前言
PREFACE

Vue.js 最佳学习线路

本书以 Vue.js 的最佳学习模式来安排内容结构，第 1～3 篇可使读者掌握 Vue 的基础知识、Vue 的核心应用、Vue 的核心技术等知识，第 4 篇可使读者拥有多个行业项目开发经验。

本书内容

全书分为 4 篇，共 18 章。

第 1 篇（第 1～8 章）为基础知识，本篇主要讲解 Vue.js 的基本知识、简单 Vue 实例的创建等内容，为后面更加深入地学习做铺垫、为使用 Vue.js 前端框架开发项目奠定基础。通过本篇内容的学习，读者可以了解 Vue.js 基本简介、Vue 简单实例创建、指令、基本特性、数据及事件绑定、过滤器、Vue 的动画和过渡等内容。

第 2 篇（第 9～13 章）为核心应用，本篇将介绍 Vue 核心应用技术的使用，包括如何使用 Vue 组件、常用插件、实例方法、Render 函数，以及在学习过程中可能出现的一些问题，包括安装错误、运行错误和你问我答等内容。通过本篇的学习，读者将对 Vue 有深刻的理解，进行深入的学习后，编程能力会有进一步的提高。

第 3 篇（第 14～15 章）为核心技术，本篇介绍 Vue 中常见的状态管理 Vuex，并且结合前面内容介绍 Vue 工程实例等知识内容，还将结合案例示范学习 Vue 中 webpack 开发中的打包、介绍 Vue 中的目录结构等知识内容，为编写和研发项目奠定基础。

第 4 篇（第 16～18 章）为项目实践，本篇融会贯通前面所学的编程知识、技能及开发技巧来开发实践项目。项目包括订餐管理系统、网上图书销售系统及仿网易云音乐系统等。通过本篇的学习，读者将对前端 Vue 框架在实际项目开发中的应用有一个深切的体会，为日后进行软件项目管理及实战开发积累经验。

全书不仅融入了笔者丰富的工作经验和多年的使用心得，还提供了大量来自工作现场的实例，具有较强的实用性和可操作性。读者系统学习后可以掌握 Vue 前端框架的基础知识，拥有全面编写框架的编程能力、优良的团队协同技能和丰富的项目实战经验。编写本书的目标就是让框架初学者快速成长为合格的中级程序员，通过演练积累项目开发经验和团队合作技能，在未来的职场中获取一个较高的起点，并能迅速融入软件开发团队中。

本书特色

1. 结构科学，自学更易

本书在内容组织和范例设计中充分考虑到中级学者的特点，由浅入深，循序渐进，无论读者是否接触过框架，都能从本书中找到最佳的起点。

2. 视频讲解，细致透彻

为降低学习难度，提高学习效率，本书录制了同步微视频（模拟培训班模式）。通过视频讲解，除了能轻松学会专业知识外，还能获取老师的软件开发经验，使学习变得轻松有效。

3. 超多、实用、专业的范例和实践项目

本书结合实际工作中的应用范例逐一讲解 Vue 前端框架的各种知识和技术，在项目实践篇中更以 3 个项目实践来总结前 15 章介绍的知识和技能，使读者在实践中掌握知识、轻松拥有项目开发经验。

4. 随时检测自己的学习成果

每章首页中均提供了“本章概述”和“本章要点”，以指导读者重点学习及学后检查；章后的“就业面试技巧与解析”均根据当前最新求职面试（笔试）题精选而成，读者可以随时检测自己的学习成果，做到融会贯通。

5. 专业创作团队和技术支持

本书由聚慕课教育研发中心编著并提供在线服务。读者在学习过程中遇到任何问题，可加入图书读者服务（技术支持）QQ 群（529669132）进行提问，笔者和资深程序员将为读者在线答疑。

本书附赠超值王牌资源库

本书附赠了极为丰富超值的王牌资源库，具体内容如下。

（1）王牌资源 1：随赠本书“配套学习与教学”资源库，提升读者的学习效率。

- 全书同步 180 节教学微视频录像（支持扫描二维码观看），总时长 12 学时；
- 全书 3 个大型项目案例以及全部范例源代码；
- 本书配套上机实训指导手册，本书学习、授课与教学 PPT 课件。

