

```

#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
#mirror_ob.select = 0
#one = bpy.context.selected_objects[0]
#bpy.data.objects[one.name].select = 1
cept:
    print("please select exactly two objects, the last one gets the modifier unless 1

----- OPERATOR CLASSES -----
ool

```

```

orX(bpy.types.Operator):
s adds an X mirror to the selected object""
ame = "object.mirror_mirror_x"
el = "Mirror X"

method
1(context):
    bpy.context.scene.objects.active = one

```

第 1 章

理解 Vue.js 及其产生的背景

Vue.js 是一套响应式系统、前端开发库。Vue 自问世以来，所受关注度不断提高，在当前的市场上，Vue 是非常流行的 JavaScript 技术开发框架之一。本章将对什么是 Vue.js、Vue.js 产生的背景及如何学习 Vue.js 等内容进行介绍。

1.1 什么是 Vue.js

在介绍 Vue 之前，我们先来简单介绍一下它的作者——尤雨溪(Evan You)，以及它的由来。尤雨溪是一位美籍华人，在上海复旦大学附中读完高中后，在美国完成大学学业，本科毕业于科尔盖特大学(Colgate University)，后在帕森斯设计学院(Parsons School of Design)获得设计与科技(Design & Technology)艺术硕士学位。他是 Vue Technology LLC 创始人，曾经在谷歌创意实验室(Google Creative Lab)就职，参与过多个项目的界面原型研发，后加入 Meteor，参与 Meteor 框架本身的维护和 Meteor Galaxy 平台的交互设计与前端开发。

2014 年 2 月，尤雨溪开源了一个前端开发库 Vue.js。Vue.js 是构建 Web 界面的 JavaScript 库，也是一个通过简洁的 API 提供高效数据绑定和灵活组件的系统。2016 年 9 月 3 日，在南京的 JSConf 上，尤雨溪正式宣布以技术顾问的身份加盟阿里巴巴 Weex 团队，来做 Vue 和 Weex 的 JavaScript runtime 整合，目标是让大家能用 Vue 的语法跨三端。目前，他全职投入 Vue.js 的开发与维护，立志将 Vue.js 打造成与 Angular/React 平起平坐的世界顶级框架。

我们也可以说 Vue.js 是一套响应式系统(reactivity system)。数据模型层(model)只是普通的 JavaScript 对象，如图 1-1 所示，“data”代表一个 JavaScript 对象，修改它则更新相应的 HTML 片段(DOM)，这些 HTML 片段也称为“视图”(view)。这会让状态管理变得非常简单且直观，可实现数据的双向绑定，因此，我们也称其为响应式系统。



图 1-1 Vue.js 双向绑定

1.1.1 为什么使用 Vue.js

我们都知道完整的网页是由 DOM 组合与嵌套形成最基本的视图结构，再加上 CSS 样式的修饰，使用 JavaScript 接受用户的交互请求，并通过事件机制来响应用户交互操作而形成的。最基本的视图结构即视图层，这个被称为视图层的部分就是 Vue.js 核心库关注的部分。为什么关注它呢？因为一些页面元素非常多，结构庞大的网页如果使用传统开发方式，

数据和视图会全部混合在 HTML 中，处理起来十分不易，而且结构之间还存在依赖或依存关系，代码就会出现更多问题。有前端开发基础的读者都基本了解 jQuery，jQuery 提供了简洁的语法和跨平台的兼容性，极大地简化了 JavaScript 开发人员遍历 HTML 文档、解析 DOM、进行事件处理等的操作。

用过 jQuery 的读者都有体会，开始时页面元素不多，有时需要一层层地不断向上寻找父辈元素，如 `$('#xxx').parent().parent()`，但后期修改页面结构时，代码可能就会变得臃肿，如 `$('#xxx').parent().parent().parent()`，随着产品升级的速度越来越快，修改的工作量变得越来越大，页面中相似的关联和嵌套 DOM 元素多得数不清，而 jQuery 选择器及 DOM 操作本身也存在性能缺失问题，想要修改无从下手。原本轻巧简洁的 jQuery 代码，在产品需求面前变得特别冗长。

