

从基础
到实践

基础知识全面覆盖
实践操作循序渐进

从理论
到应用

理论讲解详尽具体
动手应用实操实练

从入门
到进阶

内容编排由浅入深
进阶案例综合拓展

重点
推荐

Vue.js 3.x

前端开发实战



■ 李文奎 朱继宏◎主编



 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

T1327.002.2
L-852

Vue.js 3.x 前端开发实战

主 编 李文奎 朱继宏
副主编 李小奎 陶宝健 张慧娥



 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书结合最新的 Vue.js 3.x 技术,深入浅出地介绍前端开发框架 Vue “全家桶”需要掌握的相关知识及技能。在理论知识讲解方面,从初学者的角度,以通俗易懂的语言、实用的案例讲解了 Vue 基础知识和技能;在实例制作方面,注重实践、注重知识的运用;在考查知识掌握程度方面,书中每章配有自测题与实践题;最后,以综合案例整合所学知识,有助于学生理解知识、掌握相关技能,达到学以致用目的。

本书既可以作为软件技术、移动应用开发、云计算技术应用、大数据技术专业及其他相关专业的教材,也可以作为广大 IT 技术人员和 Vue.js 爱好者的参考读物。

版权专有 侵权必究

图书在版编目(CIP)数据

Vue.js 3.x 前端开发实战 / 李文奎, 朱继宏主编

. -- 北京: 北京理工大学出版社, 2021. 9

ISBN 978 - 7 - 5763 - 0183 - 0

I. ①V… II. ①李… ②朱… 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 (2021) 第 165942 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)
(010) 82562903 (教材售后服务热线)
(010) 68944723 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 唐山富达印务有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 14.25

字 数 / 335 千字

版 次 / 2021 年 9 月第 1 版 2021 年 9 月第 1 次印刷

定 价 / 75.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 刘亚男

责任印制 / 施胜娟

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

前言

在前端产品日益复杂的今天，越来越多的开发者不再满足于 HTML + JavaScript + CSS 的开发方式，现代前端技术越来越具有广度和深度。本书以当前最流行的前端开发框架——Vue.js 3.x 为编写思路，基于数据驱动和组件化的思想，提供了更加简洁、易于理解的 API，其具有简单易学、组件化、视图、数据和结构分离、虚拟 DOM、运行速度快的特点，能够在很大程度上降低 Web 前端开发难度，深受广大前端开发人员的喜爱。

本书依据计算机相关专业人才培养方案的需要，结合高等职业教育对学生特定职业或职业群所需的知识和技术能力的要求、行业相应岗位能力要求及现代前端开发技术中的新技术和新标准，以“实用为主、够用为度”为原则，在理论知识讲解方面，从初学者的角度，以通俗易懂的语言、实用的案例讲解 Vue.js “全家桶”的基本知识和技巧；在实例制作方面，注重实践、注重知识的运用；书中每章配有自测题与实践题，贴近实际，由浅入深，具有很强的实用性，有助于学生理解知识、掌握知识，最终达到学以致用的目的。

本书由陕西电子信息职业技术学院李文奎、西安城市建设职业学院朱继宏担任主编，陕西电子信息职业技术学院李小奎、西安思源学院张慧娥及西安网星软件咨询服务有限公司经理陶宝健担任副主编。其中李文奎编写第 1 ~ 3 章、朱继宏编写第 4 章、李小奎编写第 5 ~ 7 章、张慧娥编写第 8 ~ 9 章、陶宝健编写第 10 章。全书由李文奎、朱继宏负责教学案例的设计、优化和校对工作。

教材建设是一项长期而艰巨的任务，它对稳定学校教学秩序、全面提高教学质量、培养国家所需高技能人才起着十分重要的作用。本书在编写过程中得到了学院领导的



大力支持和帮助，特别感谢张成现、贺瑞缠、朱蓉蓉三位教授对书稿知识体系的指导。

虽然我们在编写本书的过程中倾注了大量心血，但难免会有疏漏与不妥之处，恳请广大读者及专家批评指正，不吝赐教。本书配备了丰富的教学资源，包括教学大纲、教学课件、教材案例源代码、课后习题答案、考试模拟试卷及前端开发面试题等，请发送电子邮件至 345066179@qq.com 获取。