（2）王牌资源 2：随赠“职业成长”资源库，突破读者职业规划与发展弊端与瓶颈。

- 求职资源库：100 套求职简历模板库、600 套毕业答辩与 80 套学术开题报告 PPT 模板库；
- 面试资源库：程序员面试技巧、200 道求职常见面试（笔试）真题与解析；
- 职业资源库：100 套岗位竞聘模板、程序员职业规划手册、开发经验及技巧集、软件工程师技能手册。

（3）王牌资源 3：随赠“Vue.js 开发魔典”资源库，拓展读者学习本书的深度和广度。

- 案例资源库：80 个实例及源码注释；
- 程序员测试资源库：计算机应用测试题库、编程基础测试题库、编程逻辑思维测试题库、编程英语水平测试题库；
- 软件开发文档模板库：10 套八大行业软件开发文档模板库等；
- 电子书资源库：Vue.js 速查手册、Vue.js API 速查手册、Vue.js 库速查手册，包列表速查手册、Vue.js 常见错误及解决方案、Vue.js 开发经验及技巧大汇总等。

（4）王牌资源 4：编程代码优化纠错器。

- 本纠错器能让软件开发更加便捷和轻松，无须安装配置复杂的软件运行环境即可轻松运行程序代码。
- 本纠错器能一键格式化，让凌乱的程序代码更加规整美观。
- 本纠错器能对代码精准纠错，让程序查错不在难。

上述资源获取及使用

注意：由于本书不配送光盘，书中所用及上述资源均需借助网络下载才能使用。

1. 资源获取

采用以下任意途径，均可获取本书所附赠的超值王牌资源库。

（1）加入本书微信公众号“聚慕课 jumooc”，下载资源或者咨询关于本书的任何问题。

（2）加入本书图书读者服务（技术支持）QQ 群（529669132），读者可以打开群“文件”中对应的 Word 文件，获取网络下载地址和密码。

2. 使用资源

读者可通过电脑/平板 App 端、微信端学习和使用本书微视频及资源。

本书适合哪些读者阅读

本书非常适合以下人员阅读。

- 没有任何前端 Vue 框架基础的初学者。
- 有一定的前端 Vue 框架开发基础，想精通编程的人员。
- 有一定的前端 Vue 框架开发基础，缺乏项目实践经验的人员。
- 正在进行软件专业相关毕业设计的学生。
- 大、中专院校及培训学校的老师和学生。

创作团队

本书由聚慕课教育研发中心组织编写，李良任主编，刘凯燕、李存永任副主编，参与本书编写的人员还有陈梦、裴垚等。

在编写过程中，我们尽己所能将最好的讲解呈现给读者，但也难免有疏漏和不妥之处，敬请读者不吝指正。

编著者

目录

CONTENTS

第 1 篇　基础知识篇

第 2 篇　核心应用篇

第 3 篇 核心技术篇

第 4 篇 项目实践篇

第 1 篇

基础知识篇

本篇主要讲解 Vue.js 的基本知识、简单 Vue 实例的创建等内容，为后面更加深入地学习做铺垫、为使用 Vue.js 前端框架开发项目奠定基础。通过本篇内容的学习，读者可以了解 Vue.js 基本简介、Vue 简单实例创建、指令、基本特性、数据及事件绑定、过滤器、Vue 的动画和过渡等内容。

- **第 1 章**　Vue.js 基本简介
- **第 2 章**　创建 Vue.js 简单实例
- **第 3 章**　Vue.js 指令
- **第 4 章**　Vue.js 基本特性
- **第 5 章**　Vue 数据及事件绑定
- **第 6 章**　Vue.js 过滤器
- **第 7 章**　Vue.js 过渡
- **第 8 章**　Vue.js 动画

第1章

Vue.js 基本简介

本章概述

本章主要讲解 Vue.js 的基本知识、Vue.js 的发展历程、使用的开发软件等内容，为使用 Vue.js 前端框架开发项目奠定基础。通过本章内容的学习，读者可以了解 Vue.js 的基本知识及发展历程，还可以了解 Vue.js 的模式及它和其他流行前端框架之间的对比等。

本章要点

- Vue.js 基本介绍。
- Vue.js 框架的发展历程。
- Vue.js 的模式介绍。
- Vue.js 与 Angular 对比。
- Vue.js 与 React 对比。
- Vue.js 的兼容性。

