



胡同江 / 编著

Vue.js

从入门到项目实战

清华大学出版社



Vue.js

从入门到项目实战



清华社官方微信号



扫我有惊喜

ISBN 978-7-302-53919-3



9 787302 539193 >

定价：78.00元

Vue.js 从入门到项目实战

胡同江 编著

清华大学出版社
北 京

内 容 简 介

本书从零基础开始讲解，用实例引导读者深入学习，深入浅出地讲解了Vue框架的各项实战技能。

本书共16章，主要讲解了Vue.js的基本概念，Vue实例和模板语法，计算属性、侦听器和过滤器，内置指令，页面元素样式的绑定，事件处理，双向数据绑定，组件技术，使用webpack打包，项目脚手架vue-cli，前端路由技术，状态管理等。最后列举了4个行业热点项目实训，包括神影视频App、音乐之家App、仿手机QQ页面、仿饿了么App。

本书适合任何想学习Vue框架的人员，无论您是否从事计算机相关行业，无论您是否接触过Vue框架，通过学习本书内容均可快速掌握Vue框架设计的方法和技巧。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Vue.js从入门到项目实战 / 胡同江编著. —北京：清华大学出版社，2019.10 (2020.5重印)
ISBN 978-7-302-53919-3

I. ①V… II. ①胡… III. ①网页制作工具—程序设计 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字 (2019) 第 224350 号

责任编辑：张彦青

封面设计：李 坤

责任校对：李玉茹

责任印制：宋 林

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：21.25 字 数：513千字

版 次：2019年11月第1版 印 次：2020年5月第2次印刷

定 价：78.00元

产品编号：082916-01

前言

为什么要写这样一本书

Vue.js 是当下很火的一个 JavaScript MVVM 库，它是以数据驱动和组件化的思想构建的。Vue.js 提供了更加简洁、更易于理解的 API，能够在很大程度上降低 Web 前端开发的难度，因此深受广大 Web 前端开发人员的喜爱。Vue.js 库功能虽然强大，但是对于初学者来说入门比较困难，因此我们选择了让读者从入门到掌握 Vue.js 作为编写本书的思路，没有基础的读者通过本书学习也没有障碍。本书将项目开发中的技能融入本书的案例中，通过本书的学习，不仅可以掌握 Vue.js 库的使用方法，还可以积累项目开发经验，从而满足企业实际开发的需求。

本书特色

零基础、入门级的讲解

无论您是否从事计算机相关行业，无论您是否接触过 Vue 框架，都能从本书中找到最佳起点。

实用、专业的范例和项目

本书在编排上紧密结合深入学习 Vue 框架技术的过程，从 Vue 框架基本操作开始，逐步带领读者学习 Vue 框架的各种应用技巧，侧重实战技能，使用简单易懂的实际案例进行分析和操作指导，让读者学起来简明轻松，操作起来有章可循。

随时随地学习

本书提供了微课视频，通过手机扫码即可观看，随时随地解决学习中的困惑。

细致入微、贴心提示

本书在讲解过程中，在各章中使用了“注意”“提示”“技巧”等小栏目，使读者在学习过程中能更清楚地了解相关操作、理解相关概念，并轻松掌握各种操作技巧。

超值资源大放送

全程同步教学录像

涵盖本书所有知识点，详细讲解每个实例及项目的过程及技术关键点。比看书更能轻松

地掌握书中所有的网页制作和设计知识，而且扩展的讲解部分可使您得到比书中更多的收获。

超多容量王牌资源

赠送大量王牌资源，包括实例源代码、教学幻灯片、本书精品教学视频、实用网页模板、网页开发必备参考手册、HTML 5 标签速查手册、精选的 JavaScript 实例、CSS 3 属性速查表、JavaScript 函数速查手册、CSS+DIV 布局案例赏析、精彩网站配色方案赏析、网页样式与布局案例赏析、Web 前端工程师常见面试题等。读者可以在清华大学出版社官网搜索本书获取赠送资源。