编 者

目录

第1章 Web 开发基础	1
1.1 Web 开发基础	1
1.1.1 Web 1.0 时代	1
1.1.2 Web 2.0 时代	2
1.2 Web 开发模式	2
1.2.1 MVC	2
1.2.2 MVP	3
1.2.3 MVVM	3
1.3 Vue 简介	4
1.3.1 Vue 的开发模式	4
1.3.2 Vue 的安装	4
1.4 开发工具的使用	6
1.4.1 VS Code	6
1.4.2 Vue.js Devtools 插件	8
1.5 第一个 Vue 程序	9
1.5.1 编写 Vue 程序	9
1.5.2 通过浏览器运行程序	10
习题与实践	11
第2章 Vue 基础	13
2.1 Vue 实例	13
2.1.1 创建 Vue 实例及挂载	13
2.1.2 Vue 数据与方法	14
2.2 计算属性和侦听属性	16
2.2.1 计算属性	16
2.2.2 侦听属性	18



2.3 事件	20
2.3.1 事件监听及事件处理	20
2.3.2 事件修饰符	21
2.3.3 按键修饰符	24
2.4 Vue 生命周期	25
2.4.1 生命周期钩子函数	27
2.4.2 生命周期钩子函数实例	27
习题与实践	29
第3章 Vue 数据绑定	32
3.1 Vue 模板语法	32
3.1.1 模板语法	32
3.1.2 v-text	34
3.1.3 v-html	35
3.2 v-bind	36
3.2.1 class 属性绑定	37
3.2.2 style 属性绑定	39
3.3 条件渲染	40
3.3.1 v-show	40
3.3.2 v-if	41
3.3.3 v-else	43
3.3.4 v-else-if	44
3.4 v-for	45
3.4.1 v-for	45
3.4.2 双重 v-for	47
3.5 v-model	48
3.5.1 绑定文本框	48
3.5.2 绑定单选按钮	50
3.5.3 绑定复选框	50
3.5.4 绑定列表框	52
3.5.5 绑定修饰符	53
3.6 自定义指令	54
3.6.1 自定义指令构成	55
3.6.2 动态指令参数	57
3.7 购物车案例	59
3.7.1 案例说明	59
3.7.2 案例分析与实施	59

习题与实践	63
第4章 Vue 组件	68
4.1 组件的定义与使用	68
4.1.1 注册全局组件	68
4.1.2 注册局部组件	69
4.1.3 .vue 文件	71
4.2 组件通信	71
4.2.1 父组件给予组件传递信息	71
4.2.2 子组件给父组件传递信息	73
4.3 插槽的使用	74
4.3.1 单个插槽	74
4.3.2 具名插槽	75
4.3.3 作用域插槽	77
4.4 简易留言板案例	78
4.4.1 案例说明	78
4.4.2 案例分析与实施	78
4.4.3 源代码	80
习题与实践	84
第5章 vue-router 路由	87
5.1 前端路由	87
5.1.1 vue-router 的引入	87
5.1.2 vue-router 基础用法	88
5.1.3 路由重定向	91
5.2 动态路由	92
5.2.1 query 方式传参	92
5.2.2 params 方式传参	94
5.3 嵌套路由	96
5.3.1 嵌套路由基础	96
5.3.2 嵌套路由的实现	96
5.4 命名路由和命名视图	99
5.4.1 命名路由	99
5.4.2 命名视图	102
5.5 编程式导航	103
5.5.1 push() 方法	103
5.5.2 replace() 方法	106