1.1 前端框架的发展历程

我们都知道，三个非常受欢迎的前端框架 Vue、Angular、React 已经逐渐应用到各个项目和实际的应用中，它们都是 MVVM 数据驱动框架的一种。前端静态页面 HTML、JavaScript、Ajax、Node.js 等的问世，都对前端框架技术有着重大的影响。

1.1.1 前端静态页面走向动态页面的转变

前端主要是针对浏览器进行开发的，代码在浏览器中运行。想要学习前端的基础还需从学习 HTML 开始。

1991 年出现了世界上第一个网页，当时的 HTML 代码如下：

```
<HEADER>
<TITLE>The World Wide Web project</TITLE>
```

```
<NEXTID N="55">
</HEADER>
<BODY>
<H1>World Wide Web</H1>
The World Wide Web(W3) is a wide-area
<A NAME=0 HREF="WhatIs.html">hypermedia</A>
```

从上面的 HTML 源码中可以看到代码标签多，且格式没有明确规范，对于规范书写存在着问题。在这种情况下，Tim Berners Lee 创建了 W3C 标准机构，使得 HTML 的代码有一定的规范，但是在网页设计方面还是存在局限问题，不满足当时的需求。

在这种情况下 Sun 公司编写了 Java 小程序（Applet），可以在页面中实现酷炫的动态效果，大大增加了页面的美观效果。紧接着网景公司为了适应发展和需求花了两周时间开发出 JavaScript 语言，就是我们经常说的 JS 脚本语言。

HTML 语言和 JavaScript 的诞生使得网页之间可以进行更好的交互，但是出现了页面规划不整齐的情况，而且代码量大，代码利用率低。后来 Tim 的朋友发布了 CSS，至此前端三大核心出现。

在前端技术不断发展的同时，也出现了一系列的问题，比较繁杂的就是前端页面的浏览器不兼容问题。同一个 DOM 操作可能会需要写很多适配代码来兼容不同浏览器，于是比较便捷的语言 jQuery 诞生了，一套代码可以多端运行，使得代码的开发更加便捷。

1.1.2 程序后端走向前端的转变

后端主要针对服务器的开发，代码在服务器中运行。在前端发展的早期，网页的开发主要是由后端来主导的，前端可以对 DOM 进行操作。

后端的开发我们一般也称为服务器端开发。开发的数据不会对用户显示，主要是负责前端的请求，从而进行逻辑处理和数据的交互。例如上班的打卡信息，后端进行逻辑判断，是否在规定的时间、规定的地点进行打卡，若符合则将打卡的数据信息存储到数据库。

简单来说，后端负责数据，前端负责其他工作，这种分工模式使得开发更加清晰也更加高效。随着基础设置的不断完善以及代码封装层级的不断提高，使得前端能够完成的事越来越多，这是技术积累的必然结果。

前后端分离的好处是前端关注页面展现，后端关注业务逻辑，分工明确，职责清晰，前端工程师和后端工程师并行工作，大大提高了开发效率。

1.2 Vue.js 介绍

随着 Vue.js 不断地完善，慢慢地适应了市场的需求，它深得开发者的喜爱。Vue.js 建立于 Angular 和 React 的基础之上，保留了 Angular 和 React 的优点并强化了自身的独特之处，这保证了 Vue.js 足够的美好来吸引 JS 开发者的“胃口”。从 JS 前端框架的市场占有率和商业应用上讲，React 仍占据很大的市场。但是毫无疑问，Vue.js 不会消失，它正在持续地、一步步地被大家认可并付诸实践。事实上，据 StateOfJS 的调查结果显示，使用过 Vue.js 并将再次使用的开发者数量占比从 2017 年的 19.6%上升到 2018 年的 28.8%。在同一份调查的“用户满意度最高的前端框架”这一项上，Vue.js 获得了 91.2%的满意度。

1.2.1 Vue.js 是什么

Vue.js 是一款用于构建用户界面的渐进式框架。与其他大型框架不同的是，Vue 被设计为自下向上逐层应用。Vue.js 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另外，当与现代化的工具链及各种支持类库结合使用时，Vue.js 也完全能够为复杂的单页应用提供驱动。Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。

