

巍巍交大 百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn



策划编辑 李 勇
责任编辑 胡思佳
封面设计 刘文东

Vue.js 3
QIANDUAN KAIFA XIANGMU JIAOCHENG

Vue.js 3 前端开发 项目教程

免费提供

★★★ 精品教学资料包

服务热线: 400-615-1233
www.huatengzy.com



扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-29802-7



9 787313 298027

定价: 55.00元



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

校企「双元」合作新形态教材

Vue.js 3 前端开发项目教程

主编 陈希球 肖红 黄校尧

校企“双元”合作新形态教材

Vue.js 3 前端开发 项目教程

主编 陈希球 肖红 黄校尧



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

校企“双元”合作新形态教材

Vue.js 3 前端开发 项目教程

主 编 陈希球 肖 红 黄校尧

副主编 周 艳 杨本胜 蒋荣军 宋敏鹏



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书共设计了5个教学情景,包含20个任务。教学情景1讲解前端开发环境的搭建与Vue单页面应用的创建,通过学习学生可以利用构建工具快速地创建Vue.js 3项目;教学情景2讲解Vue.js 3数据绑定与指令知识,旨在帮助学生掌握Vue.js 3的基础语法知识,熟悉Vue.js 3各种数据绑定语法与指令的应用方法;教学情景3讲解组件的创建及应用,通过学习学生可以领会面向组件编程的思想,并能使用组件进行程序设计;教学情景4讲解路由的安装及应用,通过学习学生可以掌握路由的正确配置和使用方法,并能运用路由策略开发大型项目;教学情景5讲解一个综合项目——“国货之光”商城实战,通过学习学生可以了解项目从需求分析到代码实现的全过程。

本书既可以作为计算机相关专业的教材,也可以作为广大计算机编程爱好者的参考书。

图书在版编目(CIP)数据

Vue.js 3 前端开发项目教程/陈希球,肖红,黄校尧主编. —上海:上海交通大学出版社,2024. 5
ISBN 978-7-313-29802-7

I. ①V… II. ①陈… ②肖… ③黄… III. ①网页制
作工具—程序设计—教材 IV. ①TP393.092.2

中国国家版本馆 CIP 数据核字(2024)第 001597 号

Vue.js 3 前端开发项目教程

Vue.js 3 QIANDUAN KAIFA XIANGMU JIAOCHENG

主 编:陈希球 肖 红 黄校尧

出版发行:上海交通大学出版社

邮政编码:200030

印 制:三河市骏杰印刷有限公司

开 本:787 mm×1 092 mm 1/16

字 数:485 千字

版 次:2024 年 5 月第 1 版

书 号:ISBN 978-7-313-29802-7

定 价:55.00 元

地 址:上海市番禺路 951 号

电 话:021-64071208

经 销:全国新华书店

印 张:18.5

印 次:2024 年 5 月第 1 次印刷

电子书号:ISBN 978-7-89424-596-0

版权所有 侵权必究

告读者:如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话:0316-3662258

前言

PREFACE

党的二十大报告指出，教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑。党的二十大报告明确了人才强国战略在新时代的科学内涵和使命任务，对加快世界重要人才中心和创新高地建设作出了科学擘画。

Vue.js 是目前流行的 Web 应用开发的前端框架之一，是一套构建用户界面的渐进式框架，采用 MVVM 设计模式，支持数据驱动和组件化开发。与 Angular、React 等前端框架相比，Vue.js 提供了更加简洁、更易于理解的 API，能实现 Web 应用项目的快速开发。

本书是按照 Web 前端开发职业岗位的实际需要编写而成的，内容的编排以“情景—任务”为主线，引导学生在学中做，在做中学，并注重思维启发，达到熟能生巧、举一反三的目的。

本书共设计了 5 个教学情景，分别是初识 Vue.js 3、走进数据绑定与指令、进入组件、迈入路由、进阶商城实战。每个教学情景又包含若干个任务，各任务又以任务导入、任务目标、素质目标、知识准备、任务设计、任务实施和实践练习的框架模式编写。通过学习，学生可以掌握运用 Vue 开发 Web 应用程序的基础理论和实践技能。

为落实立德树人根本任务，本书坚持教育与产业、课程内容与职业标准、教学过程与生产过程的深度对接；运用任务驱动模式，将任务结果呈现在读者面前，用任务引领知识，让学生在完成任务的过程中学习相关知识，掌握岗位技能，提升综合职业能力。

本书按照前端工程师成长规律将相关工作内容序化成学习任务，紧紧围绕完成任务来设计知识点与技能，注重内容的实用性和针对性，使之符合学生的认知规律，打破了长期以来理论和实践分离的教学模式。

1. 学习本书所需基础知识

在实际的案例与任务中，前端开发都是使用 HTML5 编写页面，使用 CSS3 美化其样式的，因此学生需具备 HTML5、CSS3 基础语法知识，以及利用 HTML5+CSS3 进行网页设计的基本能力。

Vue.js 3 以 JavaScript 为基础，因此学生也需具备 JavaScript 的基础知识，包括关键字、语法、函数、事件、对象等。同时，Vue.js 3 中使用了大量的 ECMAScript 6.0 (ES6) 新特性，包括变量、箭头函数、解构、扩展运算符、模板字符串等。书中对用到的语法进行了简单的注释以确保学生能理解，且将 ECMAScript 6.0 (ES6) 新特性作为拓展阅读做了专门的介绍，以帮助学生更好地掌握相关知识。

2. 学习本书所需开发工具

选择一个合适的代码编辑器对于学习 Vue.js 3 尤其重要。本书推荐使用 Visual Studio Code 或 HBuilderX。它们都非常友好地支持 Vue.js 3，特别是可以提供大量插件帮助学生编辑代码。此外，还需要使用 Web 浏览器，建议使用 Chrome 或 Edge，调试更加方便。

3. 本书阅读建议

对于没有前端页面编程经验的初学者而言，建议按本书知识点的编排顺序学习，学习过程中可参照书中丰富的案例、详细的任务实现步骤一步步动手实践。

对于有前端框架使用经验或 Vue.js 2.0 开发经验的学生而言，可依需选择合适的教学情景及任务学习，重点关注组合式 API、语法糖等新特性与语法。

本书中配有丰富且详尽的实例源码，也给出了源码的运行结果，学生在阅读本书时不妨打开编辑器编写、修改这些代码并实现其效果。动手编码可以帮助自己进一步加深理解，有助于更快地掌握相关知识点。

本书由广州松田职业学院陈希球、肖红、黄校尧担任主编，广州松田职业学院周艳、杨本胜、蒋荣军、宋敏鹏担任副主编。广州文豆网络科技有限公司、广州腾科网络技术有限公司为本书的编写提供了大量的案例，在此一并表示感谢。

由于计算机科学技术发展迅速，加之编者水平有限，书中难免存在不足之处，望广大读者提出宝贵意见或建议，以便我们修订和完善。

编 者

目录

CONTENTS

教学情景 1	初识 Vue.js 3	1
任务 1-1	搭建前端开发环境	1
任务 1-2	创建 Vue 单页面应用	16
教学情景 2	走进数据绑定与指令	25
任务 2-1	新闻详情单页的设计与实现	26
任务 2-2	商品购买单页的设计与实现	36
任务 2-3	用户注册单页的设计与实现	57
任务 2-4	投票单页的设计与实现	74
教学情景 3	进入组件	91
任务 3-1	选项卡组件的设计与实现	92
任务 3-2	轮播图组件的设计与实现	108
任务 3-3	折叠面板组件的设计与实现	126
教学情景 4	迈入路由	141
任务 4-1	垂直导航栏单页面的设计与实现	142
任务 4-2	搜索单页面的设计与实现	157
任务 4-3	二级导航菜单单页面的设计与实现	174
教学情景 5	进阶商城实战	193
任务 5-1	开发前的准备	194
任务 5-2	商城首页	215
任务 5-3	列表页	237