但是 Vue.js 解决了这些问题，这些问题将在 Vue.js 中消失。

1.1.2 Vue.js 的主要特点

Vue.js 是一个优秀的前端界面开发 JavaScript 库，之所以非常火，是因为它有众多突出的特点，其中主要的特点有以下几个。

1. 轻量级的框架

Vue.js 能够自动追踪依赖的模板表达式和计算属性，并提供 MVVM 数据绑定和一个可组合的组件系统，具有简单、灵活的 API，使读者更加容易理解。

2. 双向数据绑定

声明式渲染是数据双向绑定的主要体现，同样也是 Vue.js 的核心，它允许采用简洁的模板语法将数据以声明式渲染整合进 DOM。

3. 指令

Vue.js 与页面进行交互，主要通过内置指令来完成。指令的作用是当表达式的值改变时，相应地将某些行为应用到 DOM 上。

4. 组件化

组件(component)是 Vue.js 最强大的功能之一。组件可以扩展 HTML 元素，封装可重用的代码。在 Vue.js 中，父子组件通过 props 通信，从父组件向子组件单向传递。子组件与父组件通信，通过触发事件通知父组件改变数据。这样就形成了一个基本的父子通信模式。

在开发中，当组件和 HTML、JavaScript 等有非常紧密的关系时，可以根据实际的需要自定义组件，使开发变得更加便利，并大量减少代码编写量。组件还支持热重载(hot-reload)。当我们做了修改时，不会刷新页面，只是对组件本身进行立刻重载，不会影响整个应用当前的状态。CSS 也支持热重载。

5. 客户端路由

Vue-router 是 Vue.js 官方的路由插件，与 Vue.js 深度集成，用于构建单页面应用。Vue.js 单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来，传统的页面通过超链接实现页面的切换和跳转。

6. 状态管理

状态管理实际就是一个单向的数据流，状态驱动视图的渲染，而用户对视图进行操作产生动作，使状态产生变化，从而使视图重新渲染，形成一个单独的组件。

最新发布的 Vue 3.3，针对开发者体验进行了改进，特别是在使用 TypeScript 时的 SFC `<script setup>`，同时与 Vue 语言工具(以前称为 Volar)的发布相结合。

1.1.3 Vue.js 的优势

Vue.js 与其他框架相比有什么优势呢？上面我们已经提到了 jQuery，还有其他的前端框架，如 React、Angular 等。相比较而言，Vue.js 最为轻量化，而且已经形成了一套完整的生态系统，可以快速迭代更新，是前端开发人员的首选入门框架，Vue.js 有很多优势。

- (1) Vue.js 可以进行组件化开发，使代码编写量大大减少，读者更加易于理解。
- (2) Vue.js 最突出的优势在于，可以对数据进行双向绑定(在后面的编写中我们会明显地感觉到这个特点的便捷性)。
- (3) 使用 Vue.js 编写的界面效果本身就是响应式的，这使网页在各种设备上都能显示出非常好看的效果。
- (4) 相比传统的页面通过超链接实现页面的切换和跳转，Vue.js 使用路由不会刷新页面。

1.2 Vue.js 产生的背景

Vue.js 是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue.js 采用自底向上增量开发的设计。Vue.js 的核心库只关注视图层，并且非常容易学习，方便与其他库

或已有项目整合。另外，Vue.js 完全有能力驱动采用单文件组件和 Vue.js 生态系统支持的库开发的复杂单页应用。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。

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十年前写个 HTML CSS 那样简单了，新的概念层出不穷，比如 ECMAScript Node.js NPM、前端工程化等。这些新概念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”的技术已被许多商业项目用于生产环境：
jQuery + RequireJS (SeaJS) + artTemplate (doT) + Gulp (Grunt)

这套技术以 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，因为它们的客户很可能还在用 IE II 及以下浏览器。使用 RequireJS(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，由于它具有简洁、高效、实用等特点，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满足我们的需求，这时就出现了如 Angular、React 及我们要介绍的主角 Vue.js。

1.2.2 Vue.js 的开发模式

根据项目需求，可以选择从不同的维度来使用 JavaScript 框架。如果只是想体验 Vue.js 带来的快感，或者开发几个简单的 HTML 页面或小应用，可以直接通过 Script 代码加载 CDN 文件，例如：

```
<!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>vue 示例</title>
  <script src="../../js/vue.global.min.js"></script>
</head>
<body>

  <div id="vueImpl">
    <ul>
      <li v-for="page in book">{{ page.name }}</li>
    </ul>
  </div>
  <script>
    const app = Vue.createApp({
      data() {
        return {
          book: [{
            "name": "理解 Vue.js 及产生的背景"
          }, {
            "name": "快速开启一个 Vue.js 应用"
          }]
        }
      }
    })
    app.mount('#vueImpl')
  </script>
</body>
</html>
```