Vue.js 是用于构建交互式的 Web 界面的库，它提供 MVVM 数据绑定和一个可组合的组件系统，具有简单、灵活的 API。从技术上讲，Vue.js 集中在 MVVM 模式上的视图模型层（ViewModel），并通过双向数据绑定连接视图（View）和模型（Model），实际的 DOM 操作和输出格式则被抽象出来构成指令和过滤器。相比其他的 MVVM 模式框架，Vue.js 更容易上手，可以通过简单而灵活的 API 创建由数据驱动的 UI 组件。

1.2.2 Vue.js 发展历程

Vue.js 正式发布于 2014 年 2 月，从脚手架、组件、插件，到编辑器工具、浏览器插件等，基本涵盖了 Vue.js 从开发到测试等多个环节的工具。

Vue.js 的发展历程如下。

（1）2013 年 12 月 24 日，发布 v0.7.0。

（2）2014 年 1 月 27 日，发布 v0.8.0。

（3）2014 年 2 月 25 日，发布 v0.9.0。

（4）2014 年 3 月 24 日，发布 v0.10.0。

（5）2015 年 10 月 27 日，正式发布 v1.0.0。

（6）2016 年 4 月 27 日，发布 v2.0 的 Preview 版本。

（7）2017 年第一个发布的 Vue.js 为 v2.1.9，最后一个发布的 Vue.js 为 v2.5.13。

（8）2019 年发布的 Vue.js 为 v2.6.10，是比较稳定的版本。

1.3 Vue.js 中的开发模式

无论是前端还是后端开发，都有一定的开发模式。下面将介绍 MVC 模式（这种模式在 Java 后端开发中很常见）、MVP 模式及 Vue.js 中常见的 MVVM 模式。

1.3.1 MVC 模式介绍

MVC 的英文全称是 Model View Controller，它是一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特地发展起来，用于映射传统的输入、处理和输出功能在一个图形化用户界面的逻辑结构中。MVC 开始是存在于桌面程序中的，M 是指业务模型，V 是指用户界面，C 则是指控制器。使用 MVC 的目的是将 M 和 V 的实现代码分离，从而使同一个程序可以使用不同的表现形式。例如一批统计数据可以分别用柱状图、饼图来表示。C 存在的目的则是确保 M 和 V 的同步，一旦 M 改变，V 应该同步更新。

模型-视图-控制器（MVC）模式是 Xerox PARC（施乐帕克研究中心）在 20 世纪 80 年代为编程语言 Smalltalk-80 发明的一种软件设计模式，已被广泛使用。后来被推荐为 Oracle 旗下 Sun 公司 Java EE 平台的设计模式，并且受到越来越多使用 ColdFusion 和 PHP 的开发者的欢迎。MVC 模式也存在一定的优点和缺点。下面详细解析 MVC。

（1）模型：模型表示企业数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。例如，可能用像 EJBs 和 ColdFusion Components 这样的构件对象来处理数据库。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型可以为多个视图提供数据。由于应用于模型的代码只需编写一次就可以被多个视图重用，因此减少了代码的重复性。

（2）视图：视图是用户能看到并与其交互的界面。对以前的 Web 应用程序来说，视图就是由 HTML 元素组成的界面；在现今的 Web 应用程序中，HTML 依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括 Adobe Flash 和像 XHTML、XML/XSL、WML 等一些标识语言与 Web Services。MVC 的优点是，它可以为应用程序处理多种不同的视图，而在视图中其实没有真正的处理发生。作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

（3）控制器：控制器接收用户的输入并调用模型和视图去完成用户的需求，所以当单击 Web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何内容和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后确定用哪个视图来显示返回的数据。

1.3.2 MVP 模式介绍

MVP 的英文全称为 Model View Presenter，它是从经典的 MVC 模式演变而来的。它们的基本思想有相通的地方：Controller/Presenter 负责逻辑的处理，Model 提供数据，View 负责显示。MVP 从 MVC 演变而来，通过表示器将视图与模型巧妙地分开。在该模式中，视图通常由表示器初始化，它负责呈现用户界面（UI），并接收用户所发出的命令，但不对用户的输入做任何逻辑处理，而仅仅是将用户输入转发给表示器。通常每一个视图对应一个表示器，但是也可能一个拥有较复杂业务逻辑的视图会对应多个表示器，每个表示器完成该视图的一部分业务处理工作，降低了单个表示器的复杂程度；一个表示器也能被多个有着相同业务需求的视图复用，增加单个表示器的复用度。表示器包含大多数表示逻辑，用以处理视图，与模型交互以获取或更新数据等。模型描述了系统的处理逻辑，但对于表示器和视图一无所知。