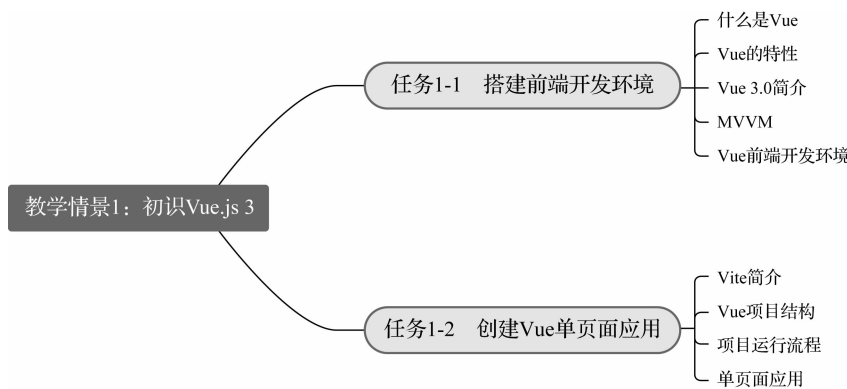
任务 5-4 详情页	246
任务 5-5 购物车	254
任务 5-6 订单页	265
任务 5-7 登录页	276
任务 5-8 打包与发布	284

参考文献	290
-------------------	-----

教学情景 1

初识 Vue.js 3

Vue.js（以下简称 Vue）是流行的前端框架之一，与流行的另外两个框架 Angular 和 React 相比，Vue 框架实现上更容易理解，上手更快。使用 Vue 开发不仅提高了开发效率，也改善了开发体验，因此，熟练掌握 Vue 框架技术已经成为前端开发者的必备技能。本教学情景将搭建前端开发环境，构建 Vue3 项目，实现第一个 Vue3 单页面应用。本教学情景思维导图如下：



任务 1-1 搭建前端开发环境



任务导入

前端技术的发展经历了从静态走向动态（前端 1.0）、从后端走向前端（前端 2.0）的过程。目前，前端技术正从前端走向全端（前端 3.0）。

Web 前端使用 HTML、CSS 和 JavaScript 作为基础语言，它们分别用来实现网页的结构、样式和行为。HTML 用来编写网页的结构，CSS 样式表用来实现元素的样式与布

局的页面效果，JavaScript 用来实现页面逻辑、行为、动作等页面交互效果，实现更好的用户体验。

在构建大型交互式项目时，开发者需要编写大量的 JavaScript 代码来操作文档对象模型（document object model，DOM），代码逻辑越来越烦琐。为了提高开发效率，使用 JavaScript 语言编写的 jQuery 库出现了。jQuery 库是对 JavaScript 代码的封装，使得操作 DOM、事件处理、动画效果等交互功能变得更加简洁，有效地提高了项目开发效率。

随着异步 JavaScript 和 XML（asynchronous JavaScript and XML，AJAX）技术在 2005 年被 Jesse James Garrett 提出，网页应用能够快速地将增量更新呈现在用户界面上，而不需要重载（刷新）整个页面，这使得程序能够更快地响应用户操作，从而获得更好的用户体验。

Node.js 的诞生标志着 JavaScript 成为与 PHP、Python、Perl、Ruby、JSP 等服务器端语言同等重要的脚本语言。前端工程师可以以较低的成本用 Node.js 和 MongoDB 搭建一个后台。至此，前端工程师与全栈工程师之间的距离只在于一个 data base（数据库）了。

近年来，随着微信小程序和支付宝小程序的兴起，前端技术早已超脱了 Web 和 Hybrid 应用的范围。前端工程师很容易基于固有的技术栈快速上手和开发小程序类微应用。

可以预见，未来的前端和前端衍生技术会应用到 Web 和桌面 PC、移动端、智能电视、游戏机等各个方面。未来的工程师会分为负责数据方面的云端工程师和全端（前端）工程师。

要想成为一名前端工程师，熟练使用一到多门主流的前端框架是必须掌握的技能。本任务将带领读者进入一个当前国内非常流行的前端框架 Vue 的世界，开启 Vue 前端框架学习之旅。



任务目标

知识目标	能力目标	教学重点	教学难点
◎简述 Vue 前端框架及其特性。 ◎阐述 MVVM 设计模式。 ◎解释 Node.js、npm、cnpm、vite 等概念	◎能安装 Node.js，并使用 npm、cnpm 安装 vue 等依赖包。 ◎能搭建 Vue 前端开发环境	◎Vue 前端开发环境的搭建	◎MVVM 设计模式



素质目标

了解 Vue.js 之父——尤雨溪和 Vue.js 登顶 GitHub 之路；激发学生的求知欲和科技报国的家国情怀和使命担当感。



知识准备



1-1 搭建前端开发环境（理论）-4itmy

知识点 1 什么是 Vue

Vue（发音为 /vju:/）是一套用于构建用户界面的渐进式 JavaScript 前端框架，由中国人尤雨溪开发。它基于标准 HTML、CSS 和 JavaScript 构建，并提供了一套声明式的、组件化的编程模型，帮助全栈（前端）工程师高效地开发用户界面。无论是设计简单还是设计复杂的界面，Vue 都可以胜任。

Vue 是一个框架，也是一个生态。其功能覆盖了大部分前端开发常见的需求。但 Web 世界是多样化的，不同的开发者在 Web 上构建的东西可能在形式和规模上会有很大的不同。Vue 的设计非常注重灵活性和“可以被逐步集成”这个特点。根据需求场景，可以用不同的方式使用 Vue，所以 Vue 也被称为“轻量级渐进式框架”。

与其他重量级框架不同的是，Vue 的核心库只关注视图层，而且被设计为可以自底向上逐层应用。得益于灵活性及逐步集成的特性，使得 Vue 十分容易上手。同时，它与第三方库或既有项目的整合也十分便捷。

知识点 2 Vue 的特性

目前市场上三大前端主流框架是 Angular、React 和 Vue。Vue 秉承了 Angular 和 React 两者的优势，且代码简洁、上手容易，广受开发者青睐，在市场上得到了大量的应用。与市场上主流的前端框架相比，Vue 具有下列优势与特性。

1. 轻量级

Vue 核心库只关注视图层，是一个构建数据的视图集合，大小只有几十字节，是一个通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。Vue 具有很多独立的功能或插件，可以根据项目需求来灵活选用。

2. 双向数据绑定

Vue 是一个 MVVM（model-view-view model，模型-视图-视图模型）框架，数据双向绑定。当数据（此处指 model）发生变化时，视图（view）也同步更新；当视图发生变化时，数据也会跟着同步更新。特别是在处理表单数据时，双向数据绑定尤其方便。

3. 组件化开发

Vue 可以进行组件化开发。在构建大型应用时，通常会把可复用性强的部分提取出来（如头部导航、侧边栏等需要在多个页面展示的部分），构成一个个组件（独立显示的视图单元），最终组合成一个完整的页面。组件化开发能够减少重复代码的编写，提高开发效率，降低代码之间的耦合程度，使项目更易维护和管理。

4. 虚拟 DOM

Vue 在渲染视图时，将视图中真实的元素节点抽象化，形成虚拟 DOM，操作虚拟 DOM，最后将虚拟 DOM 中变化的元素节点映射到真实的 DOM，能有效地减少页面渲染的次数和 DOM 重绘重排次数，从而提高渲染性能。此外，虚拟 DOM 以 JavaScript 对象

为基础而不依赖真实平台环境，配合不同的渲染工具，使得 Vue 具备跨平台（浏览器、Web 服务器）的优势。

知识点 3 Vue 3.0 的特性

Vue 从诞生（2013 年 1 月）到现在，历经 3 个版本，分别是 2014 年 1 月正式发布的 Vue 1.0，2016 年 10 月发布的 Vue 2.0 及 2020 年 9 月正式发布的 Vue 3.0。

对比 Vue 2.0，Vue 3.0 通过 tree-shaking（把无用的模块进行“剪枝”）减少约 41% 捆绑包体积，初始渲染速度提升约 55%，更新速度提升约 133%，内存使用率降低约 54%。在 Vue 2.0 的基础上，Vue 3.0 增添了许多新的特性。

1. TS 支持

全面拥抱 TS（TypeScript）。TS 有效地解决了前端工程化和规模化的问题，它在代码规范和设计模式上极大地提高了代码质量，进而增强了系统的可靠性、稳定性和可维护性。

2. 组合式 API

Vue 2.0 及之前使用的是选项式（options）API，随着业务复杂度越来越高，代码量会不断加大，整个代码逻辑都不易阅读和理解。组合式（composition）API 可以按逻辑关注点对部分代码进行分组，然后提取逻辑片段（fragment）并与其他组件共享代码，可以让程序员编写更有条理的代码，提高代码逻辑的可复用性。