在浏览器中访问上述代码，可以将本书的章节列表循环显示出来，如图 1-2 所示。

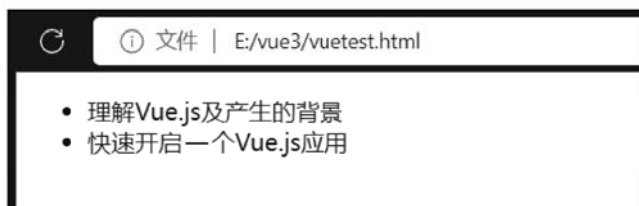


图 1-2 Vue.js 示例在浏览器中访问的效果

对于一些业务逻辑复杂并对前端工程有要求的项目，可以结合 Webpack 使用 Vue 单文件的形式，并且还会用到 Vue-router 来管理路由。这里提到的很多内容目前读者还不必过多了解，只是说明 Vue 框架的开发模式多样化，后续章节会详细介绍，到时读者就会对整个 Vue 生态有更详细的了解了。

1.2.3 Vue.js 与 jQuery 的区别

jQuery 是一个跨平台的 JavaScript 库，旨在简化 HTML 的客户端脚本。jQuery 也是开源的，许多统计表明，它是最常用和广泛部署的 JavaScript 库。

使用 jQuery 可轻松导航文档，例如选择 DOM 元素，创建动画，处理事件和开发 Ajax 应用程序。它还可以用于在 JavaScript 库之上创建插件。

下面我们通过比较 jQuery 和 Vue，来看看二者之间有什么区别。

1. 两者未来发展的趋势

jQuery 是一个类库，提供了很多方法，不能算框架。一直以来，jQuery 是最流行的 Web 前端 JS 库，可是现在，无论是在国内还是国外，它正在渐渐被其他 JS 库所替代。随着浏览器厂商统一遵循 HTML 5 规范以及 ECMA 6 在浏览器端的实现，jQuery 的使用率会越来越低。

Vue 是一个刚兴起的前端框架，有一套完整的体系，是一个精简的 MVVM(Model-View-View-Model)。从技术角度讲，Vue.js 专注于 MVVM 模型的 ViewModel 层，通过双向数据绑定把 View 层和 Model 层连接起来，通过对数据的操作就可以完成对页面视图的渲染。Vue 以其独特的优势(即简单、快速、组合、紧凑、强大)而迅速崛起。

2. 在操作思想上有本质的区别

Vue.js 通过 Vue 对象将数据和 View 完全分离开，对数据操作不再引用相应的 DOM 对象，主要是操作数据。

jQuery 是使用选择器(\$)选取 DOM 对象，并对其进行赋值、取值、事件绑定等操作；主要是操作 DOM。和原生的 JS 区别只在于可以更方便地选取和操作 DOM 对象；数据和界面是在一起的，比如获取 input 标签的内容：

```
$('#input').val();
```

还是依赖 DOM 元素的值。

Vue 基于一种 MVVM 模式，使用数据驱动的方式，通过 Vue 对象将数据和相应的 DOM 对象绑定。

3. 侧重点不同

Vue 侧重数据绑定，可以应用于复杂数据操作的后台页面，如表单填写页面。

jQuery 侧重样式操作、制作动画效果等，可以应用于一些 HTML 5 动画页面，及一些需要 JS 来操作页面样式的页面中。

1.2.4 Vue.js 与 React、Angular 的比较

1. 架构设计不同

❖ Angular

Angular 框架属于 MEAN 框架，是如今创业公司最热门的技术栈。Angular 是一个完整的基于 TypeScript 的 Web 应用开发框架，主要用于构建单页 Web 应用(SPA)。

与 Angular JS 这一早期的框架不同，Angular 2 是基于组件的，与 MV*模式没有什么关联。Angular 的结构包括模块、组件和服务。

在 Angular 框架中，每个组件都有一个类或模板，定义了应用逻辑和元数据(MetaData)。组件的这些元数据为创建和呈现其视图所需的构件提供了指引。

Angular 架构的另一个重要特点是，模板是用 HTML 编写的。这些用 HTML 编写的模板还可以包含 Angular 模板语法，并带有特殊指令，以输出响应式数据，并且可以渲染多个元素。

