

# 第1章 Vue 基础入门

Vue.js（以下简称 Vue）是前端的流行框架之一，与 Angular 和 React 相比，Vue 框架在实现上更容易理解，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 框架成为前端开发者的必备技能。本章将会介绍 Vue 框架的基本概念，并对 Vue 的使用进行讲解。

## 【教学导航】

|      |   |
|------|---|
| 学习目标 | <ol style="list-style-type: none"><li>1. 了解 Vue 的核心设计思想</li><li>2. 掌握 Vue 开发环境搭建</li><li>3. 掌握 Vue 开发和调试工具的使用</li><li>4. 掌握 Vue 项目的创建</li></ol> |
| 教学方式 | 本章主要以概念讲解、操作实践为主  |
| 重点知识 | <ol style="list-style-type: none"><li>1. 掌握 Vue 的开发环境搭建</li><li>2. 了解 Vue 的核心设计思想</li></ol>   |
| 关键词  | MVVM 设计思想、环境搭建、开发工具的使用、调试工具的使用  |

## 1.1 初识 Vue

### 1.1.1 前端技术的发展

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 主要用来编写网页的结构，例如<a></a>表示超链接。CSS 样式包括颜色、大小、字体等，实现漂亮、布局合理的页面效果。JavaScript 的功能主要包括实现页面逻辑、行为、动作等，用来动态操作元素的属性，主要是为页面提供交互效果，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 DOM（文档对象模型），并处理浏览器的兼容问题，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 的核心理念是开发者只需写很少的代码，

就可以实现更多的功能。它通过对 JavaScript 代码的封装，使得 DOM、事件处理、动画效果、Ajax 交互等功能的实现变得更加简洁、方便，有效地提高了项目开发效率。

随着移动端技术的发展，前端技术被逐渐应用到移动端开发中，用来构建单页应用。单页应用是前端开发的一种形式，在切换页面的时候，不会刷新整个页面，而是通过 Ajax 异步加载新的数据，改变页面的内容。为了方便地开发这类复杂的应用，市面上出现了 Angular、React、Vue 等框架。Vue 通过虚拟 DOM 技术来减少对 DOM 的直接操作；通过尽可能简单的 API 来实现响应的数据绑定，支持单向和双向数据绑定。组件化的特性提高了开发效率、使代码更容易复用，提高了项目的可维护性，便于团队的协同开发。

## 1.1.2 什么是 Vue

Vue（读音/Vju:/，类似于 View）是一套用于构建用户界面的渐进式框架，与其他大型框架相比，Vue 被设计为可以自底向上逐层应用。其他大型框架往往一开始就对项目的技术方案进行强制性的要求，而 Vue 更加灵活，开发者既可以选择使用 Vue 来开发一个全新项目，也可以将 Vue 引入到一个现有的项目中。

另一方面，当 Vue 与现代化的工具链以及各种支持类库结合使用时，也完全能够为复杂的单页应用提供驱动。工具链是指在前端开发过程中用到的一系列工具，例如，使用脚手架工具创建应用，使用依赖管理工具安装依赖包，以及使用构建工具进行代码编译等。

Vue 的数据驱动是通过 MVVM（Model-View-ViewModel）模式来实现的，其基本工作原理如图 1-1 所示。

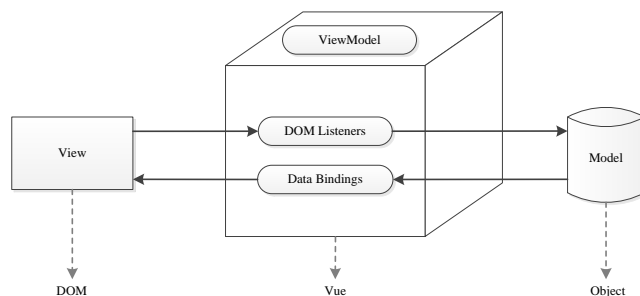


图1-1 Vue 的基本工作原理

从图 1-1 中可以看出，MVVM 主要包含 3 个部分，分别是 Model、View 和 ViewModel。Model 指的是数据部分，主要负责业务数据；View 指的是视图部分，即 DOM 元素，负责视图的处理。ViewModel 是连接视图与数据的数据模型，负责监听 Model 或者 View 的修改。