5.5.3 go() 方法	106
习题与实践	107
第6章 Vue 项目构建	109
6.1 项目脚手架 vue - cli	109
6.1.1 安装 vue - cli	109
6.1.2 使用 vue create 创建项目	110
6.1.3 项目初始目录结构	112
6.2 Vant 插件	114
6.2.1 安装 Vant 插件	114
6.2.2 引入组件	115
6.2.3 使用 Vant 组件	116
6.3 Axios	117
6.3.1 Axios 基础	117
6.3.2 在 Vue 单文件中使用 Axios	120
6.3.3 Axios 在 Vue 项目中的使用	121
6.3.4 Axios 封装及调用	123
习题与实践	125
第7章 组合式 API	127
7.1 组合式 API 简介	127
7.1.1 组合式 API 的入口函数	127
7.1.2 ref 与 reactive 函数	128
7.1.3 toRefs 函数	130
7.2 computed、watch 和 watchEffect	131
7.2.1 computed 函数	131
7.2.2 watch 和 watchEffect	133
7.3 provide 和 inject	136
7.3.1 provide/inject 使用方式	136
7.3.2 provide/inject 实例	136
7.4 组合式 API 生命周期	139
7.4.1 组合式 API 生命周期钩子函数	139
7.4.2 组合式 API 生命周期钩子函数示例	140
习题与实践	142
第8章 Vuex 状态管理	144
8.1 Vuex 概述	144
8.1.1 状态管理模式	144

8.1.2	Vuex 数据状态管理	146
8.1.3	Vuex 基本使用	148
8.2	Vuex 中的配置选项	149
8.2.1	State	149
8.2.2	Getters	150
8.2.3	Mutations	151
8.2.4	Actions	153
8.2.5	Modules	154
	习题与实践	156
第9章	综合案例开发1——万家水果 APP	158
9.1	项目前期准备工作	158
9.1.1	项目创建及插件安装	160
9.1.2	准备数据	162
9.1.3	项目目录	163
9.2	公共组件的设计与制作	163
9.2.1	头部组件	164
9.2.2	轮播图组件	164
9.2.3	导航组件	166
9.3	main.js、app.vue 及 router/index.js 项目文件	167
9.3.1	main.js	167
9.3.2	app.vue	168
9.3.3	router/index.js	169
9.4	首页组件 (index.vue)	171
9.5	商品页组件 (goods.vue)	173
9.5.1	商品页组件 (goods.vue) 结构代码	174
9.5.2	商品页组件 (goods.vue) 逻辑代码	175
9.6	购物车组件 (cart.vue)	178
9.6.1	购物车组件 (cart.vue) 结构代码	179
9.6.2	购物车组件 (cart.vue) 逻辑代码	180
9.7	订单组件 (order.vue)	183
9.7.1	订单组件 (order.vue) 结构代码	183
9.7.2	订单组件 (order.vue) 逻辑代码	184
9.8	我的组件 (about.vue)	185
9.8.1	我的组件 (about.vue) 结构代码	186
9.8.2	我的组件 (about.vue) 逻辑代码	187
9.9	项目的打包与发布	189

第 10 章 综合案例开发 2——个人博客	191
10.1 项目概述	191
10.1.1 前端环境搭建	194
10.1.2 后端环境搭建	194
10.2 MongoDB 简述	195
10.2.1 MongoDB 的安装	195
10.2.2 MongoDB 基础操作	198
10.3 项目部分代码	200
10.3.1 main.js	200
10.3.2 router/index.js	201
10.3.3 app.vue	202
10.3.4 server/index.js	203
10.4 页面组件	205
10.4.1 添加博客组件 (addBlog.vue)	205
10.4.2 分页显示博客 (ListBlog.vue)	208
10.4.3 编辑博客 (EditBlog.vue)	213
10.4.4 博客详情 (SingleBlog.vue)	215
10.5 项目的打包与发布	216

第 1 章

Web 开发基础

伴随着信息时代、大数据时代，Web 开发经历 1.0 时代和 2.0 时代，在此过程中，出现了各种开发模式及框架。本章首先介绍 Web 开发基础及三大模式，其次介绍 Vue 的特点，最后介绍 Vue 开发工具及插件。

【学习目标】

- 了解 Web 开发基础
- 了解并理解 Web 开发的三大模式
- 了解 Vue 的特点
- 熟练使用开发工具及插件安装
- 制作第一个 Vue 程序

1.1 Web 开发基础

Web 开发是创建 Web 页面和 App 等前端界面呈现给用户的过程，通过 HTML、CSS、JavaScript 及其衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互。

