林”。

大量图示：全书重点、难点用大量的图示标记，更加直观、生动。

名师编著：由具有 12 年软件开发和培训经验，并申报多项横向科技项目的实战派讲师王凤丽、“斤斗云学堂”教学总监豆连军共同编著，确保图书的实战性。

慕课平台：“斤斗云学堂”慕课平台提供了本书的配套视频、电子教案、教学课件、通关测试、书中案例源代码等大量学习资源，同时公开了本书案例中所用到的后端 API 通信接口。依靠这些公开接口，读者可以开发自己的网络单词本应用。除了“斤斗云学堂”慕课平台外，本书的实例源代码、教学课件等，也可登录人邮教育社区（<http://www.ryjiaoyu.com>）获取。

本书读者对象

本书包括基础内容和实战项目，方便初次接触 Vue 的读者快速入门。本书适合已经掌握了 JavaScript 语言基础和 HTML、CSS 基础的读者，也适合转型 Vue 的前端开发人员。

本书编写及审定委员会

本书由北京乐美无限科技有限公司授权提供 Vue 课程内容素材，由学科带头人、金牌讲师王凤丽执笔编写，教学总监豆连军审核并统改。以下人员对本书进行了审阅和校验，他们是贡雪静、董爽、张旭磊、崔阿超。在此也一并感谢在本书编写过程中给予支持的家人们。

寄语读者

互联网开发是一个高薪技术行业，是一个充满挑战、富有创造力的行业，但并不是高不可攀的行业。初学者入门以后，一定能体会到技术开发的乐趣。希望本书能帮助心中有梦想的青年学子实现梦想，早日踏入互联网开发行业，预祝大家学业、事业有成！

编者

目 录

第1章 Vue入门	1	2.2.3 Vue双向数据绑定	22
1.1 Vue简述	1	2.3 Vue计算属性	24
1.1.1 什么是Vue.js	1	2.3.1 计算属性	24
1.1.2 为什么使用Vue.js	2	2.3.2 计算属性与methods的区别	27
1.1.3 Vue.js的主要特点	2	2.4 Vue生命周期	28
1.1.4 Vue.js的优势	3	2.4.1 Vue实例生命周期图解	28
1.2 Vue.js的下载及使用	3	2.4.2 Vue生命周期详解	29
1.2.1 Vue.js下载	4	2.4.3 Vue各生命周期适合开发的 业务逻辑	31
1.2.2 Vue.js引用	4	本章小结	31
1.2.3 npm安装	4	习题	32
1.2.4 解读Vue项目文件目录结构	8	第3章 Vue指令	33
1.3 实例化Vue对象、数据和方法	9	3.1 Vue指令概述	33
1.3.1 实例化Vue对象	9	3.1.1 指令	33
1.3.2 Vue数据和方法	10	3.1.2 指令修饰符	34
1.3.3 将数据挂载到DOM页面	11	3.2 Vue指令详解	34
1.4 MVVM模式	13	3.2.1 指令v-if	34
本章小结	14	3.2.2 指令v-for	36
习题	15	3.2.3 指令v-on	41
第2章 Vue数据绑定	16	3.2.4 指令v-show	43
2.1 Vue模板语法	16	3.3 Vue动态样式绑定	46
2.1.1 模板语法	16	3.3.1 v-bind指令属性	46
2.1.2 插值	17	3.3.2 v-bind绑定class	47
2.1.3 表达式	17	3.3.3 v-bind数组表达式绑定class 属性	48
2.2 响应式声明渲染机制	18	3.3.4 v-bind对象语法绑定行内样式	49
2.2.1 响应式声明渲染机制简介	18		
2.2.2 Vue属性绑定	21		