Vue 3.0 支持组合式 API 与选项式 API，简单的项目使用选项式 API 更加简单方便，大中型需要大量逻辑复用的项目或是强支持 TS 的项目首选组合式 API。

3. 瞬移组件

瞬移（teleport）允许程序员控制在 DOM 中哪个父节点下呈现 HTML，是能够将模板移动到 DOM 中 Vue App 之外的其他位置的技术。当组件模板的一部分逻辑上属于该组件，而从技术角度来看，最好将模板的这一部分移动到 DOM 中 Vue App 之外的其他位置时可以应用瞬移技术，从而方便解决组件间的 CSS 层级问题。

4. 片段

当组件的模板结构中出现多个标签时，可以不用根标签，Vue3 会将多个标签包含在一个片段虚拟元素中，从而“自动”套上了根节点，可以减少标签层级与内存占用。

5. 单文件组件

在 Vue 单文件组件（<style scoped>）中的 style 标签上添加特殊的属性 scoped，它的 CSS 样式就只能作用于当前的 Vue 单文件组件。通过该属性，可以使单文件组件之间的样式不互相污染。如果一个项目中的所有单文件组件 style 标签全部加上了 scoped，相当于实现了样式的模块化。

6. 实验性特性

目前，Vue 3.0 还有一些实验性特性，包括单文件组件组合式 API 语法糖（<script setup>）与单文件组件状态驱动的 CSS 变量（<style vars>）。

知识点 4 MVVM

Vue 框架采用的模式是 MVVM。MVVM 本质上是 MVC（model-view-controller，模

型-视图-控制器)的改进版,其最重要的特性是数据绑定(data binding),此外还包括依赖注入、路由配置、数据模板等一些特性。

MVVM 的主要目的是分离视图层和模型层,view model 层封装了界面展示和操作的属性和接口。通过数据绑定将 view 和 view model 关联在一起,当 view model 中的数据发生变化时,view 也会同步进行更新,如图 1-1-1 所示。

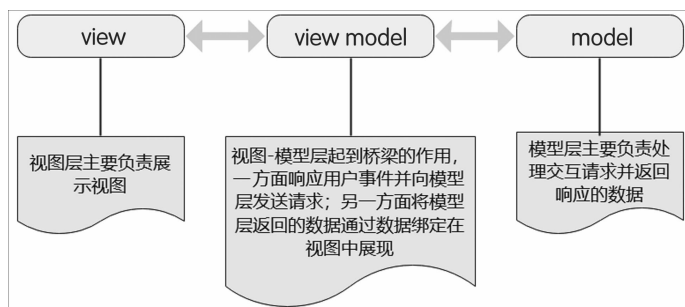


图 1-1-1 MVVM

MVVM 解耦了视图和模型。每一个视图都有一个对应的 view model,同时 view model 与模型建立联系。当接收到用户请求后,view model 获取模型响应的数据,并通过数据绑定将相应的视图页面重新渲染。模型层数据只需要传入 view model 即可实现视图的同步更新,实现了视图与模型之间的松散耦合。

知识点 5 Vue 前端开发环境

搭建前端开发环境,创建基于 Vue 前端框架的应用,涉及的技术体系和开发工具如图 1-1-2 所示。

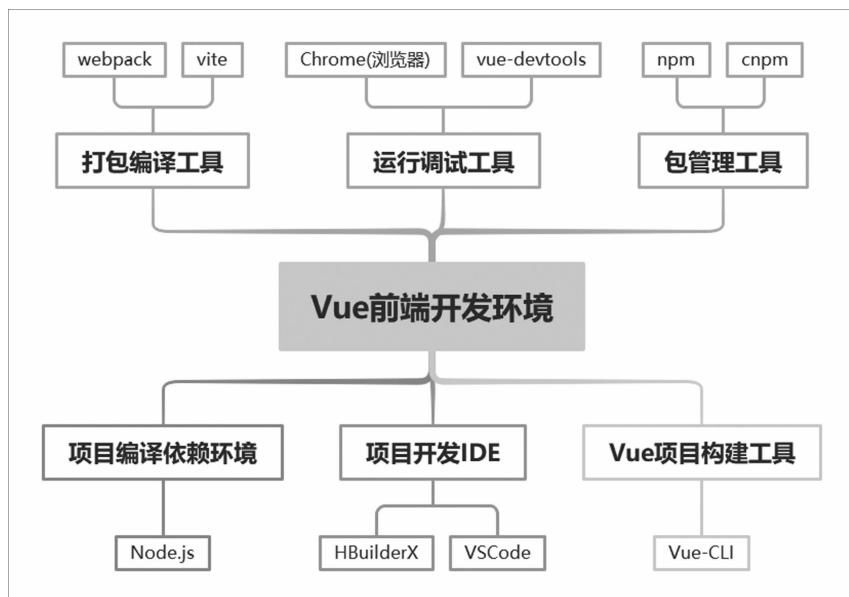


图 1-1-2 Vue 前端开发环境

1. Node.js

Node.js 发布于 2009 年 5 月，由 Ryan Dahl 开发，是一个基于 Chrome V8 引擎跨平台的 JavaScript 运行环境。相对于传统的 JavaScript 来说，Node.js 是一个运行在服务器端的框架，用它开发动态网页，能大大提升运行的性能及开发的便利性。

Node.js 也是一个服务 js 平台，结合自带的 npm、webpack 和第三方提供的 grunt 与 express 等工具，能实现强大的代码、项目管理与服务等功能。

2. npm

npm (Node.js package manager) 是一个 Node.js 包管理工具，用来解决 Node.js 代码部署问题。在安装 Node.js 时会自动安装相应的 npm 版本，不需要单独安装。使用 npm 包管理工具可以解决如下场景需求。

- (1) 从 npm 服务器下载别人编写的第三方包到本地使用。
- (2) 从 npm 服务器下载并安装别人编写的命令程序到本地使用。
- (3) 将自己编写的包或命令程序上传到 NPM 服务器供别人使用。

3. cnpm

由于 npm 工具的仓库源布置在国外，传输速度较慢，可以使用淘宝定制的命令行工具 cnpm 替代。cnpm 是 npm 在国内的镜像，下载的速度较快。npm 与 cnpm 的用法完全一致。

4. webpack

webpack 是一个 JavaScript 应用的静态模块化打包工具，可以把前端项目中的 js、css、scss/less、图片等资源文件都打包在一起，实现自动化构建，给前端开发人员带来极大的便利。

5. vite

vite 是下一代前端开发与构建工具。vite 意在提供开箱即用的配置。vite 使用简单，只需执行初始化命令，就可以得到一个预设好的开发环境，开箱即获得一堆功能，包括 CSS 预处理、HTML 预处理、异步加载、分包、压缩等。

vite 服务器启动速度比 webpack 快。当浏览器请求需要的模块时再对模块进行编译，这种按需动态编译的模式极大缩短了编译时间，当项目越大，文件越多时，vite 的开发优势越明显。vite 热更新比 webpack 快，当某个模块内容改变时，让浏览器去重新请求该模块即可，而不是像 webpack 将该模块的所有依赖重新编译。

6. vue-devtools

vue-devtools 是一款基于 Chrome 浏览器的扩展程序，它提供了一个查看 Vue 组件和全局状态管理器 Vuex 中数据的界面，便于开发者在一个更友好的界面中调试 Vue 应用。

7. 谷歌浏览器

谷歌浏览器 (Chrome) 是一款由 Google (谷歌) 公司开发的网页浏览器。它对 HTML5、CSS3、ECMAScript 6 (下一代 JavaScript 标准) 提供了良好的支持。

8. IDE

当前，前端开发主流的两款集成开发环境 (integrated development environment,

IDE) 分别是 HBuilder 与 Visual Studio Code。

HBuilderX (HX) 是一款免费的国产前端开发工具。其特点是轻巧、极速、智能语法提示、提供对 Vue 强大的支持, 深受前端开发初学者的喜爱。

Visual Studio Code (VSCode) 是由微软公司研发的一款免费、开源的跨平台代码编辑器, 具有轻巧、启动速度快、插件丰富、跟踪代码等特性, 是目前前端 (网页) 开发使用最多的一款软件开发工具, 也是本书所选用的开发工具。