读者对象

- 没有任何 Vue 框架开发基础的初学者。
- 有一定的 Vue 框架开发基础，想精通前端框架开发的人员。
- 有一定的网页前端设计基础，没有项目经验的人员。
- 大专院校及培训学校的老师和学生。

创作团队

本书由胡同江编著，参加编写的人员还有刘春茂、李艳恩和李佳康。在编写过程中，我们虽竭尽所能将最好的讲解呈现给了读者，但难免有疏漏和不妥之处，敬请读者不吝指正。

编者

目 录

Contents

第1章 Vue.js 简介与安装	001
1.1 Vue 概述.....	001
1.1.1 MVVM 模式.....	002
1.1.2 Vue 是什么.....	002
1.1.3 Vue 有什么不同.....	003
1.2 如何使用 Vue.js.....	005
1.2.1 传统的前端开发模式.....	005
1.2.2 Vue.js 的开发模式.....	005
1.3 安装 Vue.....	006
1.3.1 直接用 <script> 引入.....	006
1.3.2 NPM.....	007
1.3.3 命令行工具.....	007
1.4 第一个 Vue 程序.....	007
1.5 疑难解惑.....	010
第2章 Vue 实例和模板语法	011
2.1 Vue 实例.....	011
2.1.1 创建一个 Vue 实例.....	011
2.1.2 数据与方法.....	012
2.1.3 实例生命周期钩子.....	014
2.1.4 实例化多个对象.....	014
2.2 模板语法.....	018
2.2.1 插值.....	018
2.2.2 指令.....	021
2.2.3 缩写.....	021
2.3 疑难解惑.....	022
第3章 计算属性、侦听器和过滤器	023
3.1 计算属性.....	023
3.2 计算属性与方法.....	024
3.3 侦听属性.....	027
3.3.1 回调值为函数方法.....	027
3.3.2 回调值为对象.....	028
3.4 过滤器.....	030
3.5 疑难解惑.....	033
第4章 内置指令	035
4.1 基本指令.....	035
4.1.1 v-cloak.....	035
4.1.2 v-once.....	036
4.1.3 v-text 与 v-html.....	037
4.1.4 v-bind.....	037
4.1.5 v-on.....	038
4.2 条件渲染.....	040
4.2.1 v-if.....	040
4.2.2 在 <template> 元素上使用 v-if 条件渲染分组.....	042
4.2.3 v-else.....	043
4.2.4 v-else-if.....	044
4.2.5 用 key 管理可复用的元素.....	045
4.2.6 v-show.....	047
4.2.7 v-if 与 v-show 的区别.....	048
4.3 列表渲染.....	050

4.3.1 使用 v-for 指令遍历元素	050	4.3.5 在 <template> 上使用 v-for	057
4.3.2 维护状态	053	4.3.6 v-for 与 v-if 一同使用	058
4.3.3 数组更新检测	053	4.4 自定义指令	060
4.3.4 对象变更检测注意事项	056	4.5 疑难解惑	062

第 5 章 页面元素样式的绑定..... 064

5.1 绑定 HTML 样式	064	5.2 绑定内联样式	069
5.1.1 数组语法	064	5.2.1 对象语法	069
5.1.2 对象语法	066	5.2.2 数组语法	071
5.1.3 在自定义组件上使用 class	068	5.3 疑难解惑	072

第 6 章 事件处理..... 074

6.1 监听事件	074	6.3.5 prevent 修饰符	085
6.2 事件处理方法	075	6.3.6 passive 修饰符	086
6.3 事件修饰符	080	6.4 按键修饰符	087
6.3.1 stop 修饰符	080	6.5 系统修饰键	089
6.3.2 capture 修饰符	082	6.6 案例实战——仿淘宝 Tab 栏切换	090
6.3.3 self 修饰符	083	6.7 疑难解惑	092
6.3.4 once 修饰符	085		