3.3.5 v-bind 数组语法绑定行内样式	50	5.3.1 父组件向子组件通信	111
3.4 Vue 表单输入绑定	51	5.3.2 子组件向父组件通信	112
3.4.1 指令 v-model	51	5.3.3 任意组件及平行组件通信	116
3.4.2 v-bind 在控件上绑定 value	56	5.4 创建自己的组件	118
3.4.3 表单中的参数特性	63	本章小结	123
3.5 综合案例	64	习题	123
本章小结	67		
习题	67		
第 4 章 Vue 事件处理	68	第 6 章 自定义指令	128
4.1 Vue 事件处理器	68	6.1 自定义指令概述	128
4.1.1 监听事件	68	6.1.1 自定义全局指令	128
4.1.2 方法事件处理器	69	6.1.2 自定义局部指令	130
4.1.3 内联事件处理器	70	6.1.3 案例分析	131
4.2 修饰符	72	6.2 钩子函数	132
4.2.1 事件修饰符	72	6.2.1 钩子函数参数	133
4.2.2 按键修饰符	73	6.2.2 函数简写	136
4.2.3 修饰键	77	6.3 对象字面量	136
4.3 综合案例	77	本章小结	137
本章小结	79	习题	137
习题	79		
第 5 章 Vue 组件	80	第 7 章 过渡与动画	138
5.1 组件的基本使用	80	7.1 transition 组件	138
5.1.1 什么是组件	80	7.2 单元素/组件的过渡	141
5.1.2 组件使用	82	7.2.1 transition 标签结合 CSS 样式实现动画	141
5.1.3 组件中的 data 必须是函数	88	7.2.2 animate.css 结合 transition 实现动画	143
5.2 Vue 组件嵌套	90	7.2.3 钩子函数实现动画	144
5.2.1 组件嵌套	90	7.3 多个元素的过渡	147
5.2.2 使用 props	92	7.4 多个组件的过渡	149
5.2.3 使用 slot 插槽内容分发	101	7.5 综合案例	151
5.2.4 组件实战	107	本章小结	156
5.3 组件通信	111	习题	156

第 8 章 渲染函数	157	第 10 章 使用 webpack	193
8.1 渲染函数概述	157	10.1 webpack 基础	193
8.1.1 从虚拟 DOM 了解 Vue 渲染函数	157	10.2 webpack 的基本配置	196
8.1.2 为什么使用渲染函数	158	10.3 webpack 常用的 Loaders	198
8.1.3 什么是渲染函数	159	10.4 插件	201
8.2 createElement 方法	161	10.5 webpack 常用命令	204
8.2.1 createElement 参数	161	本章小结	204
8.2.2 VNodes 必须唯一	166	习题	204
8.3 使用 JavaScript 代替模板功能	167	第 11 章 综合案例开发	205
8.3.1 v-if 和 v-for	167	11.1 综合案例开发步骤	205
8.3.2 v-model	168	11.1.1 通过 vue-cli 构建工具初始化 项目目录	205
8.3.3 slot 插槽	170	11.1.2 初始化依赖包	206
8.3.4 作用域插槽	171	11.1.3 安装 vue-router 组件	206
本章小结	172	11.1.4 创建 router 对象及配置路由	207
习题	172	11.1.5 在 App.vue 中添加路由 导航	208
第 9 章 Vue 路由 vue-router	173	11.2 解析 Vue 生成文件目录结构	210
9.1 路由安装和使用	173	11.2.1 build 文件夹	210
9.2 传递参数及获取参数	175	11.2.2 config 文件夹	210
9.2.1 使用路由传递参数	175	11.2.3 node_modules 文件夹	211
9.2.2 地址栏传递参数	177	11.2.4 src 文件夹	211
9.3 子路由	178	11.2.5 App.vue 与 main.js	211
9.3.1 创建子路由	178	11.3 vue-resource 插件	212
9.3.2 路由切换组件	180	11.4 实战：单词本	216
9.4 命名视图和导航钩子	183	11.4.1 展示单词本	216
9.4.1 命名视图	183	11.4.2 修改删除单词	217
9.4.2 导航钩子	185	11.4.3 添加单词	221
9.5 元数据及路由匹配	188	本章小结	223
本章小结	191	习题	223
习题	192		

第 12 章 Vue 工程项目实战	224	12.3 组件化开发	235
12.1 项目分析	224	12.4 使用 vue-resource 调用后台接口	247
12.2 工程化项目搭建	224	12.4.1 理解后台接口文档	247
12.2.1 项目准备工作	225	12.4.2 根据文档调用接口	248
12.2.2 项目资源准备	226	12.4.3 具体业务逻辑开发	249
12.2.3 图标字体制作	230	12.5 项目打包	251
12.2.4 项目目录设计	232	本章小结	252
12.2.5 模拟后台数据 (mock 数据)	233	习题	252