任务设计

本任务将搭建一个基于 Vue 框架的前端开发环境, 在不使用 Vue 项目构建工具 (Vue-CLI 或 vite) 的条件下, 安装和引入 Vue 及其特定的调试工具, 创建一个 Vue3 应用, 运行调试并输出结果。

任务实施

本任务实施分为 3 个子任务, 分别是搭建前端开发环境、创建 Vue3 应用程序和运行与调试 Vue3 应用程序。

子任务 1 搭建前端开发环境

搭建本项目前端开发环境, 需要安装项目编译依赖环境 (Node.js)、包管理工具 (npm 和 cnpm)、运行调试工具 (chrome 与 vue3-devtools) 和项目开发 IDE (VSCode)。

1. 安装 Node.js

(1) 打开 Node.js 官网 (<https://nodejs.org/en/>), 即可看到下载界面, 选择 LTS 版本进行下载, 如图 1-1-3 所示。



1-1 搭建前端开发环境 (实操) -ljq50y

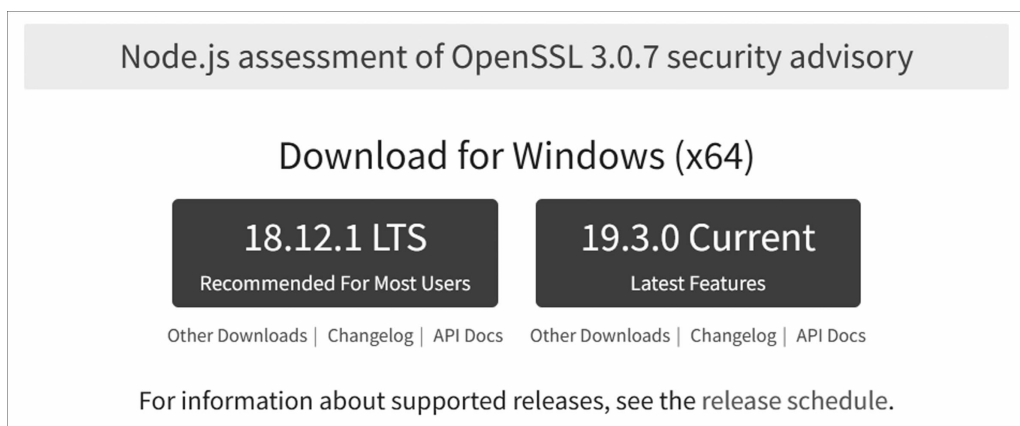


图 1-1-3 Node.js 官网下载界面

从图 1-3 中可以看出, Node.js 有两个版本。LTS (long term support) 是提供长期支持的版本, 只进行微小的 bug 修复且版本稳定, 因此有很多用户在使用, 也是本书所选用的版本; Current 是当前发布的最新版本, 增加了一些新特性, 有利于进行新技术的开发

使用。

(2) 双击下载的安装包 (node-v18.12.1-x64.msi) 进行安装, 如图 1-1-4 所示。

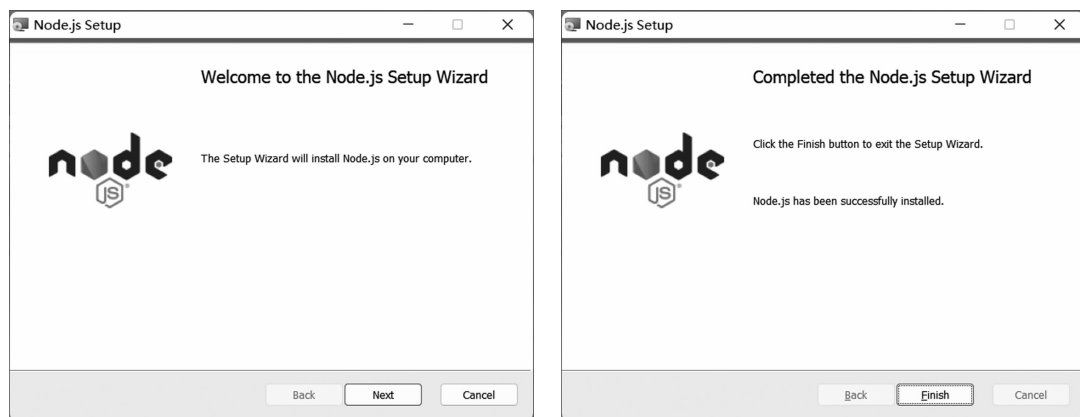


图 1-1-4 安装 Node.js

(3) 在命令行中输入命令 `node -v` 并按 Enter 键。若能输出版版本号, 则表示 Node.js 安装成功, 如图 1-1-5 所示。

```
C:\Users\xiaoguo>node -v
v18.12.1
```

图 1-1-5 验证 Node.js 安装

2. 验证 npm

安装 Node.js 时会自动安装 npm (Node.js package manager)。在命令行中输入命令 `npm -v` 并按 Enter 键。如果能输出版版本号, 则表示 npm 安装成功, 如图 1-1-6 所示。

```
C:\Users\xiaoguo>npm -v
8.19.2
```

图 1-1-6 验证 npm 安装

3. 安装 cnpm

在命令行输入命令 `npm install -g cnpm --registry=https://registry.npm.taobao.org` 并按 Enter 键安装 cnpm。完成后再输入命令 `cnpm -v` 并按 Enter 键。若能输出版版本号, 则表示 cnpm 安装成功, 如图 1-1-7 所示。

```
C:\Users\xiaoguo>cnpm -v
cnpm@9.0.1 (C:\Users\xiaoguo\AppData\Roaming\npm\node_modules\cnpm\lib\parse_argv.js)
npm@8.19.3 (C:\Users\xiaoguo\AppData\Roaming\npm\node_modules\cnpm\node_modules\npm\index.js)
node@18.12.1 (D:\Program Files (x86)\nodejs\node.exe)
npminstall@7.2.0 (C:\Users\xiaoguo\AppData\Roaming\npm\node_modules\cnpm\node_modules\npminstall\lib\index.js)
prefix=C:\Users\xiaoguo\AppData\Roaming\npm
win32 x64 10.0.22000
registry=https://registry.npmmirror.com
```

图 1-1-7 验证 cnpm 安装

4. 安装 Chrome 浏览器

打开 Google Chrome 官网 (<https://www.google.cn/intl/zh-CN/chrome/>), 下载 Chrome, 直接双击安装。

5. 安装 vue-devtools

在 Vue 学习和开发之前, 建议在浏览器 (推荐使用 Google Chrome) 上先安装 vue-devtools 拓展程序。直接访问 Google Web Store, 搜索 vuejs-devtools 进行安装。

(1) 下载 vue-devtools 程序。打开网站 <https://gitcode.net/mirrors/vuejs/devtools>, 选择当前最新的版本下载, 如图 1-1-8 和图 1-1-9 所示。



图 1-1-8 选择最新版本



图 1-1-9 下载 zip 源码

下载后解压, 解压后的文件夹名为 devtools-v6.4.5, 这里重命名为 devtools。

(2) 打开命令提示符窗口, 进入 devtools 目录, 输入命令 yarn install 并按 Enter 键, 安装构建工具所需的依赖, 如图 1-1-10 所示。