第 7 章 表单输入绑定（双向数据绑定）..... 094

7.1 双向绑定	094	7.3.2 绑定单选按钮	101
7.2 基本用法	095	7.3.3 绑定选择框	102
7.2.1 文本	095	7.4 修饰符	103
7.2.2 多行文本	096	7.4.1 lazy 修饰符	103
7.2.3 复选框	096	7.4.2 number 修饰符	103
7.2.4 单选按钮	097	7.4.3 trim 修饰符	104
7.2.5 选择框	098	7.5 案例实战 1——小游戏破坏瓶子	105
7.3 值绑定	101	7.6 案例实战 2——设计动态表格	107
7.3.1 绑定复选框	101	7.7 疑难解惑	110

第 8 章 组件技术..... 113

8.1 组件是什么	113	8.4 组件中的 props 选项	121
8.2 组件的注册	114	8.5 组件的复用	124
8.2.1 全局注册	114	8.6 组件间的数据通信	126
8.2.2 局部注册	117	8.6.1 父组件向子组件通信	126
8.3 组件中的 data 选项	119	8.6.2 子组件向父组件通信	127

8.7 插槽.....	129	8.7.4 解构插槽.....	137
8.7.1 认识插槽.....	129	8.8 案例实战——设计照片相册.....	139
8.7.2 具名插槽.....	132	8.9 疑难解惑.....	143
8.7.3 作用域插槽.....	135		
第9章 使用 webpack 打包.....	145		
9.1 前端工程化与 webpack.....	145	9.2.2 webpack 的核心概念.....	149
9.2 webpack 基础配置.....	148	9.2.3 完善配置文件.....	152
9.2.1 安装 webpack 与 webpack-dev-server.....	148	9.3 单文件组件与 vue-loader.....	154
		9.4 疑难解惑.....	160
第10章 项目脚手架 vue-cli.....	161		
10.1 脚手架的组件.....	161	10.4.1 使用命令.....	165
10.2 脚手架环境搭建.....	162	10.4.2 使用图形化界面.....	167
10.3 安装脚手架.....	164	10.5 疑难解惑.....	171
10.4 创建项目.....	165		
第11章 前端路由技术.....	172		
11.1 实现 Vue 前端路由控制.....	172	11.3 程式化导航.....	191
11.1.1 前端路由的实现方式.....	172	11.4 组件与 Vue Router 间解耦.....	194
11.1.2 路由实现步骤.....	173	11.4.1 布尔模式解耦.....	194
11.2 命名路由、命名视图和路由传参.....	177	11.4.2 对象模式解耦.....	197
11.2.1 命名路由.....	177	11.4.3 函数模式解耦.....	198
11.2.2 命名视图.....	179	11.5 疑难解惑.....	198
11.2.3 路由传参.....	183		
第12章 状态管理.....	200		
12.1 Vuex 概述.....	200	12.3 在项目中使用时 Vuex.....	205
12.1.1 状态管理模式.....	200	12.3.1 使用脚手架搭建一个项目.....	205
12.1.2 Vuex 的应用场合.....	202	12.3.2 state 对象.....	209
12.2 Vuex 的安装与使用.....	202	12.3.3 getter 对象.....	210
12.2.1 安装 Vuex.....	202	12.3.4 mutation 对象.....	212
12.2.2 Promise 对象.....	203	12.3.5 Action 对象.....	213
12.2.3 使用 Vuex.....	203	12.4 疑难解惑.....	215
第13章 项目实训1——神影视频 App.....	216		
13.1 准备工作.....	216	13.1.1 开发环境.....	216

13.1.2 搭建 Vue 脚手架	216	13.3.3 设计影院页面组件	235
13.2 网站概述	219	13.3.4 设计我的页面组件	238
13.2.1 网站结构	219	13.4 设计项目页面组件及路由配置	239
13.2.2 初始化项目文件	220	13.4.1 电影页面组件及路由	239
13.3 设计项目组件	221	13.4.2 影院页面组件及路由	242
13.3.1 设计头部和底部导航组件	222	13.4.3 我的页面组件及路由	244
13.3.2 设计电影页面组件	224		