在学习 Vue 前，读者应已学习过 HTML、CSS 和 JavaScript 基本知识。本章介绍 Vue 及其特点，讲解 Vue 的下载与生产环境搭建，要求读者掌握 Vue 基本的开发过程，熟悉 MVVM 模式，为以后更好地学习 Vue 打好基础。

本章要点：

- 什么是 Vue 及 Vue 的特点
- Vue 在前端开发中的优势
- Vue 的下载及如何引入并应用
- 实例化 Vue 对象、数据和方法
- 数据挂载到 DOM 页面
- Vue 中的 MVVM 模式

1.1 Vue 简述

Vue.js 是一套响应式系统，前端开发库。其他前端开发库也有很多，比如 jQuery、ExtJS、Angular 等。Vue 自问世以来所受关注度不断提高，在现在的市场上，Vue 是非常流行的 JavaScript 技术开发框架之一。本节将对什么是 Vue.js 及 Vue.js 的特点和优势等内容进行介绍。

1.1.1 什么是 Vue.js

在介绍 Vue 之前，我们先来简单介绍一下它的作者尤雨溪 (Evan You)，以及它的由来。尤雨溪是一位美籍华人，在上海复旦大学附中读完高中后，在美国完成大学学业，本科毕业于 Colgate University，后在 Parsons 设计学院获得 Design & Technology 艺术硕士学位。他是 Vue Technology LLC 创始人，曾经在 Google Creative Lab 就职，参与过多个项目的界面原型研发，后加入 Meteor，参与 Meteor 框架本身的维护和 Meteor Galaxy 平台的交互设计与前端开发。

2014 年 2 月，尤雨溪开源了一个前端开发库 Vue.js。Vue.js 是构建 Web 界面的 JavaScript 库，也是一个通过简洁的 API 提供高效数据绑定和灵活组件的系统。2016 年 9 月 3 日，在南京的 JSConf 上，尤雨溪正式宣布以技术顾问的身份加盟阿里巴巴 Weex 团队，来做 Vue 和 Weex 的 JavaScript runtime 整合，目标是让大家能用 Vue 的语法跨三段。目前，他全职投入 Vue.js 的开发与维护，立志将 Vue.js 打造成与 Angular/React 平起平坐的世界顶级框架。

Vue.js (读音/vju: /, 发音类似于 view) 是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，

并且非常容易学习，也非常容易与其他库或已有项目整合。另外，Vue 完全有能力驱动采用单文件组件和 Vue 生态系统支持的库开发的复杂单页应用。Vue.js 还提供了 MVVM 数据绑定和一个可组合的组件系统，具有简单、灵活的 API，其目标是通过尽可能简单的 API 实现响应式的数据绑定和可组合的视图组件。

我们也可以说 Vue.js 是一套响应式系统 (Reactivity System)。数据模型层 (Model) 只是普通 JavaScript 对象，如图 1-1 所示，“{ }” 代表一个 JavaScript 对象，修改它则更新相应的 HTML 片段 (DOM)，这些 HTML 片段也称为“视图 (view)”。这会让状态管理变得非常简单且直观，可实现数据的双向绑定，所以我们也称之为响应式系统。

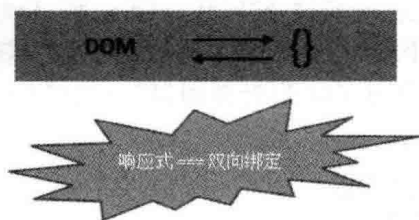


图 1-1 Vue 数据模型与 DOM

1.1.2 为什么使用 Vue.js

我们都知道完整的网页是由 DOM 组合与嵌套形成最基本的视图结构，再加上 CSS 样式的修饰，使用 JavaScript 接受用户的交互请求，并通过事件机制来响应用户交互操作而形成的。我们把最基本的视图结构拿出来，称为视图层。这个被称为视图层的部分就是 Vue 核心库关注的部分。为什么关注它呢？因为一些页面元素非常多。结构庞大的网页如果使用传统开发方式，数据和视图会全部混合在 HTML 中，处理起来十分不易，并且结构之间还存在依赖或依存关系，代码上就会出现更多问题。有过前端开发基础的读者都应当了解过 jQuery，jQuery 给予我们简洁的语法和跨平台的兼容性，极大地简化了 JavaScript 开发人员遍历 HTML 文档、操作 DOM、事件处理等操作。