服务是 Angular 应用中的一个独特元素，被组件用于委托业务逻辑任务，如获取数据或验证输入。

❖ React

React 是一个开源的前端库，主要用于开发用户界面。这种灵活的前端解决方案并不强制执行特定的项目结构。React 开发者可能只需要几行代码就可以开始使用它。

React 是基于 JavaScript 的，但在大多数情况下，它与 JSX(JavaScript XML)结合使用。JSX 是一种语法扩展，允许开发人员同时创建包含 HTML 和 JavaScript 的元素。实际上，开发者用 JSX 创建的任何东西都可以用 React JavaScript API 创建。React 元素比 DOM 元素更强大，它们是 React 应用的最小组成部分，即组件。

React 组件是一种构建模块，它决定了在整个 Web 应用中使用哪些独立和可重用的组件。

❖ Vue

Vue 用于开发用户界面和单页 Web 应用，它是一个开源的 MVVM 前端 JavaScript 库，被称为渐进式框架，与其他工具一起用于前端开发。Vue 的多用途、高性能和它在 Web 应用程序上的最佳用户体验，使其得以流行。

使用 Vue 时，开发者主要在 ViewModel 层上工作，以确保应用数据的处理方式能让

框架呈现最新的视图。

Vue 的模板语法将可识别的 HTML 与特殊的指令和功能相结合，允许开发人员创建 View 组件。

现在 Vue 中的组件是小巧、自成一体和可复用的。单文件组件(SFC)使用扩展名 .vue，包含 HTML、JavaScript 和 CSS，所有相关代码都存放在同一个文件中。

在大型的 Vue.js 项目中，我们通常推荐使用 SFC 来组织代码。要将 SFC 移植到工作的 JavaScript 代码中，需要 Webpack 或 Browserify 这样的构建工具。

2. 适用目标和范围

❖ Angular

Angular 比较适合大型和高级项目，可能包括但不限于：

- 开发渐进式 Web 应用程序(PWA)。
- 重新设计网站应用程序。
- 建立基于内容的动态网页。
- 创建有着复杂基础架构的大型企业应用程序。

❖ React

React 是一种以构建复杂的业务应用程序而闻名的技术架构。当将它与 Redux、MobX 或其他 Flux 模式的状态管理库一起使用时，React 就能够成为强大的工具。React 比较适合以下项目：

- 涉及包含导航项、折叠或展开的手风琴分节、可用或不可用状态、动态输入、可用或不可用按钮、用户登录、用户访问权限等许多组件的应用程序。
- 具有扩展和增长可能的项目，因为 React 组件具有声明性，所以它可以轻松处理此类复杂结构。
- 当 UI 是网络应用程序中心的项目。

❖ Vue

因为 Vue 简单易学，所以比较适合解决短期的、小型的问题。它可以轻松地与现有代码块集成。以下情况可能需要 Vue：

- 带有动画或交互式元素的 Web 应用程序的开发项目。
- 无须高级技能即可进行原型制作。
- 需要与多个其他应用程序无缝集成的应用程序。
- 更早推出 MVP。

3. 性能和开发

❖ Angular

Angular 在性能方面的一些亮点包括：

- 有无缝的第三方集成，可以增强产品或应用程序的功能。
- 可提供强大的组件集合，从而简化了编写、更改和使用代码的过程。
- Angular 的“提前编译器”赋予了应用程序更快的加载时间和更高的安全性。
- MVC 模型通过允许视图分离来帮助减少后台查询。
- 促进组件解耦依赖性引入的外部元素，从而为可复用性、简化管理及测试铺平道路。
- 通过将任务分成逻辑块来减少网页的初始加载时间。
- 可以完全自定义地设计。
- 便于将 HTML 和 TypeScript 编译为 JavaScript —— 这大大加快了代码的编译速度，并在浏览器开始加载 Web 应用程序之前就开始编译。

❖ React

因为两者具有相同的架构，所以在性能方面 React 与 Vue 不相上下。React 开发 Web 的优势如下：

- 支持打包和树摇优化(tree-shaking) —— 这对于减少最终用户的资源负载至关重要。
- 由于提供了单向数据绑定支持，所以可以更好地控制项目。
- 便于进行测试和监控管理。
- 最适合需要频繁更改的复杂应用程序。

❖ Vue

最贴切的形容 Vue 的词组是“令人难以置信的快速”。它的一些性能优势是：

- 更快的学习曲线。
- 单页应用程序高效、精密。
- 高级功能使它具有多功能性。

4. 各自的优点

❖ Angular

- 有模板、表单、引导程序或架构、组件以及组件之间交互的完整的文档。
- 平滑的双向数据绑定。
- MVC 架构。

- 内置模块系统。
- 大大减少了网页的初始加载时间。
- 使用 Angular 构建的流行应用程序有 Youtube TV、PayPal、Gmail、Forbes、Google Cloud。