第 14 章 项目实训 2——音乐之家 App..... 246

14.1 项目概述	246	14.3.1 store/api.js	252
14.1.1 开发环境	246	14.3.2 store/index.js	253
14.1.2 技术概括	247	14.4 项目组件设计	254
14.1.3 项目结构	247	14.4.1 欢迎组件	255
14.2 入口文件	248	14.4.2 播放组件	256
14.2.1 项目入口页面 (index.html)	248	14.4.3 歌曲信息组件	259
14.2.2 程序入口文件 (main.js)	249	14.4.4 歌曲列表组件	261
14.2.3 组件入口文件 (App.vue)	249	14.4.5 歌曲详情组件	262
14.3 状态管理	252		

第 15 章 项目实训 3——仿手机 QQ 页面..... 266

15.1 项目概述	266	15.4 项目组件及路由	276
15.1.1 开发环境	266	15.4.1 配置路由	276
15.1.2 技术概括	267	15.4.2 顶部导航栏组件	277
15.1.3 项目结构	267	15.4.3 侧边栏导航组件	278
15.2 入口文件	268	15.4.4 搜索组件	280
15.2.1 项目入口页面	268	15.4.5 个人信息页面组件	283
15.2.2 程序入口文件	269	15.4.6 底部 tab 栏组件	286
15.2.3 组件入口文件	269	15.4.7 消息页面组件	288
15.3 状态管理	272	15.4.8 聊天组件	291
15.3.1 action.js	272	15.4.9 朋友页面组件	295
15.3.2 getters.js	273	15.4.10 动态页面组件	298
15.3.3 mutations.js	273	15.5 模拟请求数据	300
15.3.4 store.js	275		

第 16 章 项目实训 4——仿饿了么 App..... 302

16.1 项目概述	302	16.1.2 项目结构	303
16.1.1 开发环境	302	16.2 入口文件	304

16.2.1	项目入口页面.....	304	16.3.4	评论内容组件.....	314
16.2.2	程序入口文件.....	304	16.3.5	商品详情组件.....	317
16.2.3	组件入口文件.....	305	16.3.6	星级组件.....	320
16.3	项目组件.....	306	16.3.7	商品组件.....	321
16.3.1	头部组件.....	306	16.3.8	评论组件.....	324
16.3.2	商品数量控制组件.....	309	16.3.9	商家信息组件.....	326
16.3.3	购物车组件.....	310			



第1章

Vue.js简介与安装

很多人都说 Web 前端开发是程序员中门槛最低的，5 天就可以从入门到精通，你是否有“道理我都懂，就是不知道怎么入手”的困惑？你真的知道如何做好一个前端开发者吗？其实前端开发远不止切图、写 HTML、调样式，前端的世界很大，Node.js、webpack 和各种各样的前端框架都是 Web 前端世界的一部分。

本章主要介绍与 Vue.js（以下简称 Vue）有关的一些概念与技术，并帮助读者了解它们背后相关的工作原理。通过对本章的学习，即使从未接触过 Vue，也可以运用这些知识点快速构建出一个 Vue 应用。

1.1 Vue 概述

说到前端框架，当下比较流行的有 3 个：Vue、React.js（以下简称 React）和 Angular.js（以下简称 Angular）。其中 Vue 以其容易上手的 API、不俗的性能、渐进式的特性和活跃的社区，在三大框架中脱颖而出，截至 2019 年 3 月，Vue 在 github 上的 star 数已经超过了其他两个框架，成为三大框架中最热门的框架。

Vue 被定义成一个用来开发 Web 界面的前端框架，是个轻量级的工具。使用 Vue 可以让 Web 开发变得简单，同时也颠覆了传统前端开发模式。它提供了现代 Web 开发中常见的高级功能，例如：