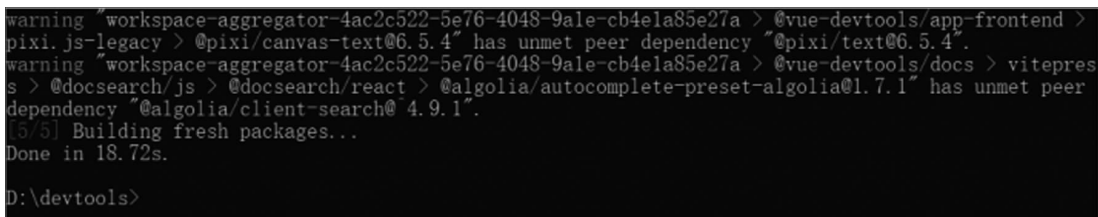


图 1-1-10 安装依赖

(3) 输入命令 yarn run build 并按 Enter 键, 构建本地软件包, 如图 1-1-11 所示。

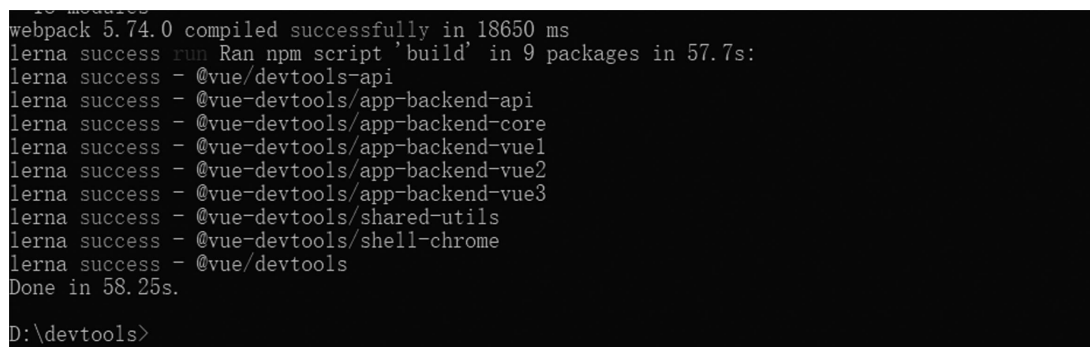


图 1-1-11 构建软件包

构建软件打包成功后，在 devtools 目录下的 packages 目录中会产生 shell-chrome 文件夹，这个文件夹就是打包后的 devtools 程序。

(4) 打开 Chrome 浏览器，选择“菜单”→“更多工具”→“拓展程序”选项，开启开发者模式（单击右上角的“开发者模式”按钮），如图 1-1-12 所示。



图 1-1-12 开启开发者模式

(5) 单击左侧的“加载已解压的扩展程序”按钮，选择 devtools/packages/shell-chrome 文件夹，等待安装完成，如图 1-1-13 所示。



图 1-1-13 安装 devtools 扩展程序

安装成功后，可以在网页上按 F12 键调出调试工具查看 vue devtools，如果能监测到页面中加载了 Vue.js，则界面如图 1-1-14 所示。

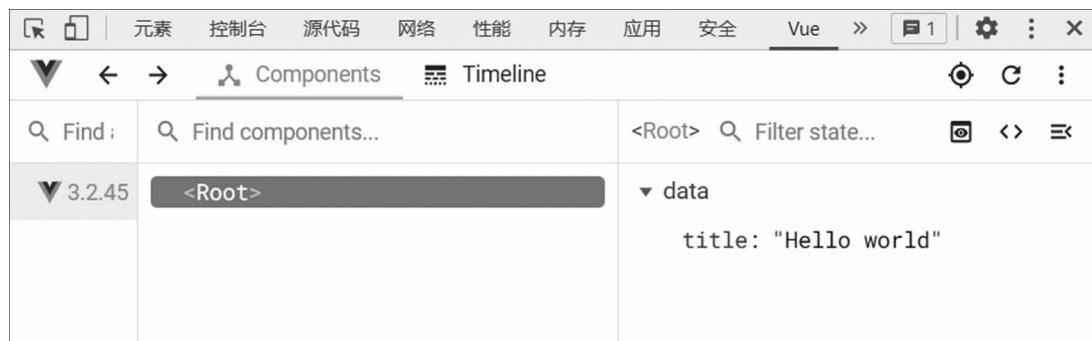


图 1-1-14 devtools 调试工具

6. 安装 Visual Studio Code

打开 Visual Studio Code 官网 (<https://code.visualstudio.com/>), 下载与操作系统对应的版本 (Windows x64), 直接双击进行安装, 如图 1-1-15 所示。

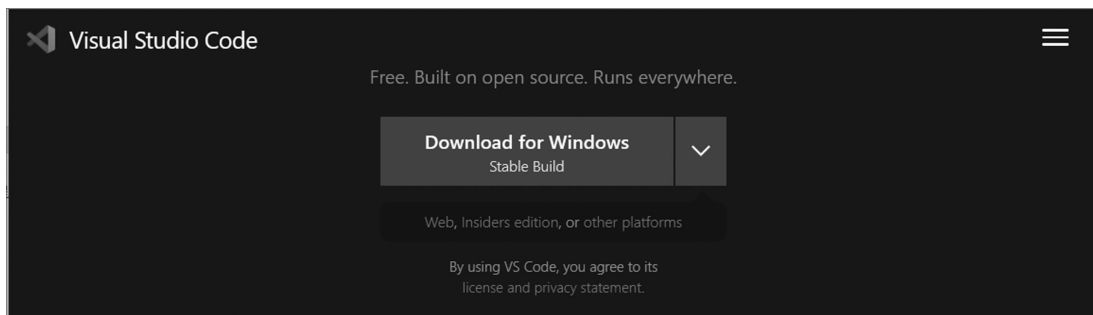


图 1-1-15 下载 Visual Studio Code

子任务 2 创建 Vue3 应用程序

创建 Vue3 应用程序, 在页面中输入 “Hello, Vue3! ”。选择不借助 Vue 构建工具 (Vue-CLI) 构建项目, 直接在项目中引入 Vue.js 库, 这种方式对初学者较为友好。

1. 创建项目

(1) 在 D 盘 (或其他盘) 新建项目文件夹, 本项目文件夹为 PROJECT-1-1, 在 PROJECT-1-1 文件夹中新建 css、images 与 js 三个文件夹。

(2) 打开 VSCode, 选择 “文件” → “打开文件夹” → “PROJECT-1-1” 选项, 项目结构如图 1-1-16 所示。

(3) 添加一个 HTML 文件到项目, 将其命名为 index.html, 如图 1-1-17 所示。

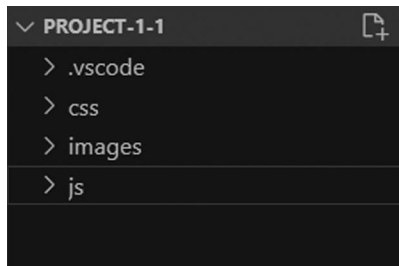


图 1-1-16 项目结构

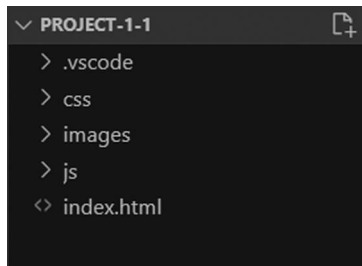


图 1-1-17 添加文件

常用的两种添加文件到项目的方法如下。

① 选择 “文件” → “新建文件” 选项, 在弹出的新建文件窗口中输入文件名称与后缀, 选择存放在当前项目目录下。

② 在资源管理器中将鼠标指针移到项目名称上, 在右侧会出现一排快捷命令按钮, 单击第一个添加文件命令按钮, 自动在当前项目中生成一个新文件, 输入文件名称与后缀即可。

技术小贴士 VSCode 自动生成 HTML 文档结构小妙招

在文档中输入 “!”, 然后按 Tab 键, 就能自动生成 HTML5 文档结构。

2. 引入 Vue.js

将 Vue 3.0 添加到项目的方式主要有以下三种。

第一种方式：以 CDN package 的形式导入，适合于学习或制作原型，代码如下：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>以 CDN package 的形式导入 Vue 3.0</title>
    <script src="https://unpkg.com/vue@next"></script>
  </head>
  <body>
  </body>
</html>
```

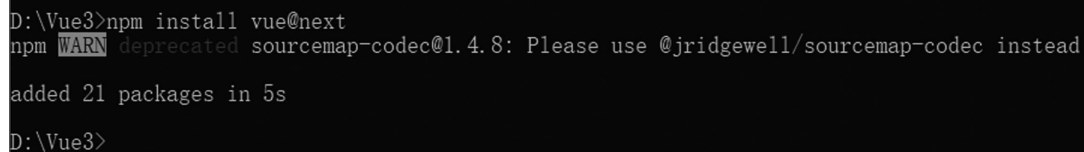
对于生产环境，推荐链接到一个明确的版本号和构建文件，以避免新版本造成不可预期的破坏。例如：

```
<script src="https://unpkg.com/vue@3.2.45"></script>
```

第二种方式：通过 npm（cnpm）工具安装 Vue 3.0，再引入 Vue.js 到页面。

（1）在 D 盘或其他盘新建文件夹，这里命名为 Vue3，用户存放安装的 Vue 3.0 及其依赖文件。

（2）打开命令行窗口，进入 Vue3 文件夹，输入命令 npm install vue@next 并按 Enter 键。如果安装了 cnpm，也可以用 cnpm 代替 npm，如图 1-1-18 所示。