用过 jQuery 的读者都有体会，开始页面元素不多，有时需要一层层地不断向上寻找父辈元素，如 `$('#xxx').parent().parent()`，但后期修改页面结构，代码可能就会变得臃肿，如 `$('#xxx').parent().parent().parent()`，随着产品升级的速度越来越快，修改变得越来越多，页面中相似的关联和嵌套 DOM 元素多得数不清，而 jQuery 选择器及 DOM 操作本身也存在性能缺失问题，想要修改无从下手。原本轻巧简洁的 jQuery 代码，在产品需求面前变得啰嗦冗长。

但是 Vue.js 解决了这些问题，这些问题将在 Vue 中消失。

1.1.3 Vue.js 的主要特点

Vue.js 是一个优秀的前端界面开发 JavaScript 库，它之所以非常火，是因为有众多突出的特点，其中主要的特点有以下几个。

1. 轻量级的框架

Vue.js 能够自动追踪依赖的模板表达式和计算属性，提供 MVVM 数据绑定和一个可组合的组件系统，具有简单、灵活的 API，使读者更加容易理解，能够更快上手。

2. 双向数据绑定

声明式渲染是数据双向绑定的主要体现，同样也是 Vue.js 的核心，它允许采用简洁的模板语法将数据声明式渲染整合进 DOM。

3. 指令

Vue.js 与页面进行交互，主要就是通过内置指令来完成的，指令的作用是当其表达式的值改变时相应地将某些行为应用到 DOM 上。

4. 组件化

组件 (Component) 是 Vue.js 最强大的功能之一。组件可以扩展 HTML 元素，封装可重用的代码。在 Vue 中，父子组件通过 props 传递通信，从父向子单向传递。子组件与父组件通信，通过触发事件通知父组件改变数据。这样就形成了一个基本的父子通信模式。

在开发中组件和 HTML、JavaScript 等有非常紧密的关系时，可以根据实际的需要自定义组件，使开发变得更加便利，可大量减少代码编写量。组件还支持热重载 (hot-reload)。当我们做了修改时，不会刷新页面，只是对组件本身进行立刻重载，不会影响整个应用当前的状态。CSS 也支持热重载。

5. 客户端路由

Vue-router 是 Vue.js 官方的路由插件，与 Vue.js 深度集成，用于构建单页面应用。Vue 单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来，传统的页面是通过超链接实现页面的切换和跳转的。

6. 状态管理

状态管理实际就是一个单向的数据流，State 驱动 View 的渲染，而用户对 View 进行操作产生 Action，使 State 产生变化，从而使 View 重新渲染，形成一个单独的组件。

最新发行的 Vue 2.X 比 Vue 1.X 更轻、更快，不仅在 Vue 1.X 基础上有很大的提升，相比于其他框架还有很大的性能优势。Vue 学起来比较简单，更容易上手。

1.1.4 Vue.js 的优势

Vue 与其他框架相比有什么优势呢？上面我们已经提到了 jQuery，还有其他的前端框架，如 React、Angular 等。相比较而言，Vue 最为轻量化，而且已经形成了完整的一套生态系统，可以快速迭代更新。作为前端开发人员的首选入门框架，Vue 有很多优势。

(1) Vue.js 可以进行组件化开发，使代码编写量大大减少，读者更加易于理解。

(2) Vue.js 最突出的优势在于可以对数据进行双向绑定（在之后的编写中我们会明显地感觉到这个特点的便捷）。

(3) 使用 Vue.js 编写出来的界面效果本身就是响应式的，这使网页在各种设备上都能显示出非常好看的效果。

(4) 相比传统的页面通过超链接实现页面的切换和跳转，Vue 使用路由不会刷新页面。



Vue 必须在 ES5 版本以上的环境下使用，一些不支持 ES5 的旧浏览器中无法运行 Vue。

1.2 Vue.js 的下载及使用

通过前面的介绍，我们对于什么是 Vue.js 及 Vue 的特点和优势已经有了一个初步的了解，接下来我们来学习 Vue.js 的使用，首先我们需要下载 Vue.js。