- 逻辑视图与数据。
- 可复用的组件。
- 前端路由。
- 状态管理。
- 虚拟 DOM（Virtual DOM）。

1.1.1 MVVM 模式

学习 Vue 之前，先来了解一下 MVVM 模式。MVVM 是 Model-View-ViewModel 的简写，即模型 - 视图 - 视图模型。模型指的是后端传递的数据，视图指的是 HTML 页面，视图模型是 MVVM 模式的核心，它是连接 View 和 Model 的桥梁。MVVM 有两个方向：一是将模型转化成视图，即将后端传递的数据转化成所看到的页面，实现的方式是数据绑定。二是将视图转化成模型，即将所看到的页面转化成后端的数据，实现的方式是 DOM 事件监听。如果这两个方向都实现，称之为数据的双向绑定。

在 MVVM 的框架中，视图和模型是不能直接通信的，它们通过 ViewModel 来通信，ViewModel 通常要扮演一个监听者的角色，当数据发生变化时，ViewModel 能够监听到数据的变化，然后通知对应的视图做自动更新；而当用户操作视图时，ViewModel 也能监听到视图的变化，然后通知数据做改动，这实际上就实现了数据的双向绑定。并且 MVVM 中的 View 和 ViewModel 可以互相通信。MVVM 流程图如图 1-1 所示。

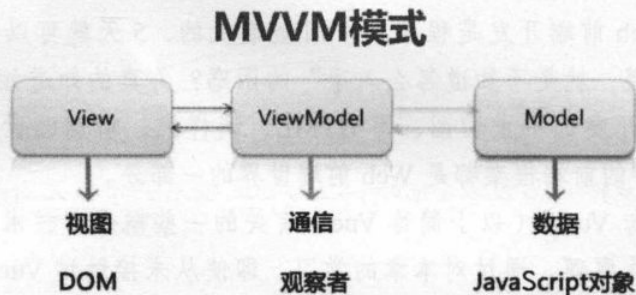


图 1-1 MVVM 流程图

Vue 就是基于 MVVM 模式实现的一套框架。在 Vue 中，Model 指的是 JavaScript 中的数据，例如对象、数组等，View 指的是页面视图，ViewModel 指的是 Vue 实例化对象。

1.1.2 Vue 是什么

在过去的十年里，我们的网页变得更加动态化和强大了，因为有 JavaScript，我们可以把很多传统的服务端代码放到浏览器中，这样就产生了成千上万行的 JavaScript 代码，它们连接了各式各样的 HTML 和 CSS 文件，但缺乏正规的组织形式，这也是为什么越来越多的开发者使用 JavaScript 框架的原因，例如 Vue、Angular 和 React。

Vue 是一款友好的、多用途且高性能的 JavaScript 框架，它能够帮助你创建可维护性和可测试性更强的代码库。Vue 是一个渐进式的 JavaScript 框架，渐进式的意义如下：

- (1) 用户可以一步一步、有阶段性地来使用 Vue，不必一开始就使用所有的东西。
- (2) 如果已经有一个现成的服务端应用，可以将 Vue 作为该应用的一部分嵌入其中，带来更加丰富的交互体验。

(3) 如果希望将更多业务逻辑放到前端来实现，那么 Vue 的核心库及其生态系统也可以满足用户的各种需求。和其他前端框架一样，Vue 允许用户将一个网页分割成可

复用的组件，每个组件都包含属于自己的 HTML、CSS、JavaScript 以用来渲染网页中相应的地方。

(4) 如果用户构建一个大型的应用，可能需要将东西分割成为各自的组件和文件，Vue 有一个命令行工具，使快速初始化一个真实的工程变得非常简单。

可以看出，Vue 的使用可大可小，它都会有相应的方式来整合到用户的项目中。所以说它是一个渐进式的框架。

Vue 本身具有响应式编程和组件化的特点。

响应式：即为保持状态和视图的同步，也被称为数据绑定，声明实例 `new Vue({data:data})` 后自动对 `data` 里面的数据进行了视图上的绑定，修改 `data` 的数据，视图中对应数据也会随之更改。