❖ React

- 通过模块化的结构使其拥有灵活的代码，节省了时间和成本。
- 助力复杂应用程序高性能的实现。
- 使用 React 前端开发能够更容易地进行代码维护。
- 支持适用于 Android 和 iOS 平台的移动端原生应用程序。
- 使用 React 构建的流行应用程序有 Tesla、AirBnB、CNN、Nike、Udemy、Linked-in。

❖ Vue

- 体积小巧，便于安装和下载。
- 倘若正确利用，就可以在多处重用 Vue。
- Vue.js 允许更新网页中的元素，而无须渲染整个 DOM，因为它是虚拟的 DOM。
- 需要较少的优化。
- 加速 Web 应用程序的开发，并允许将模板虚拟到 DOM，与编译器分开。
- 拥有经过验证的兼容性和灵活性。
- 不管应用程序的规模如何，代码库都不会变。

1.2.5 如何选择 Angular、React 和 Vue.js

❖ Angular

Angular 是成熟的框架之一，拥有优秀的贡献者和确保应用程序开发的完整包。但是，它需要深入学习并创建观察者来查看更新，这可能会让新的应用程序开发人员望而却步。总而言之，对于需要开发大规模应用程序的公司来说，Angular 是一个理想的选择。

❖ React

React 现在已经有多年的历史了，它已获得全球认可，是前端开发的不错选择。它非常适合希望创建单页面应用程序(Single Page Application, SPA)的公司。

❖ Vue

Vue 是一个年轻的库，虽然没有任何大公司的支持，但仍被认为是 Angular 和 React 的强大竞争对手。由于其具有灵活性和易用性，Vue 已成为众多行业巨头的选择。

1.3 如何学习 Vue.js

Vue.js 是一套用于构建用户界面的渐进式框架，主要用于快速地构建前端界面，与其他大型的前端框架不同，Vue.js 可以自底向上逐层应用。

相比 Angular.js，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，是初创项目的前端首选框架。当与现代化的工具链以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支持。

Vue.js 是一个用于构建用户界面的前端库，本身就具有响应式编程和组件化的诸多优点。所谓响应式编程，就是一种面向数据流和变化传播的编程范式，可以在编程时很方便地表达静态或动态的数据流，而相关的计算模型会自动将变化的值通过数据流进行传播。

响应式编程在前端开发中得到了大量的应用，在大多数前端 MVX 框架中都可以看到它的影子。相较于 Angular.js 和 React.js，Vue.js 并没有引入太多的新概念，只是对已有的概念进行了精简。并且 Vue.js 很好地借鉴了 React.js 的组件化思想，应用开发起来更加容易，真正实现了模块化开发的目的。

Vue.js 一直以轻量级、易上手而被人们称道。MVVM 的开发模式也使前端开发从传统的 DOM 操作中释放出来，开发者不需要再把时间浪费在视图和数据的维护上，只需要关注数据的变化即可。并且 Vue 的渲染层基于轻量级的 Virtual-DOM 实现，在大多数的场景下，初始化速度和内存效率都提高了 2~4 倍。目前，越来越多的移动客户端也开始支持使用 Vue.js 进行开发，可以坚信，未来使用 Vue.js 打造三端一致的 Native 应用将变成可能。

作为一个新兴的前端框架，Vue.js 大量借鉴和参考了 Angular.js 和 React.js 等优秀的前端框架。而在版本支持上，Vue.js 放弃了对 IE 8 的支持，对移动端的支持也有一定的要求，也就是说，使用 Vue.js 进行移动跨平台开发时，需要 Android 4.2+ 和 iOS 7+ 支持。

1.3.1 知识储备

1. JavaScript 与 Web 基础

Vue.js 作为一个用于构建 Web 用户界面的 JavaScript 框架，在开始使用之前，必须了解 JavaScript 和 Web 开发的基础知识，并且还需要掌握 Vue.js 生态系统的一些核心知识，包括 Vue.js 核心库、Vue Router 和 Vuex。

