

Vue.js 作者 尤雨溪 作推荐序

Vue.js in Action

Vue.js 2



· 梁灏 编著 ·

# Vue.js

## 实战



清华大学出版社



梁 灏

网名 Aresn，基于 Vue.js 的开源 UI 组件库 iView 的作者。目前在大数据公司 TalkingData 负责可视化基础架构。创办了程序员社区 TalkingCoder。



· 梁灏 编著 ·

# Vue.js

## 实战

清华大学出版社  
· 北京

## 内 容 简 介

本书以 Vue.js 2 为基础, 以项目实战的方式来引导读者渐进式学习 Vue.js。本书分为基础篇、进阶篇和实战篇三部分。基础篇主要是对 Vue.js 核心功能的介绍; 进阶篇主要讲解前端工程化 Vue.js 的组件化、插件的使用; 实战篇着重开发了两个完整的示例, 所涉及的内容涵盖 Vue.js 绝大部分 API。通过阅读本书, 读者能够掌握 Vue.js 框架主要 API 的使用方法、自定义指令、组件开发、单文件组件、Render 函数、使用 webpack 开发可复用的单页面富应用等。

本书示例丰富、侧重实战, 适用于刚接触或即将接触 Vue.js 的开发者, 也适用于对 Vue.js 有过开发经验, 但需要进一步提升的开发者。

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 实战/梁灏编著. —北京: 清华大学出版社, 2017 (2017.11重印)  
ISBN 978-7-302-48492-9

I. ①V… II. ①梁… 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 (2017) 第 230264 号

责任编辑: 王金柱  
封面设计: 王 翔  
责任校对: 闫秀华  
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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 190mm×260mm

印 张: 21.5

字 数: 550 千字

版 次: 2017 年 10 月第 1 版

印 次: 2017 年 11 月第 2 次印刷

印 数: 3501~6500

定 价: 79.00 元

产品编号: 074441-01

# 推 荐 序

在撰写 Vue 文档的过程中，出于篇幅和精力的限制，主要着力于对 Vue 本身 API 的解释。对于缺乏实战经验的读者来说，虽然可能明白了 API 的用法，但对于如何将它使用在实际项目中仍然会感到困惑。而这本书的优点，正是在于对重要的知识点结合了一些实战范例来帮助读者更好地理解 API 设计的初衷和使用场景，并且在 GitHub 有对应的源码可以下载研究。

本书的作者梁灏是优秀的开源 Vue 组件库 iView 的作者，也为 Vue 社区的活跃做出了很多贡献。同时，对开源的投入也使得他对 Vue 的设计和底层实现有相当深入的理解。

如果你喜欢通过实例来学习，那么这本书会是你上手 Vue 的一个好选择。

尤雨溪  
2017年6月

# 引言

两年前，我开始接触 Vue.js 框架，当时就被它的轻量、组件化和友好的 API 所吸引。之后我将 Vue.js 和 webpack 技术栈引入我的公司（TalkingData）可视化团队，并经过一年多的实践，现已成为整个公司的前端开发规范。

与此同时，我开源了 iView 项目，它是基于 Vue.js 的一套高质量 UI 组件库，从设计规范、工程构建到国际化都提供了完整的解决方案，并支持 SSR。在许多志愿者的帮助下，将文档全部翻译为英文，在 Vue 开发者社区颇受欢迎。

如今，前端框架可谓百家争鸣，但每一个框架的产生都是为了解决具体问题的。Vue.js 以渐进式切入，对不同阶段的开发者提供了不同的开发模式，由浅入深。Vue.js 提供了友好的 API 和强大的功能，包括双向数据绑定、路由、状态管理、动画、组件化、SSR，无论是简单的页面还是复杂的系统，从可复用性、便捷性和维护性上都精益求精。

有幸完成此书，希望能给 Vue.js 社区带来一点帮助。

梁灏（Aresn）

2017年7月10日

# 关于本书

本书分为“基础篇”“进阶篇”和“实战篇”三部分，其中基础篇涵盖了 Vue.js 2 的所有基础内容，包括：

- 双向绑定数据；
- 计算属性；
- 内置指令与自定义指令；
- 组件。

基础篇内容相对容易，适合刚入门 Vue.js 的开发者。

进阶篇更深入 Vue.js 的工程化，内容包括：

- Render 函数；
- webpack 的使用；
- Vue.js 插件。

实战篇首先剖析了 iView 的两个经典组件的设计和实现思路，然后充分利用 Vue.js 的内容完成了两个完整的实战项目。

## 本书读者对象

本书基础内容和实战项目共存，适用于刚接触 Vue.js 的前端或后端开发者。当然，有一定 Vue.js 开发经验的读者也能从中收获不少实战中的经验。

本书要求读者有一定的 JavaScript 编程能力，并假定读者已经掌握了 HTML 和 CSS，否则阅读本书会有一些的难度。

## 其他说明

本书基础篇示例只需要有编辑器和浏览器两个环境即可，浏览器以 Chrome 最佳，如果是 IE，至少在 IE 9 版本以上。进阶篇和实战篇的示例需要读者了解并安装 Node.js 和 NPM，本书将不会介绍它们的安装方法。