在 MVVM 中，数据 (Model) 和视图 (View) 是不能直接通信的，视图模型 (ViewModel) 就相当于一个观察者，监控着双方的动作，并及时通知进行相应操作。当 Model 发生变化的时候，ViewModel 能够监听到这种变化，并及时通知 View 做出相应的修改。反之，当

View 发生变化时，ViewModel 监听到变化后，通知 Model 进行修改，实现了视图与模型的互相解耦。

### 1.1.3 Vue 的优势

目前市场三大前端主流框架分别是 Angular、React 和 Vue.js。Vue 之所以被开发者青睐，主要是 Vue 秉承了 Angular 和 React 框架两者的优势，并且 Vue 的代码简洁、上手容易，在市场上也得到大量应用。下面将对 Vue 的特性进行简单介绍。

#### 1. 轻量级

Angular 的学习成本高，使用起来比较复杂，而 Vue 相对简单、直接，所以 Vue 使用起来更加友好。

#### 2. 数据绑定

Vue 是一个 MVVM 框架，即数据双向绑定，即当数据发生变化的时候，视图也就发生变化，当视图发生变化的时候，数据也会跟着同步变化，这也算是 Vue 的精髓之处，尤其是在进行表单处理时，Vue 的双向数据绑定非常方便。

#### 3. 指令

指令主要包括内置指令和自定义指令，以“v-”开头，作用于 HTML 元素。指令提供了一些特殊的特性，将指令绑定在元素上时，指令会给绑定的元素添加一些特殊的行为。例如，v-bind 动态绑定指令、v-if 条件渲染指令、v-for 列表渲染指令等。

#### 4. 插件

插件用于对 Vue 框架功能进行扩展，通过 MyPlugin.install 完成插件的编写，简单配置后就可以全局使用。常用的扩展插件有 vue-router、Vuex 等。

Vue 很多特性与 Angular 和 React 有着相同的地方，但是也有着性能方面的差别。Vue 使用基于依赖追踪的观察系统并且使用异步队列更新，所有的数据都是独立触发的，提高了数据处理能力。

React 和 Vue 的中心思想一切都是组件，组件之间可以实现嵌套。React 采用了特殊的 JSX 语法，Vue 中也推崇编写以\*.vue 后缀命名的文件格式，对文件内容都有一些规定，两者需要编译后使用。

值得一提的是，Vue 在模板中提供了指令，过滤器等，可以非常方便和快捷的操作 DOM。推荐将 Vue 使用在具有复杂交互逻辑的前端应用中，以确保用户的体验效果。

## 1.2 Vue 开发环境

为了快速上手 Vue 项目开发，本节将对 Vue 的开发环境以及常用工具的使用进行讲解，并通过 Hello World 案例演示 Vue 的基本使用。

### 1.2.1 Visual Studio Code 编辑器

Visual Studio Code (VS Code) 是由微软公司推出的一款免费、开源的编辑器，推出之后便很快流行起来，深受开发者的青睐。作为前端开发人员来说，一个强大的编辑器可以让开发变得简单、便捷、高效。本书选择基于 VS Code 编辑器进行讲解。

VS Code 编辑器具有如下特点。

(1) 轻巧极速，占用系统资源较少。

(2) 具备语法高亮显示、智能代码补全、自定义快捷键和代码匹配等功能。

(3) 跨平台。不同的开发人员为了工作需要，会选择不同平台来进行项目开发工作，这样就一定程度上限制了编辑器的使用范围。VS Code 编辑器不仅跨平台（支持 Mac、Windows 以及 Linux），使用起来也非常简单。

(4) 主题界面的设计比较人性化。例如，可以快速查找文件直接进行开发，可以通过分屏显示代码，主题颜色可以进行自定义设置（默认是黑色），也可以快速查看最近打开的项目文件并查看项目文件结构。