1.2.1 Vue.js 下载

下载 Vue.js 可以直接去官网，如图 1-2 所示。

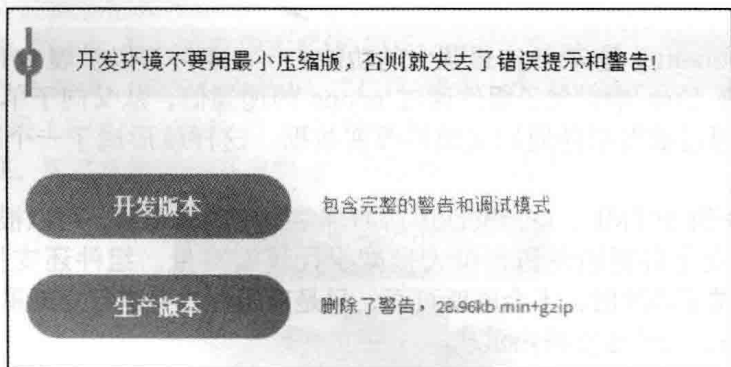


图 1-2 Vue 下载



在开发中最好使用开发版本，在控制台有错误提示信息方便调试程序。

1.2.2 Vue.js 引用

已经下载了 Vue.js，如何把 Vue.js 引入代码中呢？可以使用 `<script>` 标签引入，格式如下：

```
<script src="文件路径/vue.js"></script>
```