本书相关的示例代码都放在 GitHub，网址如下：

<https://github.com/icarusion/vue-book>

读者可到该项目的 issues 讨论区提问讨论。

# 目 录

## 第 1 篇 基础篇

第 1 章 初识 Vue.js .....	3
1.1 Vue.js 是什么 .....	3
1.1.1 MVVM 模式 .....	3
1.1.2 Vue.js 有什么不同 .....	4
1.2 如何使用 Vue.js .....	5
1.2.1 传统的前端开发模式 .....	5
1.2.2 Vue.js 的开发模式 .....	5
第 2 章 数据绑定和第一个 Vue 应用 .....	8
2.1 Vue 实例与数据绑定 .....	9
2.1.1 实例与数据 .....	9
2.1.2 生命周期 .....	10
2.1.3 插值与表达式 .....	11
2.1.4 过滤器 .....	13
2.2 指令与事件 .....	15
2.3 语法糖 .....	18
第 3 章 计算属性 .....	19
3.1 什么是计算属性 .....	19
3.2 计算属性用法 .....	20
3.3 计算属性缓存 .....	23
第 4 章 v-bind 及 class 与 style 绑定 .....	25
4.1 了解 v-bind 指令 .....	25
4.2 绑定 class 的几种方式 .....	26
4.2.1 对象语法 .....	26
4.2.2 数组语法 .....	27
4.2.3 在组件上使用 .....	29
4.3 绑定内联样式 .....	30



第 5 章 内置指令 .....	32
5.1 基本指令 .....	32
5.1.1 v-cloak .....	32
5.1.2 v-once .....	33
5.2 条件渲染指令 .....	33
5.2.1 v-if、v-else-if、v-else .....	33
5.2.2 v-show .....	36
5.2.3 v-if 与 v-show 的选择 .....	36
5.3 列表渲染指令 v-for .....	37
5.3.1 基本用法 .....	37
5.3.2 数组更新 .....	41
5.3.3 过滤与排序 .....	43
5.4 方法与事件 .....	44
5.4.1 基本用法 .....	44
5.4.2 修饰符 .....	46
5.5 实战：利用计算属性、指令等知识开发购物车 .....	47
第 6 章 表单与 v-model .....	55
6.1 基本用法 .....	55
6.2 绑定值 .....	61
6.3 修饰符 .....	63
第 7 章 组件详解 .....	65
7.1 组件与复用 .....	65
7.1.1 为什么使用组件 .....	65
7.1.2 组件用法 .....	66
7.2 使用 props 传递数据 .....	70
7.2.1 基本用法 .....	70
7.2.2 单向数据流 .....	72
7.2.3 数据验证 .....	74
7.3 组件通信 .....	75
7.3.1 自定义事件 .....	75
7.3.2 使用 v-model .....	77
7.3.3 非父子组件通信 .....	79
7.4 使用 slot 分发内容 .....	83
7.4.1 什么是 slot .....	83
7.4.2 作用域 .....	84

7.4.3	slot 用法	85
7.4.4	作用域插槽	87
7.4.5	访问 slot	89
7.5	组件高级用法	90
7.5.1	递归组件	90
7.5.2	内联模板	92
7.5.3	动态组件	93
7.5.4	异步组件	94
7.6	其他	95
7.6.1	\$nextTick	95
7.6.2	X-Templates	96
7.6.3	手动挂载实例	97
7.7	实战：两个常用组件的开发	98
7.7.1	开发一个数字输入框组件	98
7.7.2	开发一个标签页组件	106
<b>第 8 章</b>	<b>自定义指令</b>	<b>118</b>
8.1	基本用法	118
8.2	实战	121
8.2.1	开发一个可从外部关闭的下拉菜单	121
8.2.2	开发一个实时时间转换指令 v-time	126

## 第 2 篇 进阶篇

<b>第 9 章</b>	<b>Render 函数</b>	<b>133</b>
9.1	什么是 Virtual Dom	133
9.2	什么是 Render 函数	136
9.3	createElement 用法	140
9.3.1	基本参数	140
9.3.2	约束	143
9.3.3	使用 JavaScript 代替模板功能	147
9.4	函数化组件	153
9.5	JSX	157
9.6	实战：使用 Render 函数开发可排序的表格组件	159
9.7	实战：留言列表	172
9.8	总结	183