1. MVP 模式的优点

MVP 模式的优点体现在以下三个方面。

（1）View 与 Model 完全隔离。Model 和 View 之间具有良好解耦性的设计，这就意味着，如果 Model 或 View 中的一方发生变化，只要交互接口不发生变化，另一方就无须对上述变化做出相应的变化，这使得 Model 层的业务逻辑具有很好的灵活性和可重用性。

（2）Presenter 与 View 的具体实现技术无关。也就是说，采用诸如 Windows 表单、WPF（Windows Presentation Foundation）框架、Web 表单等用户界面构建技术中的任意一种来实现 View 层，都无须改变系统的其他部分。甚至为了使 B/S、C/S 部署架构能够被同时支持，应用程序可以用同一个 Model 层适配多种技术构建的 View 层。

（3）可以进行 View 的模拟测试。由于 View 和 Model 之间的紧耦合，在 Model 和 View 同时开发完成前对其中一方进行测试是不可能的。出于同样的原因，对 View 或 Model 进行单元测试很困难。MVP 模式解决了上述所有的问题。在 MVP 模式中，View 和 Model 之间没有直接依赖，开发者能够借助模拟对象注

入测试两者中的任意一方。

2. MVP 模式与 MVC 模式的区别

MVP 模式示意图如图 1-1 所示。作为一种新的模式，MVP 与 MVC 有着一个重大的区别：在 MVP 中 View 并不直接使用 Model，它们之间的通信是通过 Controller 来进行的，所有的交互都发生在 Controller 内部；而在 MVC 中 View 会直接从 Model 中读取数据，而不是通过 Controller。在 MVC 中，View 是可以直接访问 Model 的。View 中会包含 Model 信息，不可避免地还要包括一些业务逻辑。在 MVC 模式中，更关注 Model 的不变，而同时有多个对 Model 的不同显示及 View。所以在 MVC 模式中，Model 不依赖于 View，但 View 是依赖于 Model 的。不仅如此，因为有一些业务逻辑在 View 中实现，导致要更改 View 也是比较困难的，至少那些业务逻辑是无法重用的，代码复用率低。

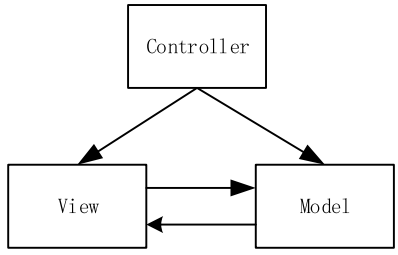

图 1-1　MVP 模式

1.3.3　MVVM 模式介绍

MVVM 是 Model View ViewModel 的简写，它本质上就是 MVC 模式的改进版，目的是将其中 View 的状态及行为抽象化，将视图 UI 和业务逻辑分开。ViewModel 可以做这些事情，它可以取出 Model 的数据，同时帮助处理 View 中由于需要展示内容而涉及的业务逻辑。如今越来越多的新技术，例如 Silverlight、音频、视频、3D、动画等技术的发展导致了软件 UI 层更加细节化、可定制化。同时，在技术层面，WPF 带来了如 Binding、Dependency Property、Routed Events、Command、DataTemplate、ControlTemplate 等新特性。MVVM 模式便是由 MVP 模式与 WPF 结合的应用方式发展演变过来的一种新型架构模式。它立足于原有 MVP 模式并把 WPF 的新特性纳入进去，以应对客户日益复杂的需求变化。

1. MVVM 模式的优点

MVVM 模式和 MVC 模式一样，主要目的是分离视图（View）和模型（Model），以下是 MVVM 模式的优点。

（1）低耦合。View 可以独立于 Model 变化和修改，一个 ViewModel 可以绑定到不同的 View 上，当 View 变化时 Model 可以不变，当 Model 变化时 View 也可以不变。