2. Vue.js 核心功能

从根本上说, Vue.js 就是一个用于同步网页的 JavaScript 技术框架, 其关键特性是反应式(reactive)数据, 以及指令和插值等模板功能。

要构建一个 Vue.js 应用程序, 还需要知道如何在网页中安装 Vue.js, 并了解 Vue.js 实例的生命周期等知识。

3. 组件

Vue.js 组件是独立的可重用 UI 元素。因此, 需要了解如何声明组件, 以及如何通过 prop 和 event 在组件之间发生交互。

了解如何组合组件也很重要, 因为这对使用 Vue.js 构建健壮、可伸缩的应用程序来说至关重要。

4. 单页面应用程序

单页面应用程序(SPA)架构通过单个网页实现传统多页面网站一样的功能, 而且不会在每次用户触发导航时重新加载和重建页面。

Vue Router 是一个用于构建 SPA 的工具, 由 Vue.js 团队维护, 在将“页面”构建为 Vue.js 组件之后, 就可以使用 Vue Router 将每个“页面”映射到一个唯一的路径。

5. 状态管理

随着应用程序变得越来越大, 项目变得越来越复杂, SPA 页面中会有很多组件, 管理全局状态变得异常困难, 而且随着 prop 和 event 监听器的增加, 组件变得越来越臃肿。

由 Vue.js 团队维护的 Vuex 库可以帮助用户在 Vue.js 应用程序中实现 Flux。通过 Flux 模式, 可以将数据保存在可预测且稳定的中央存储中。

6. 软件选择

1) VS Code

VS Code(Visual Studio Code)(见图 1-3)是微软公司推出的一款轻量级代码编辑器, 免费而且功能强大, 对 JavaScript 和 Node.js 的支持非常好, 自带很多功能, 例如代码格式化、代码智能提示补全、Emmet 插件等。VS Code 推荐以文件夹的方式打开项目。

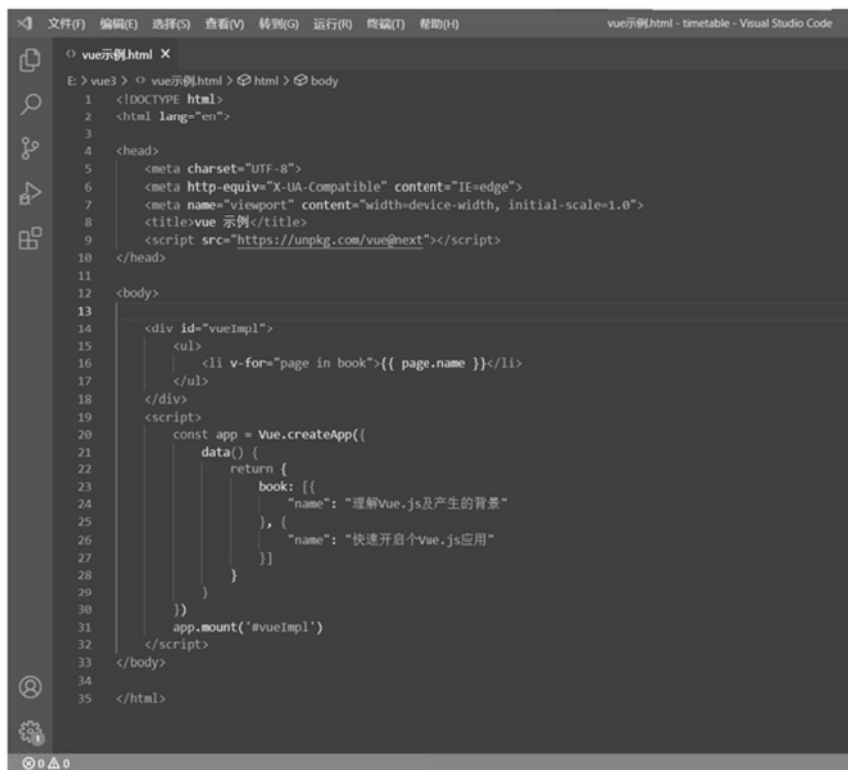


图 1-3 VS Code 软件界面

注意：想开发 Vue 还需要下载插件。

2) WebStorm

WebStorm(见图 1-4)是 JetBrains 公司旗下的一款 JavaScript 开发工具,已经被中国广大 JS 开发者誉为“Web 前端开发神器”“最强大的 HTML 5 编辑器”“最智能的 JavaScript IDE”等。WebStorm 与 IntelliJ IDEA 同源,继承了 IntelliJ IDEA 强大的 JS 部分的功能。