(5) 提供了丰富的插件。VS Code 提供了插件扩展功能，用户根据需要自行下载安装，只需在安装配置成功之后，重新启动编辑器，就可以使用此插件提供的功能。

### 1.2.2 Vue 的下载和引入

Vue 目前的最新版本是 2.x，从 Vue 官方网站可以获取下载地址，如图 1-2 所示。



图1-2 获取 Vue

从图 1-2 中可以看出，Vue 的核心文件有两种版本，分别是开发版本（vue.js）和生产版本（vue.min.js）。生产版本是压缩后的文件。为了方便学习，推荐选择开发版本。将 vue.js 文件下载后，打开文件，在代码开头的注释中查看版本号，如下所示。

```
/*!  
 * Vue.js v2.6.10  
 * (c) 2014-2019 Evan You  
 * Released under the MIT License.  
 */
```

在上述代码中，2.6.10 就是 Vue 核心文件的版本号。

当在 HTML 网页中使用 Vue 时，使用<script>标签引入 vue.js 即可，示例代码如下：

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

上述代码表示引入当前路径下的 vue.js 文件。

### 1.2.3 git-bash 命令行工具

在进行 Vue 开发时，我们经常会使用一些命令，如 npm（包管理器）、vue-cli（脚手架），这些命令需要在命令行下使用。git-bash 是 git（版本管理器）中提供的一个命令行工具，外观类似于 Windows 系统内置的 cmd 命令行工具，但用户体验更友好。在实际开发中，经常会使用 git-bash 工具代替 cmd。下面我们来讲解 git-bash 的安装步骤。