（2）可重用性。可以将一些视图逻辑放在一个 ViewModel 中，让很多 View 重用这段视图逻辑。

（3）独立开发。开发者可以专注于业务逻辑和数据的开发（ViewModel），设计人员可以专注于页面设计，使用 Expression Blend 工具可以很容易地设计界面并生成 XAML 代码。

（4）可测试。界面素来是比较难测试的，而基于 MVVM 模式，测试可以针对 ViewModel 来写。

2. MVVM 模式的组成部分

（1）模型：模型是指代表真实状态内容的领域模型（面向对象），或指代表内容的数据访问层（以数据为中心）。

（2）视图：就像在 MVC 和 MVP 模式中一样，视图是用户在屏幕上看到的结构、布局和外观。

（3）视图模型：视图模型是暴露公共属性和命令的视图抽象。MVVM 模式没有 MVC 模式的控制器，也没有 MVP 模式的 Presenter，有的是一个绑定器。在视图模型中，绑定器在视图和数据绑定器之间进行通信。

在 Microsoft 解决方案中，绑定器是一种名为 XAML 的标记语言。绑定器使开发者免于被迫编写样板式逻辑来同步视图模型和视图。声明性数据和命令绑定隐含在 MVVM 模式中，声明性数据绑定技术的出现是实现该模式的一个关键因素。

1.4 Vue.js 与其他框架比较

Vue.js 是一款友好的、多用途且高性能的 JavaScript 框架，它能够创建可维护性和可测试性更强的代码库。Vue.js 允许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。下面将介绍 Vue.js 与其他两个流行框架 Angular 及 React 的比较。

1.4.1 Vue.js 与 Angular 的比较

Vue.js 可以说是开源的 JavaScript 框架，可以帮助开发者构建出美观的 Web 界面。当和其他开发工具配合使用时，Vue.js 的优势会大大加强。如今，已有许多开发者使用 Vue.js 进行开发。那么 Vue.js 和 Angular 有什么区别呢？下面我们会对这两种框架进行介绍和深度对比。

1. Vue.js 框架

Vue.js 前端框架是由 Google 公司时任员工 Evan You 开发的，于 2014 年发布。许多开发者都大力推荐并使用 Vue.js 进行开发，因为 Vue.js 比较容易学习和应用。如果拥有深厚的 HTML、CSS 和 JavaScript 基础，那么学习 Vue.js 只需几个小时。Vue.js 对于开发者最有吸引力的地方就是它新颖、轻便，且复杂性比较低。Vue.js 不但非常灵活、简单，而且功能非常强大，同时还提供了双向数据绑定功能，就像 Angular 和 React 的虚拟 DOM 功能一样。Vue.js 可以帮助开发者以任何想要的方式来构建应用程序，而 Angular 做不到这一点。Vue.js 是一个多样化的 JavaScript 框架。作为一个跨平台、高度进步的框架，Vue.js 成了许多需要创建单页应用程序开发者的首选。在开发 Web 应用程序的典型 MVC 体系结构中，Vue.js 充当了 View，意味着可以让开发者看到数据的显示部分。

下面总结 Vue.js 的其他优势功能。

（1）容易使用。如果开发者已掌握其他前端框架的一些知识，那么学习 Vue.js 相对较为简单，因为 Vue.js 的核心库专注于 View 层，可以轻松地将其与第三方库进行整合并与现有项目一起使用。

（2）学习曲线很低。熟悉 HTML 的开发者会发现 Vue.js 的学习曲线很低，同时对于经验较少的开发者和初学者来说，也能够快速地学习和理解 Vue.js。

（3）轻便。由于 Vue.js 主要关注 ViewModel 或双向数据绑定，因此 Vue.js 很轻便。此外，Vue.js 提供了简单、易懂的学习文档。将 Vue.js 用作 View 层，意味着开发者可以将它用作页面中的亮点功能。比起全面的 SPA，Vue.js 为开发者提供了更好的选择。

（4）虚拟 DOM。由于 Vue.js 是基于 Snabbdom 库的轻量级虚拟 DOM 实现，因此 Vue.js 的性能有些许提升。开发者可以直接进行更新，这是虚拟 DOM 的主要新功能之一。当需要在实际的 DOM 中进行更改时，只需执行一次这样的更新功能。