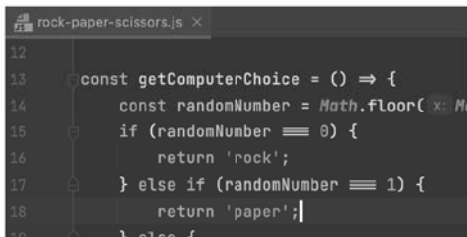


图 1-4 WebStorm 软件界面

3) Sublime Text

Sublime Text(见图 1-5)是一个文本编辑器(收费软件, 可以无限期试用, 但是有激活提示弹窗), 同时也是一款先进的代码编辑器。Sublime Text 由程序员 Jon Skinner 于 2008 年 1 月开发, 最初被设计为一个具有丰富扩展功能的文本编辑器。

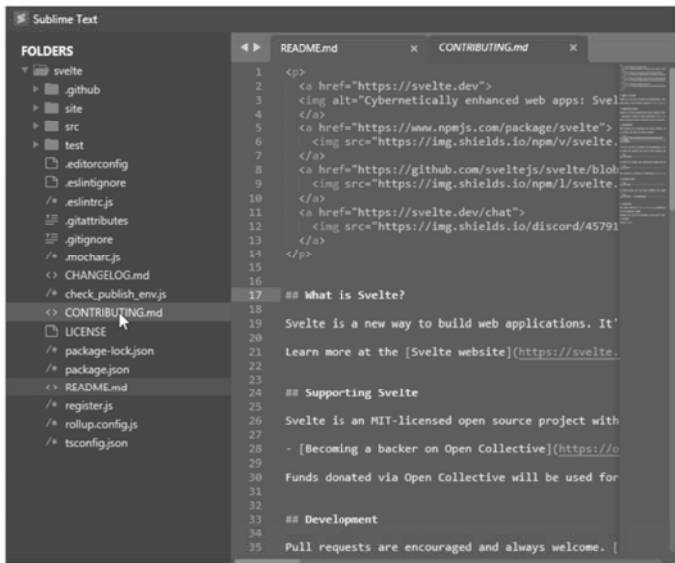


图 1-5 Sublime Text 软件界面

Sublime Text 具有漂亮的用户界面和强大的功能, 例如代码缩略图、Python 的插件、代码段等, 还可自定义键绑定、菜单和工具栏。Sublime Text 的主要功能包括拼写检查、书签、完整的 Python API、Goto、即时项目切换、多选择、多窗口等。Sublime Text 是一款跨平台的编辑器, 可以同时支持 Windows、Linux、Mac OS X 等操作系统。

Sublime Text 支持多种编程语言的语法高亮显示, 拥有优秀的代码自动完成功能, 还拥有代码片段(Snippet)的功能, 可以将常用的代码片段保存起来, 在需要时随时调用; 支持 Vim 模式, 可以使用 Vim 模式下的多数命令; 支持宏, 简单地说, 宏就是把操作录制下来或者自己编写命令, 然后播放刚才录制的操作或者命令。

4) HBuilder

HBuilder(见图 1-6)是 DCloud(数字天堂)公司推出的一款支持 HTML 5 的 Web 开发 IDE。HBuilder 的编写用到了 Java、C、Web 和 Ruby 语言。HBuilder 本身主体是用 Java 编写的。它基于 Eclipse, 因此, 顺其自然地兼容了 Eclipse 的插件。

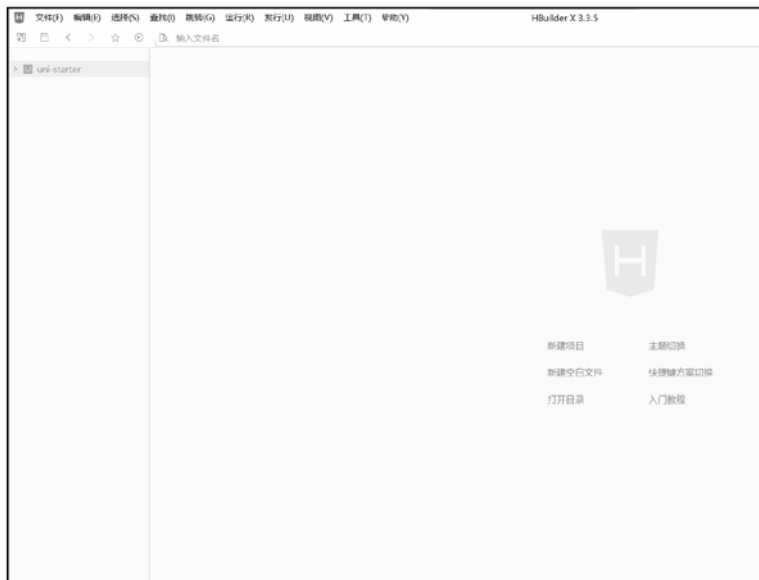


图 1-6 HBuilder 软件界面

5) Atom

Atom(见图 1-7)是 GitHub 专门为程序员推出的一款跨平台文本编辑器，具有简洁和直观的图形用户界面，并有很多有趣的特点，支持 CSS、HTML、JavaScript 等网页编程语言。Atom 支持宏，可自动完成分屏功能，集成了文件管理器。