第 10 章 使用 webpack .....	184
10.1 前端工程化与 webpack .....	184
10.2 webpack 基础配置 .....	187
10.2.1 安装 webpack 与 webpack-dev-server .....	187
10.2.2 就是一个 js 文件而已 .....	188
10.2.3 逐步完善配置文件 .....	191
10.3 单文件组件与 vue-loader .....	194
10.4 用于生产环境 .....	201
第 11 章 插件 .....	206
11.1 前端路由与 vue-router .....	207
11.1.1 什么是前端路由 .....	207
11.1.2 vue-router 基本用法 .....	208
11.1.3 跳转 .....	212
11.1.4 高级用法 .....	213
11.2 状态管理与 Vuex .....	216
11.2.1 状态管理与使用场景 .....	216
11.2.2 Vuex 基本用法 .....	217
11.2.3 高级用法 .....	221
11.3 实战：中央事件总线插件 vue-bus .....	227

## 第 3 篇 实战篇

第 12 章 iView 经典组件剖析 .....	235
12.1 级联选择组件 Cascader .....	236
12.2 折叠面板组件 Collapse .....	249
12.3 iView 内置工具函数 .....	257
第 13 章 实战：知乎日报项目开发 .....	261
13.1 分析与准备 .....	261
13.2 推荐列表与分类 .....	265
13.2.1 搭建基本结构 .....	265
13.2.2 主题日报 .....	267
13.2.3 每日推荐 .....	271
13.2.4 自动加载更多推荐列表 .....	276
13.3 文章详情页 .....	278

---

13.3.1 加载内容	278
13.3.2 加载评论	281
13.4 总结	286
<b>第 14 章 实战：电商网站项目开发</b>	<b>288</b>
14.1 项目工程搭建	288
14.2 商品列表页	290
14.2.1 需求分析与模块拆分	290
14.2.2 商品简介组件	291
14.2.3 列表按照价格、销量排序	297
14.2.4 列表按照品牌、颜色筛选	306
14.3 商品详情页	309
14.4 购物车	313
14.4.1 准备数据	314
14.4.2 显示和操作数据	316
14.4.3 使用优惠码	320
14.5 总结	324
<b>第 15 章 相关开源项目介绍</b>	<b>325</b>
15.1 服务端渲染与 Nuxt.js	325
15.1.1 是否需要服务端渲染	325
15.1.2 Nuxt.js	326
15.2 HTTP 库 axios	327
15.3 多语言插件 vue-i18n	329

# 第 1 篇 基础篇

基础篇将循序渐进地介绍 Vue.js 的核心功能，包括数据的双向绑定、计算属性、基本指令、自定义指令及组件等。通过对基础篇的学习，可以快速构建出 Vue.js 应用并直接用于生产环境。

# 第 1 章

## 初识 Vue.js

本章主要介绍与 Vue.js 有关的一些概念与技术，并帮助你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.js，你也可以运用这些知识点快速构建出一个 Vue.js 应用。

### 1.1 Vue.js 是什么

Vue.js 的官方文档中是这样介绍它的。

简单小巧的核心，渐进式技术栈，足以应付任何规模的应用。

简单小巧是指 Vue.js 压缩后大小仅有 17KB。所谓渐进式 (Progressive)，就是你可以一步一步、有阶段性地来使用 Vue.js，不必一开始就使用所有的东西。随着本书的不断介绍，你会深刻感受到这一点，这也正是开发者热爱 Vue.js 的主要原因之一。

使用 Vue.js 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，比如：

- 解耦视图与数据。
- 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

与知名前端框架 Angular、Ember 等一样，Vue.js 在设计上也使用 MVVM (Model-View-View Model) 模式。

MVVM 模式是由经典的软件架构 MVC 衍生来的。当 View（视图层）变化时，会自动更新到 ViewModel（视图模型），反之亦然。View 和 ViewModel 之间通过双向绑定（data-binding）建立联系，如图 1-1 所示。

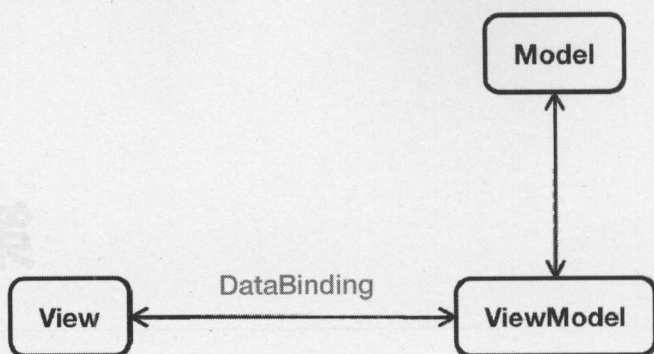


图 1-1 MVVM 关系

### 1.1.2 Vue.js 有什么不同

如果你使用过 jQuery，那你一定对操作 DOM、绑定事件等这些原生 JavaScript 能力非常熟悉，比如我们在指定 DOM 中插入一个元素，并给它绑定一个点击事件：

```
if (showBtn) {  
  var btn = $('<button>Click me</button>');  
  btn.on('click', function () {  
    console.log('Clicked!');  
  });  
  $('#app').append(btn);  
}
```