1.1.1 Web 1.0 时代

在互联网的演进过程中，网页制作是 Web 1.0 时代的产物，早期网站的主要内容都是静态的，以图片和文字为主，用户使用网站的行为以浏览为主。随着 JavaScript 的出现，可以更改前端 DOM 的样式，但大部分的前端界面还很简单，显示的都是纯静态的文本和图片。这种静态页面不能读取后台数据库中的数据，为了使 Web 更加充满活力，以 PHP、JSP、ASP.NET 为代表的动态页面技术相继诞生。

随着这些动态服务器页面技术的出现，页面不再是静止的，页面可以获取服务器数据信息并不断更新。以 Google 为代表的搜索引擎及各种论坛相继出现，使得 Web 充满了活力。

随着动态页面技术的不断发展，后台代码变得庞大臃肿，后端逻辑也越来越复杂，逐渐

难以维护，Web 开发一直处于后端重、前端轻的状态。

1.1.2 Web 2.0 时代

随着 AJAX（Asynchronous Java And XML）的流行，越来越多的网站使用 AJAX 动态获取数据，这使得动态网页内容变成可能，前端呈现出了欣欣向荣的局面。AJAX 使得浏览器客户端可以更方便地向服务器发送数据信息，这促进了 Web 2.0 的发展。

为了解决浏览器兼容性问题，2006 年用于操作 DOM 的 jQuery 出现，它的出现极大地促进了 Web 2.0 时代的进步，其优雅的语法、符合直觉的事件驱动型编程思维使开发者极易上手，很快风靡前端开发。

伴随着信息时代、大数据时代的到来，jQuery 在对大量数据操作中的弊端体现出来了，它在对 DOM 进行大量的操作中，会导致页面的加载缓慢等问题，而相继出现的 Angular、React、Vue 等 MVVM 模式的模架，以及 Webpack 的前端工程化构建，加速了数据驱动前端工程化的发展。

2009 年，Ryan 利用 Chrome 的 V8 引擎打造的 Node.js 的出现驱使前端产生第二次飞跃，它具有以下特点：基于事件循环的异步 I/O 框架，能够提高 I/O 吞吐量；单线程运行，能够避免多线程变量同步的问题；使得 JavaScript 可以编写后台代码，前后端编程语言统一。

如今，后端负责数据，前端负责其余工作越发明显化。它们之间的通信只需要后端暴露 RESTful 接口，前端通过 AJAX 以 HTTP 协议与后端通信即可，Web 开发进入全栈时代。

1.2 Web 开发模式

为了解决软件开发中出现的复杂性、协作性管理难题，出现了 MVC、MVP、MVVM 三种开发架构模式（Architectural Pattern），它们通过分离关注点（SOC）来改进代码组织方式，最终实现代码间的高内聚、低耦合。

1.2.1 MVC

MVC 模式是 MVP、MVVM 模式的基础，MVC 的全名是 Model View Controller，是 Model（模型）、View（视图）、Controller（控制器）的缩写，是一种软件设计典范。

在 Web 开发中，Model 是指数据模型层，负责提供数据，但不直接与用户产生交互。View 视图层是用户能够看到并进行交互的客户端界面，如桌面程序的图形界面、浏览器端的网页等。Controller 控制层负责收集用户输入的数据，向相关模型请求数据并返回视图实现交互请求，如图 1-1 所示。

MVC 模式具有耦合性低、重用性高、生命周

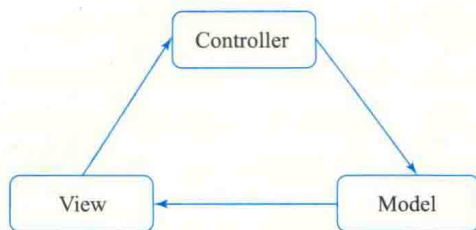


图 1-1 MVC 框架

期成本低、部署快等优点，其缺点是视图与控制器间的过于紧密的连接、视图对模型数据的低效率访问。

1.2.2 MVP

MVP 的全称是 Model View Presenter，它是从 MVC 演变而来的，Model 提供数据，View 负责显示，Presenter 负责逻辑的处理，如图 1-2 所示。在 MVP 中，Model 与 View 完全分离，它们之间的通信是通过 Presenter 进行的，这样做的优点：一个是将 Presenter 用于多个视图，而不需要改变 Presenter 的逻辑；另一个是 Model 和 View 的分离，可以将 View 层抽离出来做成组件。