(1) 打开 git for windows 官网，下载 git 安装包，如图 1-3 所示。

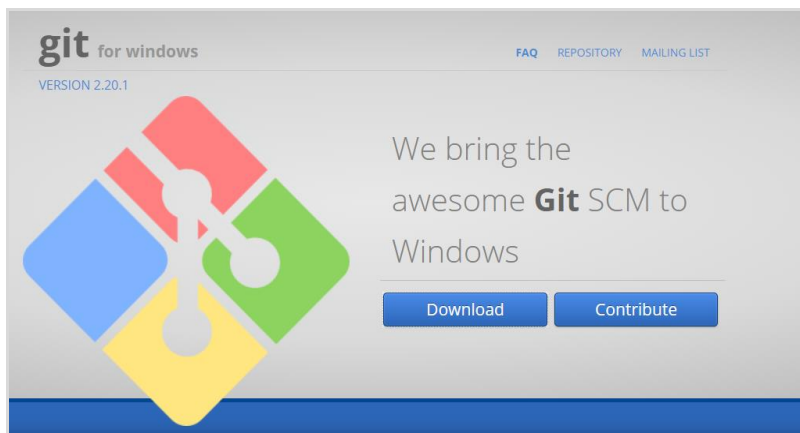


图1-3 git 下载网站

(2) 双击下载后的安装程序，进行安装，如图 1-4 所示。

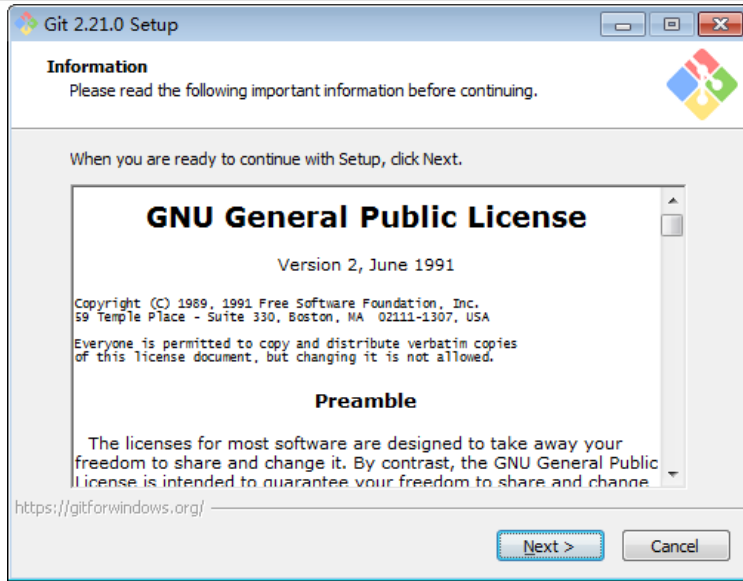


图1-4 安装协议

- (3) 单击“Next”按钮，根据提示进行安装，全部使用默认值即可。
- (4) 安装成功后，启动 git-bash，如图 1-5 所示。

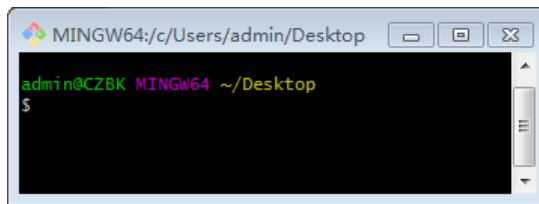


图1-5 启动 git-bash

## 1.2.4 Node.js 环境

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，它可以让 JavaScript 运行在服务器端。接下来我们就对 Node.js 的下载和安装进行详细讲解。

- (1) 打开 Node.js 官方网站，找到 Node.js 下载地址，如图 1-6 所示。

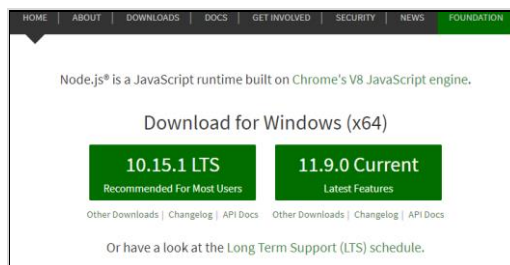


图1-6 Node.js 官网

从图 1-6 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；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。这里选择 LTS 版本进行下载即可。

(2) 双击安装包进行安装，如图 1-7 所示。

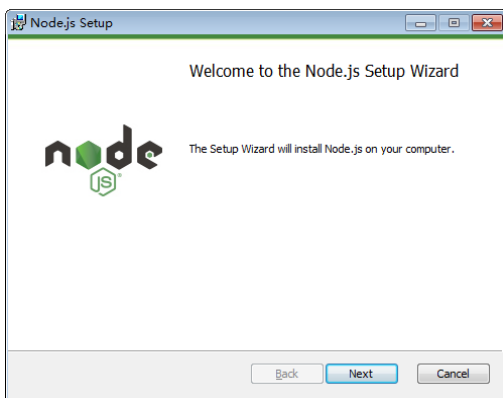


图1-7 安装界面

(3) 安装过程全部使用默认值。安装完成后，打开 git-bash 命令行工具，查看 Node.js 版本信息，如图 1-8 所示。

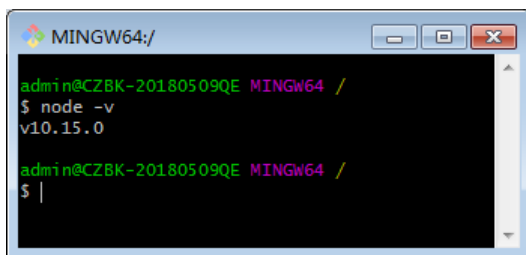


图1-8 查看 Node.js 版本

(4) 将 Node.js 安装完成后，下面我们通过代码演示 Hello World 程序的编写。创建 C:\vue\chapter01 目录，在该目录中创建 helloworld.js 文件，编写如下代码：

```
console.log('Hello World')
```

(5) 保存文件后，执行如下命令，启动 Hello World 程序：

```
node helloworld.js
```

(6) 上述代码执行后，输出结果如图 1-9 所示。

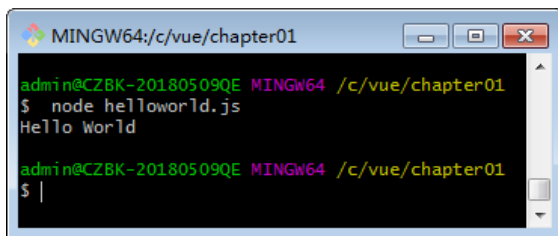




图1-9 HelloWorld 程序

(7) Node.js 还提供了交互式环境 REPL，类似 Chrome 浏览器的控制台，可以在命令行中直接输入 JavaScript 代码来执行。在命令行中执行 node 命令，即可进入交互模式，如图 1-10 所示。