组件化：Vue 组件化的理念和 React 异曲同工——“一切都是组件”。可以将任意封装好的代码注册成标签，例如：`Vue.component('example',Example)`，可以在模板中以 `<example></example>` 的形式调用。如果组件设计合理，在很大程度上能减少重复开发，而且配合 Vue 的插件 `vue-loader`，可以将一个组件的 CSS、HTML 和 JavaScript 都写在一个文件里，做到模块化的开发。除此之外，Vue 也可以与 `vue-router` 和 `vue-resource` 插件配合起来，以支持路由和异步请求，这样就满足了开发单页面应用的基本条件。

提示



Vue 不支持 IE 8 及以下版本，因为 Vue 使用了 IE 8 无法模拟的 ECMAScript 5 特性，但它支持所有兼容 ECMAScript 5 的浏览器。

1.1.3 Vue 有什么不同

在前端开发中，会遇到动画、交互效果、页面特效等业务，原生的 JavaScript 或 jQuery 库通过操作 DOM 来实现，数据和界面是连接在一起的，例如下面的示例。

在示例中添加一段文本和一个按钮，并为按钮添加一个单击事件，当单击按钮时把文本中的“王老师”更改为“李老师”，“30”更改为“40”。

```
<div>
  <p>大家好,我是<span id="name">王老师,</span></p>
  <p>今年<span id="age">30</span>岁。</p>
  <button id = "updata">修改</button>
</div>
<script>
  $("#updata").click(function(){
    $("#name").text("李老师");
    $("#age").text("40");
  });
</script>
```

在 IE 11 浏览器中运行，效果如图 1-2 所示，单击“修改”按钮时，页面中的内容更改，结果如图 1-3 所示。

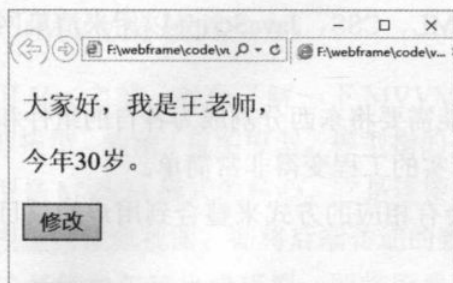


图 1-2 初始化效果

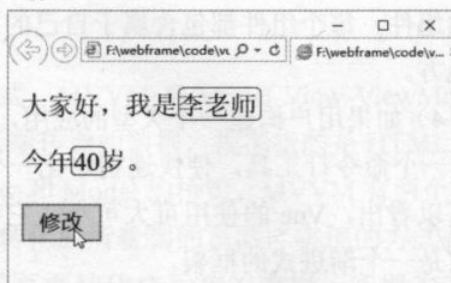


图 1-3 单击“修改”按钮后的效果

Vue 将数据层和视图层完全分离开, 不仅对 API 进行封装, 还提供了一系列的解决方案。这是一个思想的转变, 数据驱动的机制, 主要操作的是数据而不是频繁地操作 DOM (导致页面频繁重绘)。使用 Vue 实现上述示例, 代码如下:

```

<div id="app">
  <p>大家好,我是<span>{{name}}</span></p>
  <p>今年<span>{{age}}</span>岁。</p>
  <button v-on:click="update">修改</button>
</div>
<script>
  new Vue({
    el: '#app',
    data: {
      name: "王老师",
      age: "30"
    },
    methods: {
      update: function() {
        this.name = "李老师";
        this.age = "40";
      }
    }
  })
</script>

```

提示

对于上面的 Vue 代码, 暂时不用理解, 这里只是快速展示 Vue 的写法, 详细的介绍请参考下面“1.4 第一个 Vue 程序”。

以上示例总结如下。

- (1) jQuery 首先要获取到 DOM 对象, 然后对 DOM 对象进行值的修改等操作。
- (2) Vue 首先把值和 JavaScript 对象进行绑定, 然后修改 JavaScript 对象的值, Vue 框架就会自动把 DOM 的值进行更新。
- (3) 可以简单地理解为 Vue 帮我们做了 DOM 操作, 以后使用 Vue 只需要修改对象的值以及做好元素和对象的绑定, Vue 框架就会自动帮我们做好 DOM 的相关操作。
- (4) 如果 DOM 元素跟随 JavaScript 对象值的变化而变化, 叫作单向数据绑定; 如果 JavaScript 对象的值也随着 DOM 元素的值的变化而变化, 叫作双向数据绑定。