这段代码不难理解，操作的内容也不复杂，不过这样让我们的视图代码和业务逻辑紧耦合在一起，随着功能不断增加，直接操作 DOM 会使得代码越来越难以维护。

而 Vue.js 通过 MVVM 的模式拆分为视图与数据两部分，并将其分离。因此，你只需要关心你的数据即可，DOM 的事情 Vue 会帮你自动搞定，比如上面的示例用 Vue.js 可以改写为：

```
<body>  
  <div id="app">  
    <button v-if="showBtn" v-on:click="handleClick">Click me</button>  
  </div>  
</body>  
<script>  
  new Vue({  
    el: '#app',  
    data: {  
      showBtn: true  
    },  
  },
```

```
methods: {  
  handleClick: function () {  
    console.log('Clicked!');  
  }  
}  
})  
</script>
```



暂时还不需要理解上述代码，这里只是快速展示 Vue.js 的写法，在后面的章节会详细介绍每个参数的用法。

## 1.2 如何使用 Vue.js

每一个框架的产生都是为了解决某个具体的问题。在正式开始学习 Vue.js 前，我们先对传统前端开发模式和 Vue.js 的开发模式做一个对比，以此了解 Vue.js 产生的背景和核心思想。

### 1.2.1 传统的前端开发模式

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，比如 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。

随着这些技术的普及，一套可称为“万金油”的技术栈被许多商业项目用于生产环境：

jQuery + RequireJS (SeaJS) + artTemplate (doT) + Gulp (Grunt)

这套技术栈以 jQuery 为核心，能兼容绝大部分浏览器，这是很多企业比较关心的，因为他们的客户很可能还在用 IE7 及以下浏览器。使用 RequireJS 或 SeaJS 进行模块化开发可以解决代码依赖混乱的问题，同时便于维护及团队协作。使用轻量级的前端模板（如 doT）可以将数据与 HTML 模板分离。最后，使用自动化构建工具（如 Gulp）可以合并压缩代码，如果你喜欢写 Less、Sass 以及现在流行的 ES 6，也可以帮你进行预编译。

这样一套看似完美无瑕的前端解决方案就构成了我们所说的传统前端开发模式，由于它的简单、高效、实用，至今仍有不少开发者在使用。不过随着项目的扩大和时间的推移，出现了更复杂的业务场景，比如 SPA（单页面富应用）、组件解耦等。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，这时就出现了如 Angular、React 以及我们要介绍的主角 Vue.js。

### 1.2.2 Vue.js 的开发模式

Vue.js 是一个渐进式的 JavaScript 框架，根据项目需求，你可以选择从不同的维度来使用它。如果你只是想体验 Vue.js 带来的快感，或者开发几个简单的 HTML 5 页面或小应用，你可以通过 script 加载 CDN 文件，例如：



```
<!-- 自动识别最新稳定版本的 Vue.js -->
<script src="https://unpkg.com/vue/dist/vue.min.js"></script>
<!-- 指定某个具体版本的 Vue.js -->
<script src="https://unpkg.com/vue@2.1.6/dist/vue.min.js"></script>
```

两种版本都可以，如果你不太了解各版本的差别，建议直接使用最新的稳定版本。当然，你也可以将代码下载下来，通过自己的相对路径来引用。引入 Vue.js 框架后，在 body 底部使用 new Vue() 的方式创建一个实例，这就是 Vue.js 最基本的开发模式。现在可以写入以下完整的代码来快速体验 Vue：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Vue 示例</title>
</head>
<body>
  <div id="app">
    <ul>
      <li v-for="book in books">{{ book.name }}</li>
    </ul>
  </div>
  <script src="https://unpkg.com/vue/dist/vue.min.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: {
        books: [
          { name: '《Vue.js 实战》' },
          { name: '《JavaScript 语言精粹》' },
          { name: '《JavaScript 高级程序设计》' }
        ]
      }
    })
  </script>
</body>
</html>
```

在浏览器中访问它，会将图书列表循环显示出来，如图 1-2 所示。

对于一些业务逻辑复杂，对前端工程有要求的项目，可以使用 Vue 单文件的形式配合 webpack 使用，必要时还会用到 vuex 来管理状态，vue-router 来管理路由。这里提到了很多概念，目前还不必去过多了解，只是说明 Vue.js 框架的开发模式多样化，后续章节会详细介绍，到时就会对整个 Vue 生态有所了解了。