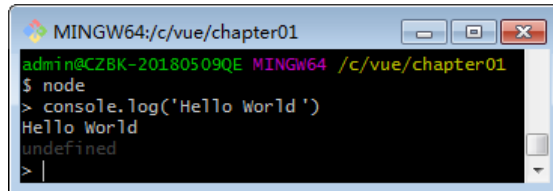


图1-10 REPL 交互式环境

(8) 若要从交互模式中退出，可以输入“.exit”并按 Enter 键，或者按两次 Ctrl+C 组合键来退出。

## 1.2.5 npm 包管理工具

npm (Node.js Package Manager) 是一个 Node.js 的包管理工具，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的需求。

- (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。

npm 提供了快速操作包的命令，只需要简单命令就可以很方便地对第三方包进行管理，下面列举 npm 中的常用命令。

- npm install: 安装项目所需要的全部包，需要配置 package.json 文件。
- npm uninstall: 卸载指定名称的包。
- npm install 包名: 安装指定名称的包，后面可以跟参数“-g”表示全局安装，“--save”表示本地安装；
- npm update: 更新指定名称的包；
- npm start: 项目启动；通过 CDN 方式引入 Vue，可以缓解服务器的压力，加快文件的下载速度。目前，网络上有很多免费的 CDN 服务器可以使用
- npm run build: 项目构建。

多学一招：

由于 npm 的服务器在国外，使用 npm 下载软件包的速度非常慢，为了提高下载速度，推荐读者切换成国内的镜像服务器来使用。以淘宝 NPM 镜像为例，使用如下命令设置即可切换。

```
npm config set registry https://registry.npm.taobao.org
```



## 1.2.6 Chrome 浏览器和 vue-devtools 扩展

浏览器是开发和调试 Web 项目的工具，目前市面上主流的浏览器都有自己的优点和缺点，本书选择基于使用量较多的 Chrome 浏览器进行讲解。

vue-devtools 是一款基于 Chrome 浏览器的扩展，用于调试 Vue 应用，只需下载官方压缩包，配置 Chrome 浏览器的扩展程序即可使用。下面我们简单介绍一下安装流程。

(1) 下载 vue-devtools-5.1.1.zip 压缩包到本地。

(2) 将压缩包进行解压，然后在命令行中切换到解压好的 vue-devtools-5.1.1 目录，输入以下命令进行依赖安装：

```
npm install
```

(3) 构建 vue-devtools 工具插件，执行命令如下。

```
npm run build
```

(4) 将插件添加至 chrome 浏览器。单击浏览器地址栏右边的“⋮”按钮，在弹出的菜单中选择“更多工具”→“扩展程序”，如图 1-11 所示。

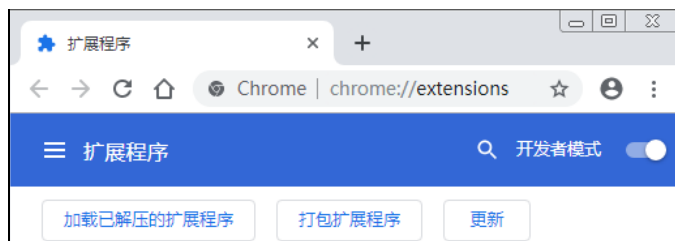


图1-11 扩展程序页面

在图 1-11 所示的界面中，单击“加载已解压的扩展程序”按钮，此时会弹出选择框，需要用户选择扩展程序目录。找到 vue-devtools-5.1.1/shells/chrome 目录，将其添加到扩展程序中。