也可以使用 #CDN 来引入 Vue.js：

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.5.13/dist/vue.js"></script>
```

这样就可以直接使用 Vue.js 库了。

1.2.3 npm 安装

在用 Vue.js 构建大型应用时推荐使用 npm 安装，npm 可以和 webpack（前端资源加载/打包工具）或 Browserify（使用类似于 node 的 `require()` 的方式来组织浏览器端的 JavaScript 代码，通过预编译让前端 JavaScript 可以直接使用 node npm 安装的一些库）配合使用。Vue.js 也提供配套工具来开发单文件组件。由于 npm 安装速度慢，因此可以使用淘宝的镜像及其命令 `cnpm`（`cnpm` 是淘宝团队开发的一个 npm 国内复制品，可以使用 npm 的所有命令选项，不用登录国外网站即可安装 npm 下的任何软件）来安装 Vue。



如果计算机中 `npm` 命令不是内部命令，需要先安装 `node.js`。

下面是具体安装的过程演示，首先查看 `node.js` 的版本号，再按照步骤安装。

(1) 如图 1-3 所示，查看 `npm` 的版本号并安装 Vue。

1. # 查看版本
2. \$ npm -v

3. # 最新稳定版
4. \$ cnpm install vue



图 1-3 npm 版本及安装 Vue

(2) 如图 1-4 所示, 安装脚手架 vue-cli。vue-cli 脚手架是 Vue.js 提供的一个官方命令行工具, 用于快速搭建大型单页应用。

1. # 命令行工具 Vue.js 提供一个官方命令行工具, 可用于快速搭建大型单页应用。
2. # 全局安装 vue-cli
3. \$ cnpm install --global vue-cli

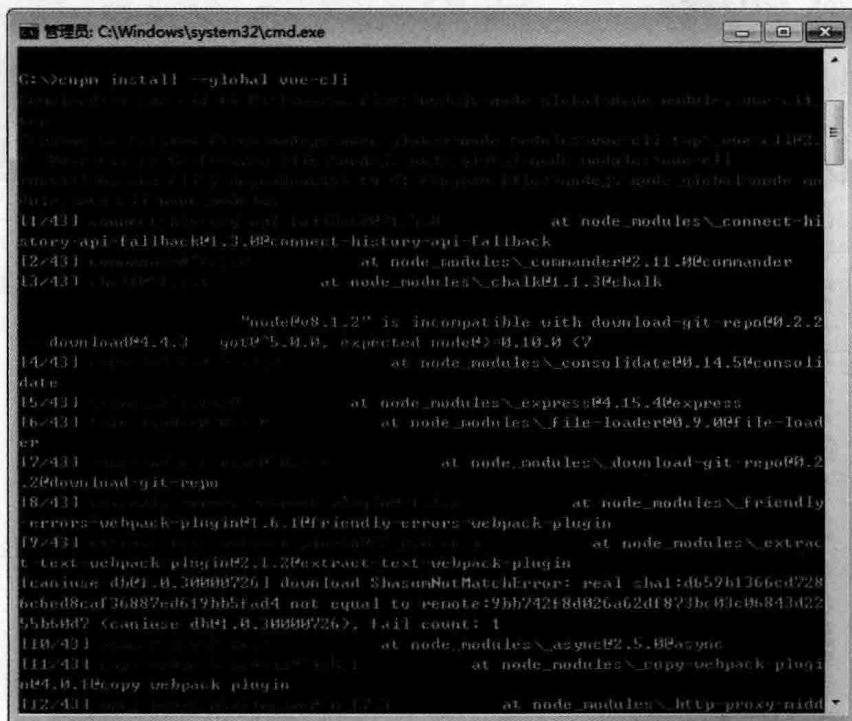


图 1-4 安装脚手架 vue-cli

(3) 创建一个基于 webpack 模板的新项目, 如图 1-5 所示, 创建项目名称为 my-project。

1. # 创建一个基于 webpack 模板的新项目
2. \$ vue init webpack my-project

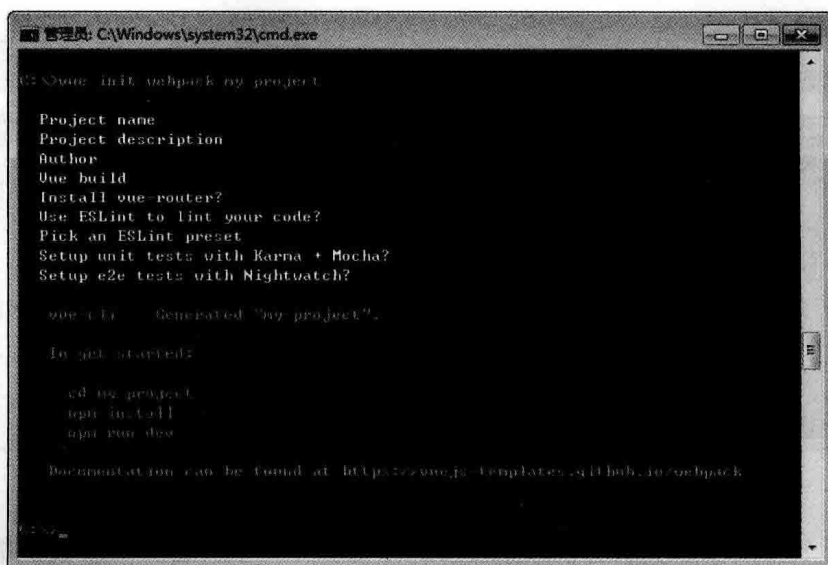


图 1-5 webpack 模板的新项目

(4) 使用 cd 命令进入项目 my-project 中，并安装项目依赖项，如图 1-6 所示。

1. \$ cd my-project
2. \$ npm install

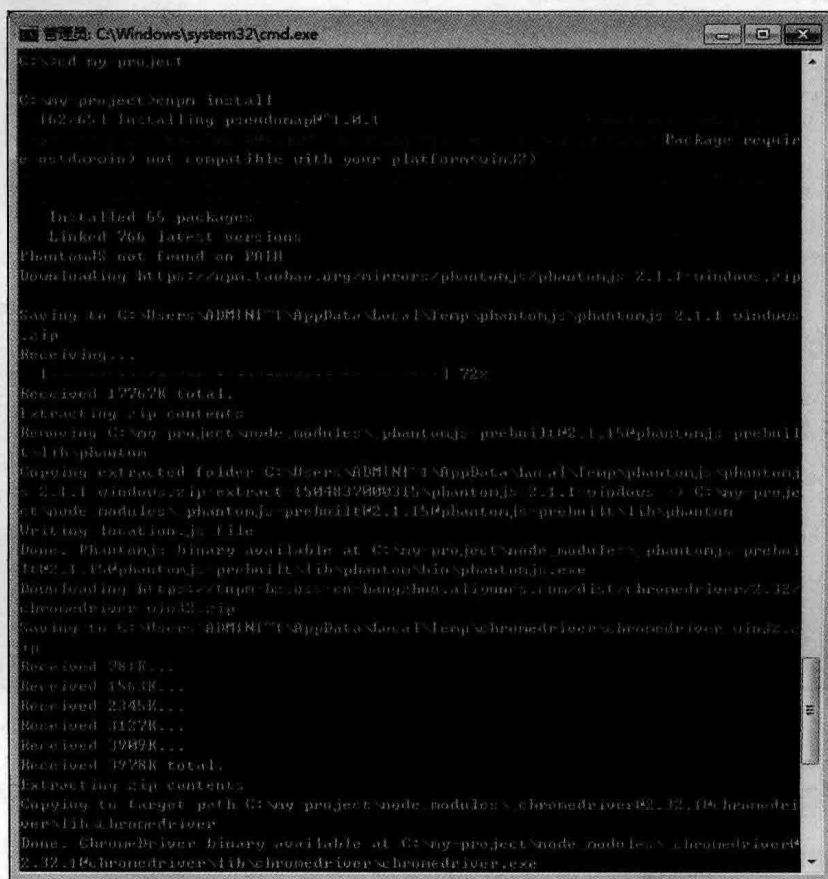


图 1-6 安装模块

(5) 使用 `cnpm run dev` 运行项目，如图 1-7 所示。