```
D:\Vue3>npm install vue@next
npm WARN deprecated source-map-codec@1.4.8: Please use @jridgewell/source-map-codec instead
added 21 packages in 5s
D:\Vue3>
```

图 1-1-18 安装 Vue3

安装成功后，进入 D 盘“Vue3”→“node_modules”→“vue”→“dist”文件夹中，找到 vue.global.js 文件，这就是需要在页面中引入的 Vue.js 库。

（3）复制 vue.global.js 文件到 project1 项目中的 js 文件夹中，在 index.html 文件中引入 vue.global.js 库，代码如下：

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

第三种方式：通过项目构建工具 Vite 或 Vue-CLI 构建项目，再通过 import 方式导入，这种方式将在下一个项目中进行讲解。

3. 创建 Vue 实例

每个 Vue 应用都是通过用 Vue 函数创建一个新的 Vue 实例开始的。Vue 实例是通过 Vue 函数传入对应的 options 参数创建出的一个实例对象。Vue 实例对应着 ViewModel 层，实例创建后，形成双向绑定：DOM Listeners 和 Data Bindings。DOM Listeners 监听 DOM（View 层）的变化，并从 Model 层获取数据，通过 Data Bindings 更新 DOM（View 层），如图 1-1-19 所示。

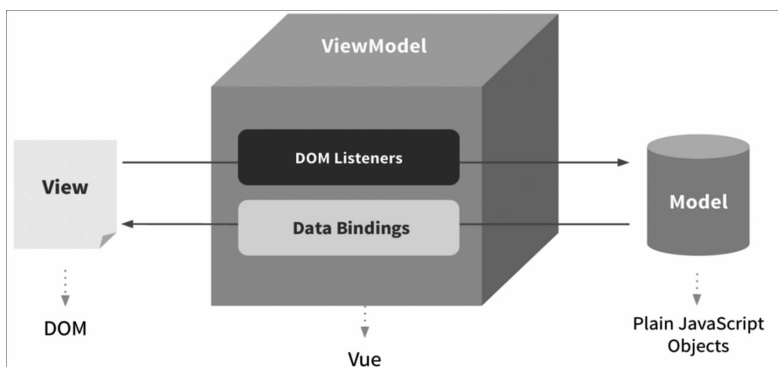


图 1-1-19 Vue 概念综述

(1) 通过 createApp 函数创建一个新的应用实例，在 index.html 页面的 body 标签的最底部添加创建实例的 JavaScript 代码，创建一个 App 实例对象，代码如下：

```
<script>
  const app = Vue.createApp({
    //添加配置选项
  })
</script>
```

(2) 在实例对象中添加绑定数据配置选项，用来存储 Vue 实例或组件中的数值。在实例中数据配置选项有两种形式，一种是对象形式，另一种是函数形式。Vue3 只支持函数形式，代码如下：

```
<script>
  const app = Vue.createApp({ //创建 Vue3 实例
    data() {                  //数据配置选项
      return {
        message: "Hello,Vue3!"
      }
    }
  })
</script>
```

(3) 编写视图代码，实例挂载到视图，绑定到视图中 id 为 “app” 的 DOM 元素上，

代码如下：