(5) 配置完成后，可以看到当前 vue-devtools 工具的信息，并在 Chrome 浏览器窗口的右上角会显示 Vue 的标识，如图 1-12 所示。

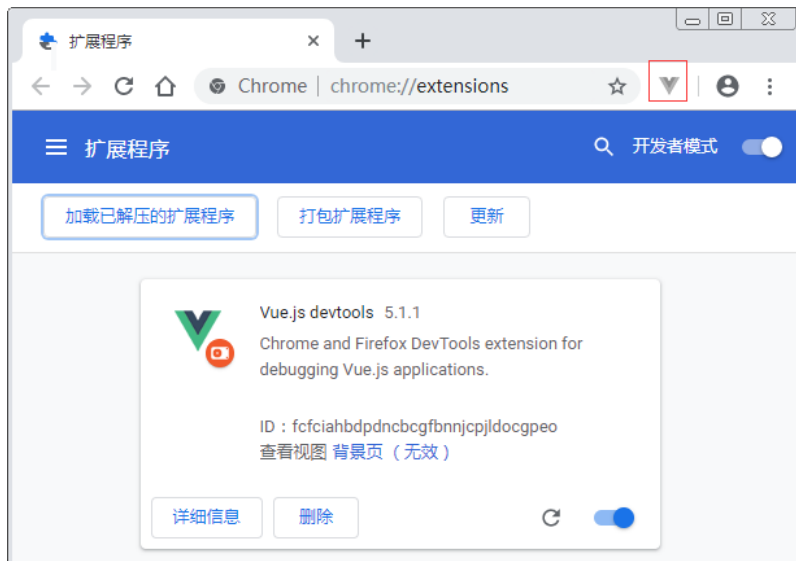


图1-12 vue-devtools 安装成功界面

## 1.2.7 Hello World 案例

学习了 Vue 的几种引用方式后，下面将使用 Vue 在页面中输出“Hello Vue.js”，开启第一个 Vue 案例的体验之旅。具体如例 1-1 所示。

### 【例1-1】

(1) 在 C:\vue\chapter01 目录中创建 demo01.html 文件，具体代码如下：

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>我是第一个Vue.js 案例</title>
6   <script src="vue.js"></script>
7 </head>
8 <body>
9   <div id="app"></div>
10 </body>
11 </html>
```

上述代码中，第 6 行引入了 vue.js 核心文件，引入后就会得到一个 Vue 构造器，用来创建 Vue 实例。第 9 行为元素设置了 id，用来作为 Vue 实例控制的元素。

(2) 在</body>结束标签前编写如下代码，创建 Vue 实例：

```
1 <script>
```

```
2 var vm = new Vue({
3   el: '#app',
4   data: {
5     msg: 'Hello Vue.js'
6   }
7 })
8 </script>
```

上述代码中，第 2 行创建了一个 Vue 实例，保存为 vm（含义为 ViewModel）；第 3 行的 el 表示当前 vm 实例要控制的页面区域，即 id 为 app 的元素；第 4 行的 data 属性用来存放 el 中要用到的数据；在第 5 行设置了 data 对象的属性 msg 为“Hello Vue.js”。

(3) 通过 Vue 提供的“{{ }}”插值表达式，把 data 数据渲染到页面。修改页面中 id 为 app 的根容器的代码，如下所示。

```
1 <div id="app">
2   <!-- 将msg绑定到p元素 -->
3   <p>{{msg}}</p>
4 </div>
```

(4) 通过浏览器访问 demo01.html，运行结果如图 1-13 所示。

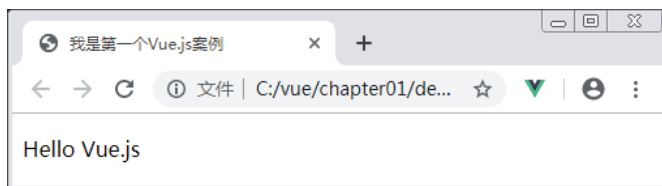


图1-13 输出 Hello Vue.js

## 1.3 webpack 打包工具

webpack 是一个模块打包工具，可以把前端项目中的 js、cs、scss/less、图片等文件都打包在一起，实现自动化构建，给前端开发人员带来了极大的便利。本节将针对如何在 webpack 中构建 Vue 项目进行讲解。

### 1.3.1 安装 webpack

在命令行中执行如下命令即可安装 webpack:

```
npm install webpack@4.27.x webpack-cli@3.3.x -g
```

在上述命令中，4.27.x 是 webpack 版本号，表示安装 4.27.x 范围内的最新版本，webpack-cli 是命令行工具，“-g”表示全局安装。

安装完成后，执行“webpack -v”命令，查看 webpack 版本，如下所示：

```
webpack -v  
4.27.1
```

另外，如果希望卸载 webpack，可以执行如下命令来卸载：

```
npm uninstall webpack -g
```

**小提示：**

4.x 版本的 webpack 需要安装 webpack-cli 命令行工具，而 3.x 版本的 webpack 打包工具已经集成了命令行工具。

## 1.3.2 webpack 简单使用

在安装 webpack 之后，我们通过例 1-2 演示 webpack 的简单使用。

**【例1-2】**

(1) 创建 C:\vue\chapter01\demo02 目录，作为项目目录。

(2) 在 demo02 目录中创建 example.js 文件，具体代码如下：

```
1 function add(a, b) {  
2   return a + b  
3 }  
4 console.log(add(1 , 2))
```

上述代码用于计算两个数之和，在控制台中输出计算结果。

(3) 在 demo02 目录下执行如下命令，进行打包操作：

```
webpack example.js -o app.js
```

执行上述命令后，就会编译 example.js 文件，将编译后的结果保存为 app.js 文件。

(4) 创建 example.html 文件，引入编译后的 app.js 文件，具体代码如下：

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

(5) 在浏览器中打开 example.html，运行结果如图 1-14 所示。

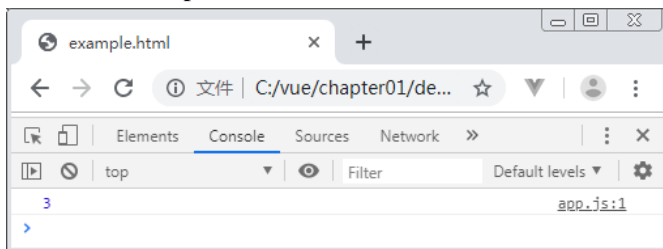


图1-14 webpack 简单使用

从图 1-14 可以看出，控制台输出的打印结果为 3，说明此时已经将 example.js 文件打包为 app.js 文件。

### 1.3.3 构建 Vue 项目

在 Vue 项目开发时，为了提高加载时间和性能，webpack 打包工具会将项目中的文件转为浏览器可以读取的静态文件。下面我们来演示如何创建一个简单的 Vue 项目。

(1) 在创建项目之前应先完成 vue-cli 脚手架工具的安装。脚手架工具可以直接生成一个项目的整体架构，帮助开发者搭建 Vue.js 的基础代码。执行命令如下：

```
npm install vue-cli@2.9.x -g
```

将 vue-cli 安装完成后，可以执行“vue-cli -V”命令，查看安装的版本号。

(2) 打开 C:\vue\chapter01 目录，执行如下命令初始化 vue 项目：

```
vue init webpack myapp
```

在上述命令中，myapp 表示项目名称，可以根据需要自定义名称。程序会自动在当前目录下创建 myapp 子目录作为项目目录。webpack 表示项目的模板。

(3) 在创建项目时，程序会询问项目的一些配置选项，直接按回车使用默认值即可。关于 vue-cli 的使用具体会在第 6 章中详细讲解。

接下来分析 myapp 项目的目录结构，具体解释如表 1-1 所示。

表1-1 myapp 目录结构

| 目录结构         | 说明                |
|--------------|-------------------|
| build        | 项目构建（webpack）相关代码 |
| config       | 配置文件目录            |
| node_modules | 依赖模块              |
| src          | 源码目录              |
| static       | 静态资源目录            |
| test         | 初始测试目录            |
| index.html   | 首页入口文件            |
| package.json | 项目配置文件            |
| README.md    | 项目说明文档            |

(4) 切换到项目目录，然后启动服务，具体命令如下：