1.2 如何使用 Vue.js

Vue 是当下流行的前端框架之一，在正式开始学习它前，我们先对传统前端开发模式和 Vue 的开发模式做一个对比，以此了解 Vue 产生的背景和 Vue 核心思想。

1.2.1 传统的前端开发模式

传统开发模式可称为硬代码开发，数据、展现和逻辑都混在一起，彼此相互混杂，整体看起来非常混乱，它有以下缺点。

(1) 由于展现、数据和逻辑都混合到一起，从而代码的可读性就会很差，很难完成知识的转移和交付。

(2) 界面变更修改复杂，无法快速调试，问题出来了，无法快速定位问题所在。

(3) 维护复杂，容易在修复中出现新 bug。

(4) 数据处理功能单一，若出现排序、筛选等工作，需要重新编写代码。

前端技术在近几年发展迅速，如今的前端开发已不再是 10 年前写个 HTML 和 CSS 那样简单了，新的概念层出不穷，例如 ECMAScript 6、Node.js、NPM、前端工程化等。这些新东西在不断优化我们的开发模式，改变我们的编程思想。

随着项目的扩大和时间的推移，出现了更复杂的业务场景，例如组件解耦、SPA（单页面应用）等。为了提升开发效率，降低维护成本，传统的前端开发模式已不能完全满足我们的需求，这时就出现了 Angular、React 以及我们要介绍的 Vue。

1.2.2 Vue.js 的开发模式

Vue 是基于 MVVM 模式实现的一套框架，MVVM 模式分离视图（View）和数据（Model），通过自动化脚本实现自动化关联，ViewModel 搭起了视图与数据的桥梁，同时在 ViewModel 里进行交互及逻辑处理。可以简单地理解为，View 就是 HTML、DOM，数据 Model 就是要处理的 Json 数据。这种模式有以下优势。

(1) 低耦合。将 View 和 Model 进行分离，两者中其中一方变更时，另一方不会受到影响。

(2) 重用性。无论是 View、ViewModel 还是 Model，三者都可以进行重用，提高了开发效率。

(3) HTML 模板化。修改模板不影响逻辑和数据，模板可直接调试。

(4) 数据自动处理。Model 实现了标准的数据处理封装，例如排序、筛选等。

(5) 双向绑定。通过 DOM 和 Model 双向绑定使数据更新自动化，缩短了开发时间。

1.3

安装 Vue

在学习 Vue 之前，先来学习一下如何安装它。

1.3.1 直接用 `<script>` 引入

直接使用 `<script>` 标签引入有两种方式，一种是官网下载独立的版本，另一种是使用 CDN 的方式。本书前面内容都是使用独立的版本来进行介绍，推荐读者也用这种方式；而本书后面的项目，是使用 Vue 的脚手架进行创建。

1. 独立的版本

在官网 <https://cn.vuejs.org/> 下载最新稳定版本：2.6.10，使用 `<script>` 标签引入到页面中，这时 Vue 会被注册为一个全局变量。

官网提供了两个版本，说明如图 1-4 所示。根据需要下载相应的版本，本书使用的是开发版本 2.6.10。

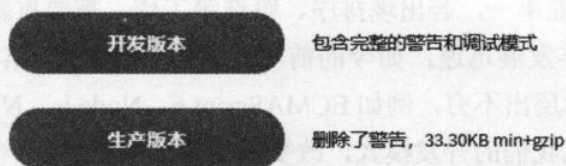


图 1-4 Vue 的下载版本

注意



在开发环境下不要使用生产版本，不然就失去了所有常见错误相关的警告。

下载完成后直接引用：

```
<script src="vue.js"></script>
```

2. CDN 方式

对于制作原型或学习，可以使用最新版本：

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

对于生产环境，推荐链接到一个明确的版本号和构建文件，以避免新版本产生的不可预期的问题，例如：

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.10/dist/vue.js"></script>
```