```
<body>
  <div id = "app">                                //视图
    {{message}}                                    //插值输出
  </div>
  <script src = "js/vue.global.js"></script>      //加载 Vue.js 库
  <script>
    const app = Vue.createApp({                    //创建 Vue3 实例
      data() {                                     //数据选项
        return {
          message: "Hello,Vue3!"
        }
      }
    })
    app.mount("#app"); //实例挂载到视图,与视图中 id 为“app”的 DOM 元素绑定
  </script>
</body>
```

子任务 3 运行与调试 Vue3 应用程序

在 VSCode 中运行与调试前端应用程序，可以选择“运行”→“启动调试”选项，或直接按 F5 键。也可以安装调试插件（open in browser），利用插件运行与调试前端应用程序，推荐使用后者进行前端应用程序的运行与调试。

1. 安装调试插件

技术小贴士 在 VSCode 中安装插件

在最左侧活动栏中单击扩展命令按钮，在搜索框中输入“open in browser”。选择搜索到的插件，单击“安装”按钮，如图 1-1-20 所示。



图 1-1-20 在 VSCode 中安装插件

2. 运行 Vue3 应用程序

在 index.html 页面的空白处右击，在弹出的快捷菜单中选择 Open In Default Browser 选项，或按 Alt+B 快捷键，运行应用程序。运行结果如图 1-1-21 所示。

Hello,Vue3!

图 1-1-21 运行结果

3. 使用 devtools 工具调试应用程序

在运行界面中右击，在弹出的快捷菜单中选择“检查”选项，或按 F12 键调出调试工具，在调试工具界面中选择 Vue 选项，可以使用 devtools 工具查看数据及各种状态信息，如图 1-1-22 所示。

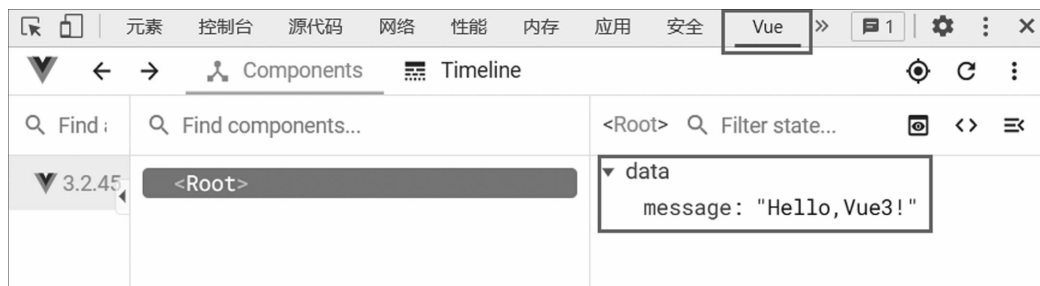


图 1-1-22 用 devtools 调试程序



实践练习

一、填空题

1. Web 前端使用 HTML、CSS 和 JavaScript 作为基础语言，_____用来编写网页的结构、_____用来实现元素的样式与布局的页面效果、_____用来实现页面逻辑、行为、动作等页面交互效果，实现更好的用户体验。
2. _____的诞生标志着 JavaScript 成为与 PHP、Python、Perl、Ruby、JSP 等服务器端语言平起平坐的脚本语言。
3. Vue（发音为 /vju:/，类似 view）是一套用于构建_____渐进式_____框架，由中国人_____开发。
4. 与其他重量级框架不同的是，Vue 的核心库只关注于_____，而且被设计为可以自底向上逐层应用。
5. Vue 3.0 支持_____API 与_____API。
6. Vue 框架采用的模式是_____，MVVM 的主要目的是分离_____层和_____层。
7. _____是一个 Node.js 包管理工具，用来解决 Node.js 代码部署问题。
8. _____是下一代前端开发与构建工具，意在提供开箱即用的配置，

同时它的插件 API 和 JavaScript API 带来了高度的可扩展性，并有完整的类型支持。

二、选择题

1. 下列（ ）不属于 Vue 开发所需的工具。

- A. vue-devtools
- B. VSCode 等开发工具
- C. Node.js
- D. 微信开发者工具

2. npm 包管理器是基于（ ）平台的。

- A. Vue
- B. Babel
- C. Node.js
- D. React

3. 下列选项中，不正确的 Vue3 安装命令是（ ）。

- A. npm install vue@next
- B. cnpm install vue@next
- C. npm install vue
- D. node install vue

4. 下列关于 Vue 的优势说法错误的是（ ）。

- A. 轻量级的框架
- B. 双向数据绑定
- C. 组合式 API
- D. 重量级的框架

三、简答题

1. 简述 Vue 前端框架及其特性。

2. 简述 MVVM 设计模式。

四、实践题

搭建前端开发环境，创建一个简单的 Vue3 项目，在页面上输出“Hello, Vue3, 我来了!”。

任务 1-2 创建 Vue 单页面应用



任务导入

当下企业项目开发一般采用前后端分离的方式进行 Web 架构的设计，对前端开发环境的搭建提出了更高的要求。一个完整的前端开发环境应该具备预编译模板、注入依赖、合并压缩资源、分离开发和生产环境及提供一个模拟的服务器端环境等功能。

在页面中通过<script>标签引入 vue.js 库文件，这种方式仅适用于简单的案例，不适合企业级的应用开发。此时需要借助项目构建工具，帮助开发者快速构建一个适用于实际项目的 Vue 环境，让开发者能够快速地进行应用开发。

当前，构建 Vue 项目常用的两种方式是 Vue-CLI（脚手架）与 Vite。这两个都是提供基本项目脚手架和开发服务器的构建工具。

本任务将带领读者使用 Vite 工具快速构建 Vue3 项目，创建 Vue 单页面应用，开启 Vue3 项目实战之旅。

任务目标

知识目标	能力目标	教学重点	教学难点
◎阐述 Vite 及其特性。 ◎描述 Vue 项目的结构与项目运行流程。 ◎解释 SPA	◎能使用 Vite 快速构建 Vue3 项目，创建一个单页面 Web 应用	◎利用 Vite 构建工具构建 Vue3 项目	◎SPA 概念的理解与应用

素质目标

了解软件开发技术、开发工具的更迭速度及优胜劣汰法则，培养学生的忧患意识，激发学生的学习动力。



知识准备

知识点 1 Vite 简介

Vite (法语意为快速的) 是一种新型前端构建工具，由 Vue 的作者尤雨溪开发，意在提供开箱即用的配置，同时它的插件 API 和 JavaScript API 带来了高度的可扩展性，并有完整的类型支持，能够显著提升前端开发体验，目前只支持 Vue 3.x。它主要由以下两部分组成。

(1) 一个开发服务器，它基于原生 ES 模块，提供了丰富的内建功能，如速度快到惊人的模块热更新 (HMR)。

(2) 一套构建指令，它使用 Rollup 打包代码，并且它是预配置的，可输出用于生产环境的高度优化过的静态资源。

作为一个基于浏览器原生 ESM 的构建工具，Vite 省略了开发环境的打包过程，利用浏览器去解析 imports，在服务器端按需编译返回。同时，在开发环境拥有速度快到惊人的模块热更新，且热更新的速度不会随模块增多而变慢。因此，使用 Vite 进行开发要比 Webpack 快 10 倍左右。

由于 Vite 使用了浏览器原生的 ES 模块导入功能，但 IE11 并不支持 ES 的模块导入，因此基于 Vite 开发项目，不能使用 IE11 浏览器，其他主流的浏览器均支持 ES 模块的导入功能。



1-2 创建 Vue 单页面应用 (理论)-rxtrmnd

知识点 2 Vue 项目结构

使用 Vite 构建项目生成的项目结构如图 1-2-1 所示。

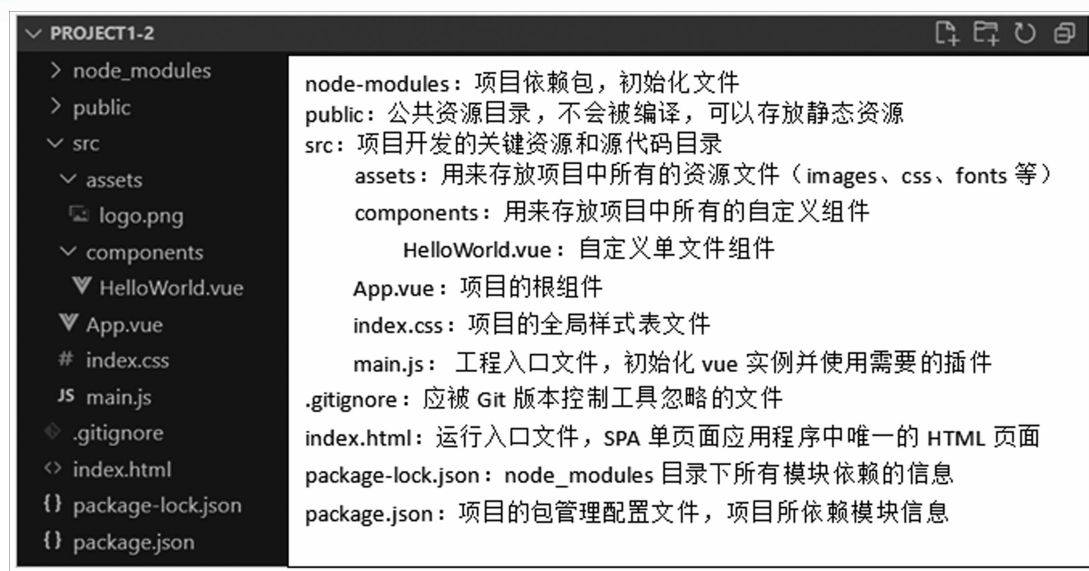


图 1-2-1 项目结构

主要的目录与文件说明如表 1-2-1 所示。

表 1-2-1 Vite 项目初始目录与文件

名 称	说 明
node_modules	项目依赖文件夹, 可以通过 npm install 下载或更新
public	公共资源目录, 用于存放项目用到的一些静态资源, 在最后打包时会直接放入 dist 文件夹中
src	项目开发的关键目录和源代码目录, 是主要的工作空间。src 目录下包含 assets、components、App.vue、index.css、main.js 等目录和文件
assets	放置静态资源, 包括公共的 css 文件、js 文件、iconfont 字体文件、img 图片文件以及其他资源类文件, 在最后打包时会直接当成一个模块打包到 js 文件夹中
components	放置通用模块组件, 如弹出框、发送手机验证码、图片上传、页头部、页尾部等, 将它们作为通用组件, 避免重复工作
App.vue	项目唯一的根组件、入口组件, 其他组件会被插入此组件中, 此组件再插入 index.html 文件中, 形成单页面应用
main.js	整个项目工程的入口文件, 用于全局配置, 主要作用是初始化 Vue 实例并引入全局使用的库、公共的样式和方法、设置路由等
index.html	项目的运行入口文件, SPA 单页面应用程序中唯一的 HTML 页面, 被注入资源的模板 HTML 文件

知识点 3 项目运行流程

在工程化的项目中, Vue 要做的事很单一, 即通过 main.js 把 App.vue 渲染到

index.html 文件中, 其他组件加载到 App.vue 中。

初始化 Vite 项目的运行流程为 index.html→main.js→App.vue→HelloWorld.vue。

(1) 在 index.html 中预留 el 区域, el 区域将被注入 App.vue 文件模板内容, 代码如下:

```
<body>
  <!--id 为 App 的 div 元素,Vue 实例要挂载的元素,将来注入 App.vue 模板内容到此元素区域-->
  <div id="app"> <!-- el 区域 -->
    <!-- 在这里注入 App.vue 模板内容-->
  </div>
  <script type="module" src="/src/main.js"></script>
</body>
```

(2) 在 App.vue 根组件中编写模板内容, 模板内容将被注入 index.html 模板文件中, 代码如下:

```
<template>
  
  <!-- 加载 HelloWorld 组件,HelloWorld 组件模板内容将会注入此处 --><HelloWorld
  msg="Hello Vue 3.0 + Vite" />
</template>
```

(3) 在 App.vue 中引入 HelloWorld.vue 组件, HelloWorld.vue 组件模板内容将被加载到此组件处, 代码如下:

```
<template>
  <h1>{{ msg }}</h1>
  <button @click="count++">count is: {{ count }}</button>
  <p> Edit <code> components/HelloWorld.vue</code> to test hot module
  replacement.</p>
</template>
```

(4) main.js 进行渲染, 把 App.vue 中的模板内容渲染到 index.html 模板 id 为 app 的元素区域中, 代码如下:

```
//从 Vue 中按需导入 createApp 函数,用户创建“单页面应用程序实例”
import { createApp } from 'vue'
//导入待渲染的 App 组件
import App from './App.vue'
//导入全局样式表
import './index.css'
//调用 createApp 函数创建“单页面应用程序实例”
//把 App 组件作为参数传给 createApp 函数,表示要把 App 渲染到 index.html 页面上
```