```
cd myapp  
npm run dev
```

执行上述命令后，如果启动成功，会看到如下提示信息：

```
Your application is running here: http://localhost:8080
```

上述信息表示当前应用已经启动，可以通过 <http://localhost:8080> 来访问。使用浏览器打开这个地址，运行结果如图 1-15 所示。

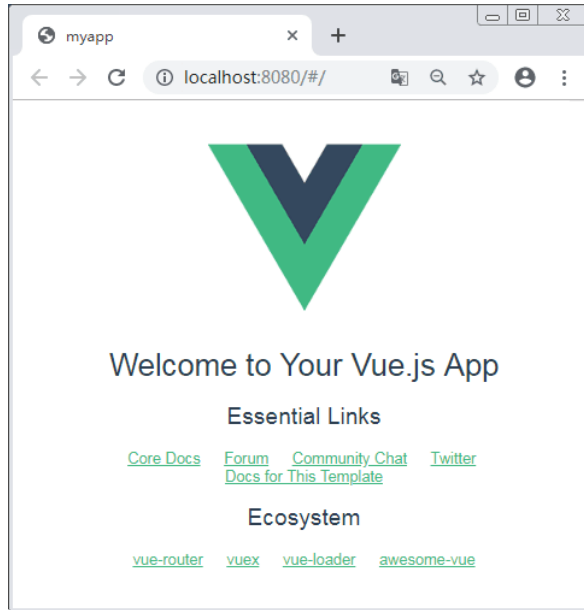


图1-15 运行结果

(5) 使用 VS Code 编辑器打开 “C:\chapter01\myapp” 目录，就可以在该目录下进行项目的开发了。

## 本章小结

本章主要讲解了什么是 Vue、Vue 的特点和发展前景、Vue 开发环境的搭建，以及 webpack 打包工具的使用。通过本章的学习，读者应对 Vue 有一个整体的认识，能够编写一个简单的 Hello World 程序。

## 课后习题

### 一、填空题

1. Vue 是一套构建\_\_\_\_\_的渐进式框架。
2. Vue 设计思想 MVVM 主要表示数据、视图和\_\_\_\_\_。
3. Vue 中通过\_\_\_\_\_属性获取相应 DOM 元素。
4. 在进行 Vue 调试时，通常使用\_\_\_\_\_工具来完成项目开发。
5. Vue 中页面结构以\_\_\_\_\_形式存在。

### 二、判断题

1. Vue 与 Angular 和 React 框架不同的是，Vue 设计为自下而上逐层应用。( )

2. Vue 完全能够为复杂的单页应用提供驱动。( )
3. Vue 是一套构建用户界面的渐进式框架，Vue 的核心只关注视图层。( )
4. Vue 中 MVVM 框架主要包含 3 部分组成：Model、View 和 ViewModle。( )
5. Vue 可以在 Node 环境下进行开发，并借助 npm 包管理器来安装依赖。( )

### 三、选择题

1. 下列关于 Vue 说法错误的是 ( )。  
A. Vue 与 Angular 都可以用来创建复杂的前端项目  
B. Vue 优势主要包括轻量级、双向数据绑定  
C. Vue 在进行实例化之前，应确保已经引入了核心文件 vue.js  
D. Vue 与 React 语法是完全相同的
2. 下列关于 Vue 的优势的说法错误的是 ( )。  
A. 双向数据绑定  
B. 轻量级框架  
C. 增加代码的耦合度  
D. 实现组件化
3. 下列不属于 Vue 开发所需工具的是 ( )。  
A. Chrome 浏览器  
B. VS Code 编辑器  
C. vue-devtools  
D. 微信开发者工具
4. npm 包管理器是基于 ( ) 平台使用的。  
A. Node.js  
B. Vue  
C. Babel  
D. Angular
5. 下列选项中，用来安装 vue 模块的正确命令是 ( )。  
A. npm install vue  
B. node.js install vue  
C. node install vue  
D. npm I vue

### 四、简答题

1. 请简述什么是 Vue。
2. 请简述 Vue 优势有那些。
3. 请简单介绍 Vue、React 之间的区别。

### 五、编程题

1. 请使用 Vue.js 动手创建一个登录页面。



2. 请手动配置 Vue.js 开发环境。