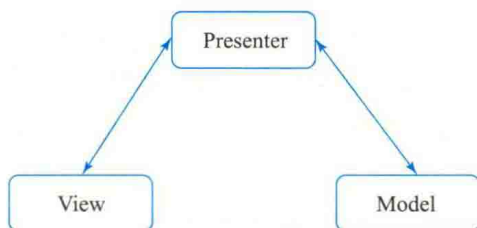


图 1-2 MVP 框架

1.2.3 MVVM

MVVM 的全称是 Model - View - ViewModel，基本上与 MVP 模式完全一致。它们的区别主要是 Presenter 层采用手动写法来调用或者修改 View 层和 Model 层，而 ViewModel 层采用 View 层和 Model 层的双向绑定，当 View 层的数据变化时，系统会自动修改 Model 层的数据，反之同理，如图 1-3 所示。

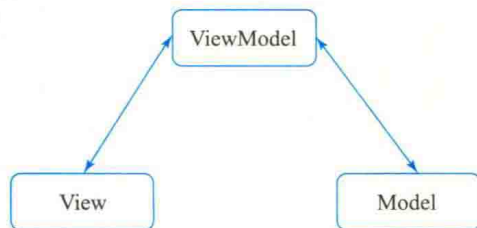


图 1-3 MVVM 框架

MVVM 模式和 MVC 模式一样，主要目的是分离视图（View）和模型（Model）。其有以下几大优点：

①低耦合。视图（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 变化的时候，Model 可以不变；当 Model 变化的时候，View 也可以不变。

②可重用。可以把一些视图逻辑放在一个 ViewModel 里面，让很多 View 重用这段视图逻辑。

③可测试。界面向来都是难以测试的，而现在测试可以针对 ViewModel 来写。

④独立开发。开发人员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 可以很容易设计界面并生成 XAML 代码。

1.3 Vue 简介

Vue.js（以下简称 Vue）是一套用于构建用户界面的渐进式框架，由尤雨溪开发并于 2014 年 2 月发布。与其他大型框架（如 Angular/React）不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，这样不仅灵活方便，还便于与第三方库（如 vue-router/vuex/axios）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 也完全能够为复杂的单页应用提供驱动。

1.3.1 Vue 的开发模式

Vue 是基于 MVVM 的开发模式，采用 MVVM 模式实现视图（View）和数据（Model）的分离，通过自动化脚本实现自动化关联，无须手动处理。ViewModel 是实现视图与数据的桥梁，同时，在 ViewModel 中可以进行交互及逻辑处理，如图 1-4 所示。

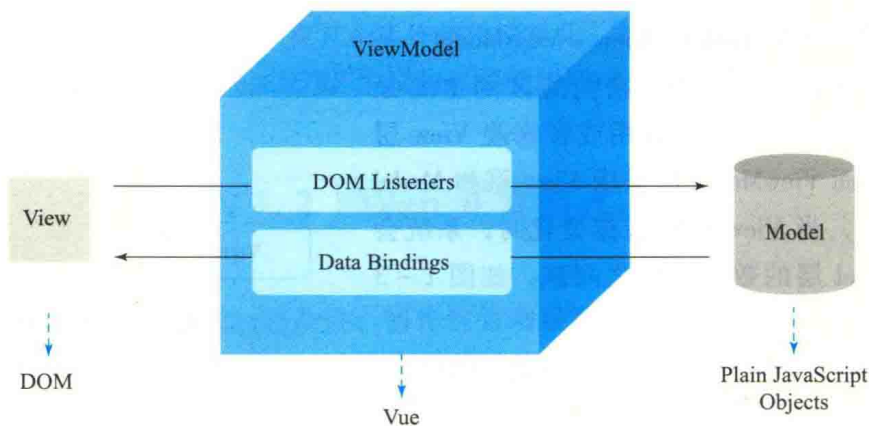


图 1-4 基于 MVVM 模型的 Vue

从图中可以看出，ViewModel 是 Vue 的核心，它是一个 Vue 实例（作用于某一个 HTML 元素上）。当创建 ViewModel 后，从 View 侧看，ViewModel 中的 DOM Listeners 工具会监测页面上 DOM 元素的变化，如果有变化，则自动更新 Model 中的数据；从 Model 侧看，当更新 Model 中的数据时，Data Bindings 工具会自动更新页面中的 DOM 元素。