```
//mount 用来指定 Vue 实际控制的区域,内容将注入 index.html 页面中 id 为 app 的元素区域中  
createApp(App).mount('#app')
```

知识点 4 单页面应用

单页面应用 (single-page application, SPA) 是一种特殊的 Web 应用,也是 Vue 所创建的应用。它将所有的活动局限于一个 Web 页面,仅在该 Web 页面初始化时加载相应的 HTML、JavaScript 和 CSS。一旦页面加载完成,SPA 不会因为用户的操作而进行页面的重新加载或跳转,取而代之的是利用 JavaScript 动态地变换 HTML 的内容,从而实现 UI 与用户的交互。由于避免了页面的重新加载,SPA 可以提供较为流畅的用户体验。

整个应用程序只有唯一完整的 HTML 页面,而其他所谓的页面其实都是组件片段,切换页面也只是切换一个 HTML 中显示不同的组件片段。

(1) SPA 的优点主要有如下几个方面。

①良好的交互体验:局部渲染,每个部分都是单独的模块,避免了不必要的跳转和重复渲染。

②架构清晰:前后端分离,前端负责 View,后端负责 Model,各司其职。

③减轻服务器的压力:服务器只需要提供数据,不需要负责前端的展示逻辑和页面合成,提高了性能。

(2) SPA 的缺点主要有如下几个方面。

①对搜索引擎优化 (search engine optimization, SEO) 并不友好:所有的内容都在一个页面中进行动态替换,不会作为 HTTP 请求中的一部分发送给服务器,在 SEO 上有着天然的弱势。

②可能出现首屏加载时间过长的问题:首屏加载时需要将 js、css、图片等资源统一加载,部分页面按需加载,当资源比较多、比较大时,加载速度比较慢。

③页面复杂度提高:后端只提供数据而不再负责前端的展示逻辑和页面合成,所以这些展示逻辑和页面合成都需要在前端进行编写 (前进、后退等),所以会大大提高页面的复杂性和逻辑的难度。

任务设计

本任务将使用 Vite 工具快速构建 Vue3 项目,创建 Vue 单页面 Web 应用程序,输出自己最喜欢、最敬佩的 IT 大咖和他 (她) 想对我们说的一句话。



1-2 创建 Vue
单页面应用
(实操)-akjwaz

任务实施

本任务实施分为 3 个子任务,分别是快速构建 Vue3 项目、修改项目和运行与调试项目。

子任务 1 快速构建 Vue3 项目

使用 Vite 快速构建 Vue 项目之前,确保已经安装好了 node.js,且版本不能低于 12。

1. 构建项目

在命令行切换进入项目要保存的目录，输入命令 `npm init vite-app project-1-2` 并按 Enter 键，其中，`project-1-2` 是项目的名称，如图 1-2-2 所示。

```
D:\workspace\vue3>npm init vite-app project-1-2
Scaffolding project in D:\workspace\vue3\project-1-2...

Done. Now run:

  cd project-1-2
  npm install (or `yarn`)
  npm run dev (or `yarn dev`)
```

图 1-2-2 使用 Vite 构建项目

2. 安装依赖

进入项目目录 `project-1-2` 中，在命令行中输入命令 `npm install` 或 `yarn` 并按 Enter 键，安装项目所需要的依赖，如图 1-2-3 所示。

3. 运行项目

在命令行输入命令 `npm run dev` 或 `yarn dev` 并按 Enter 键，搭建脚手架环境并运行项目，如图 1-2-4 所示。复制 Local 后的地址 `http://localhost:3000/` 到浏览器地址栏并按 Enter 键，显示项目运行页面，如图 1-2-5 所示。

```
D:\workspace\vue3\project-1-2>npm install
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer
@rollup/plugin-terser
npm WARN deprecated source-map-codec@1.4.8: Please use @jridgewell/source-map-codec instead
added 290 packages, and audited 291 packages in 40s
```

图 1-2-3 安装依赖

```
D:\workspace\vue3\project-1-2>npm run dev
> project-1-2@0.0.0 dev
> vite

[vite] Optimizable dependencies detected:
vue

Dev server running at:
> Network: http://192.168.1.3:3000/
> Local: http://localhost:3000/
```

图 1-2-4 运行项目



图 1-2-5 初始化运行结果

在构建过程中有可能会遇到以下两个问题。

(1) `npm install` 命令很慢或卡住不动。因为 `npm` 服务器在境外会比较慢，可以换成国内淘宝镜像 `cnpm`，先安装 `cnpm`，然后使用 `cnpm install` 命令安装依赖。

(2) `npm run dev` 命令提示“`vite` 不是内部命令或外部命令”。可能还没有安装 `Vite`，可以通过命令 `npm install vite` 进行安装。安装完成后再运行命令 `npm run dev`。

子任务 2 修改项目

添加一张图片到项目的 assets 目录中，修改根组件 App.vue 部分代码。

1. 添加图片到 assets 目录

打开 VSCode，选择“文件”→“打开文件夹”→“project-1-2”选项，在 VSCode 中打开项目 project-1-2。把需要在页面中显示的图片拖到项目的 src/assets 目录中，也可以直接进行复制。

2. 修改 App.vue 代码

(1) 删除 App.vue 文件 template 标签内的全部代码，添加如下代码：

```
<template>
  
  <p>如果你想要在某领域出类拔萃，就必须全身心投入</p>
</template>
```

(2) 删除 App.vue 文件 script 标签内的全部代码，添加如下代码：

```
<script>
  export default {
    name: 'App',
  }
</script>
```

子任务 3 运行与调试项目

在 VSCode 中也可以开启命令终端，效果与 Windows 系统的命令终端一致，且更为方便。选择“终端”→“新建终端”选项，将在 VSCode 下部显示调试终端面板。执行命令 npm run dev，运行修改后的项目，如图 1-2-6 所示。

复制 Local 后的地址 http://localhost:3000/ 到浏览器地址栏并按 Enter 键，显示页面效果，如图 1-2-7 所示。也可以直接在链接上按 Ctrl 键并单击，在浏览器中快速打开链接。



图 1-2-6 VSCode 终端运行项目



图 1-2-7 运行结果



实践练习

一、填空题

1. _____ 是一种新型前端构建工具，由 Vue 的作者尤雨溪开发，意在提供开箱即用的配置，能够显著提升前端开发体验，目前只支持_____。
2. _____ 目录是公共资源目录，用于存放项目用到的一些静态资源，如图片等，在最后打包时会直接放入 dist 文件夹中。
3. _____ 目录放置静态资源，包括公共的 css 文件、js 文件、iconfont 字体文件、img 图片文件以及其他资源类文件，在最后打包时会直接当成一个模块打包到 js 文件夹中。
4. _____ 是项目唯一的根组件。
5. _____ 是整个项目的工程入口文件，用于全局配置，主要作用是初始化 Vue 实例并引入全局使用的库、公共的样式和方法，设置路由等。
6. _____ 目录是项目开发的关键目录和源代码目录，是主要的工作空间，其中包含 assets、components、App.vue、index.css、main.js 等目录和文件。
7. Vite 初始化项目运行流程为 _____ → _____ → _____。
8. 单页面 Web 应用简称_____。
9. ECMAScript 6.0 简称_____，是 JavaScript 语言的下一代标准，于 2015 年 6 月发布。
10. 模块功能主要由两个命令构成：_____ 命令用于规定模块的对外接口_____，_____ 命令用于引入其他模块提供的功能。

二、选择题

1. 下列 () 不属于 Vue 前端构建工具。
 - A. webpack
 - B. vite
 - C. yarn
 - D. vue-devtools
2. 在使用构建工具生成的项目中，下列 () 目录是公共资源目录，不会被编译。
 - A. src
 - B. assets
 - C. public
 - D. components
3. 下列选项中，() 是整个项目工程的入口文件，用于全局配置和初始化。
 - A. App.vue
 - B. main.js
 - C. package.json
 - D. index.html
4. 下列关于使用 Vite 构建工具构建项目运行流程的说法，正确的是 ()。
 - A. index.html→main.js→App.vue→HelloWorld.vue
 - B. main.js→index.html→App.vue→HelloWorld.vue
 - C. index.html→App.vue→main.js→HelloWorld.vue
 - D. App.vue→main.js→HelloWorld.vue→index.html

三、简答题

1. 简述 Vite 及使用 Vite 创建并运行 Vue 项目的命令。

2. 简述 Vite 项目的运行流程。
3. 什么是 SPA? 简述 SPA 的优势和缺点。

四、实践题

1. 构建一个 Vue3 项目，在页面上显示“Hello, Vue3, 我来了!”。
2. 构建一个 Vue3 项目，在页面上显示神舟十八号 3 位航天员的照片与姓名，并对神舟十八号乘组人员说一句祝福语。