```
$ cnpm run dev
```

```

npm
Received 2145K...
Received 1127K...
Received 1989K...
Received 1978K total.
Extracting zip contents
Copying to target path C:\my-project\node_modules\chromedriver\2.32.1\chromedriver\lib\chromedriver
Done. Chromedriver binary available at C:\my-project\node_modules\chromedriver\2.32.1\chromedriver\lib\chromedriver\chromedriver.exe
Run 2 scripts:
  in "my-project"
  unset with --unset flag.
  declared as --
  > but using ancestor("my-project")'s dependency
  <resolved as
  >
  packages
  All packages installed (949 packages installed from npm registry, used 3m, speed 197.54KB/s, json 821(1.59MB), tarball 18.02MB)
C:\my-project>cnpm run dev
> my-project@1.0.0 dev C:\my-project
> node build/dev-server.js
> Starting dev server...
> Listening at http://localhost:8080
  
```

图 1-7 运行程序

成功执行以上命令后访问 `http://localhost:8080/`，输出结果如图 1-8 所示。



图 1-8 Vue 单页面 App

打开项目所在目录 `my-project\src\components` 下的文件 `Hello.vue` 并修改 `msg`: 把“Welcome to Your Vue.js App”改为“欢迎进入斤斗云学堂”，发现什么了？界面没有刷新而数据自动更新了，是不是很神奇！后续课程我们会继续深入学习。

1.2.4 解读 Vue 项目文件目录结构

利用 `node` 和 `npm` 环境，我们可以很快地搭建一个 Vue 环境。搭建完成后，我们可以看到生成的文件夹中包括图 1-9 所示的文件。

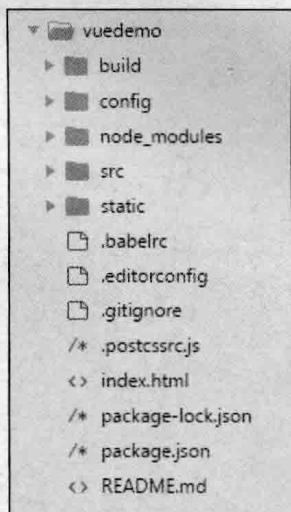


图 1-9 Vue 项目目录结构

下面具体介绍 Vue 项目目录结构。

- (1) `build` 文件夹保存 webpack 的基本配置，`config` 文件夹保存项目基本配置。
- (2) `node_modules` 是 npm 加载的项目依赖的模块。
- (3) `src` 目录是我们要开发的目录，如图 1-10 所示。

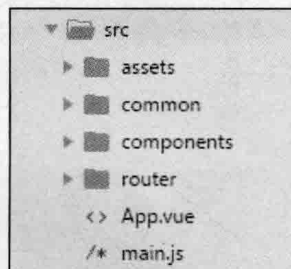


图 1-10 Vue 项目 `src` 目录结构

`assets`: 用来放置图片。

`common`: 用来存放字体文件和通用的样式文件。

`components`: 用来放组件文件。

`App.vue`: 项目入口文件。

- (4) `static` 文件夹用来放置静态资源目录。
- (5) `index.html`: 首页入口文件。
- (6) `package.json`: 项目配置文件。