1.3.2 Vue 的安装

根据不同的需求，Vue 添加到项目中有以下五种主要方式：

1. 直接下载并使用 <script> 标签引入

进入 Vue 官网，根据需求下载开发版本或生产版本。下载完成后，在项目文件中通过

<script> 标签引入，例如：

```
<script type="text/javascript" src=vue.js >
```

2. 使用 CDN 方式

对于制作原型或学习，可以使用最新版本：

```
<script src="https://unpkg.com/vue@next"></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，以避免新版本造成的不可预期的破坏。

3. NPM

若用 Vue 构建大型应用，推荐使用 NPM 安装。NPM 能很好地和诸如 Webpack 或 Browserify 模块打包器配合使用。同时，Vue 也提供配套工具来开发单文件组件。

```
# 最新稳定版
```

```
$ npm install vue@next
```

推荐使用淘宝 NPM 镜像 CNPM，先执行以下命令：

```
$ npm install -g cnpm --registry=https://registry.npm.taobao.org
```

4. 命令行工具

Vue 提供了一个官方的 CLI，为单页面应用（SPA）快速搭建繁杂的脚手架。它为现代前端工作流提供了功能齐备的构建设置。只需要几分钟就可以运行起来，并带有热重载、保存时 lint 校验，以及生产环境可用的构建版本。

通过在终端中运行以下命令，可以快速构建 Vue 项目：

```
$ npm install -g @vue/cli
$ vue create <project-name>
$ npm run serve
```

5. Vite

Vite 是一个 Web 开发构建工具，由于其原生 ES 模块导入方式，可以实现闪电般的冷服务器启动。

通过在终端中运行以下命令，可以使用 Vite 快速构建 Vue 项目：

```
$ npm init @vitejs/app <project-name>
$ cd <project-name>
$ npm install
$ npm run dev
```

1.4 开发工具的使用

“工欲善其事，必先利其器”，利用 Vue 进行前端开发，可以利用先进开发工具和调试工具帮助开发者提高开发效率和降低出错概率。

1.4.1 VS Code

Visual Studio Code（简称 VS Code）是一款免费开源的现代化轻量级代码编辑器，支持几乎所有主流的开发语言的语法高亮、智能代码补全、自定义热键、括号匹配、代码片段等特性，支持插件扩展，并针对网页开发和云端应用开发做了优化。同时，软件跨平台支持 Win、Mac 及 Linux，运行流畅。

1. 下载并安装

登录 VS Code 官网（<https://code.visualstudio.com/>），根据用户电脑版本选择相应的版本并下载，如图 1-5 所示。



图 1-5 VS Code 版本

根据默认方式进行安装，安装成功并启动，如图 1-6 所示。

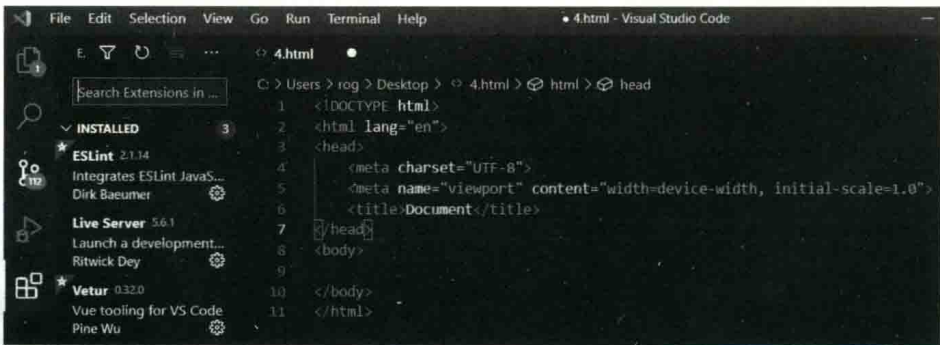


图 1-6 VS Code 界面