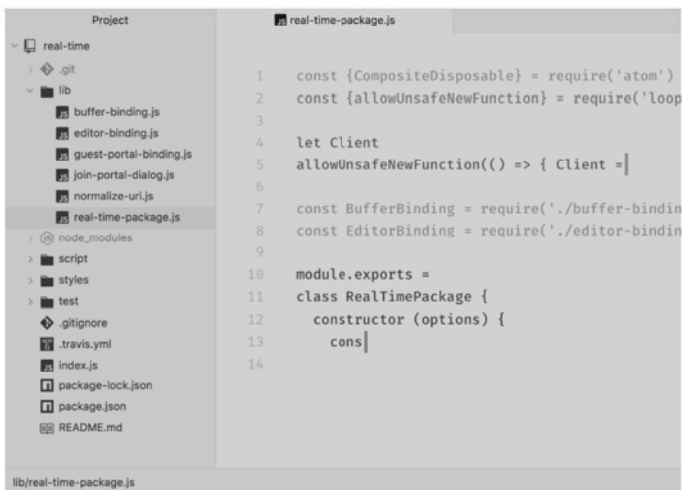


图 1-7 Atom 软件界面

1.3.2 学习安排

1. 起步

(1) 扎实的 JavaScript / HTML / CSS 基本功，是学习 Vue.js 必要的条件。

(2) 通读官方教程(guide)的基础篇。不要用任何构建工具，就只用最简单的<script>，把教程里的例子模仿一遍，理解用法。不推荐上来就直接用 vue-cli 构建项目，尤其是没有 Node/Webpack 基础的话。

(3) 仿照官网上的示例，模仿着实现来练手，加深理解。

(4) 阅读官方教程进阶篇的前半部分，到“自定义指令”(Custom Directive)为止。着重理解 Vue.js 的响应式机制和组件生命周期。如果对“渲染函数”(Render Function)理解吃力，可以先跳过。

(5) 阅读教程里关于路由和状态管理的章节，然后根据需要学习 Vue-router 和 Vuex。同样，先不要管构建工具，以通过文档里的例子理解用法为主。

(6) 阅读完基础文档后，如果对于基于 Node 的前端工程化不熟悉，就需要补课了。

2. 前端生态/工程化

(1) 了解 JavaScript 背后的规范、ECMAScript 的历史和目前的规范制定方式。学习 ES 2015/16 的新特性，理解 ES 2015 Modules，适当关注还未成为标准的提案。

(2) 学习命令行的使用，建议用 Mac。

(3) 学习 Node.js 基础。建议使用 nvm 这样的工具来管理机器上的 Node 版本，并且将 npm 的 registry 注册表配置为淘宝的镜像源。至少要了解 npm 的常用命令、npm scripts 如何使用、语义化版本号规则、CommonJS 模块规范(了解它和 ES 2015 Modules 的异同)、Node 包的解析规则，以及 Node 的常用 API。应当做到可以自己写一些基本的命令行程程序。注意，最新版本的 Node(6+)已经支持 ES 2015 的绝大部分特性，可以借此学习巩固 ES 2015。

(4) 为了兼容浏览器环境，需了解如何配置 Babel，以及将 ES 2015 编译到 ES 5。

(5) 学习 Webpack。Webpack 是一款极其强大同时也很复杂的工具，作为起步，理解它的“一切皆模块”的思想，并基本了解其常用配置选项和 loader 的概念及使用方法。比如，如何搭配 Webpack 以使用 Babel。学习 Webpack 是一个挑战，原因在于其本身文档比较混乱，建议多搜索，应该有质量不错的第三方教程。英文好的建议阅读 Webpack 2.0 的文档，比起 1.0 已有极大的改善，但需要注意其与 1.0 的不同之处。

3. Vue 进阶

(1) 有了 Node 和 Webpack 的基础,就可以通过 vue-cli 来搭建基于 Webpack 并且支持单文件组件的项目了。建议从 webpack-simple 这个模板开始,并阅读官方教程进阶篇剩余的内容以及 vue-loader 的文档,了解一些进阶配置。有兴趣的读者可以自己动手从零开始搭建一个项目以加深理解。

(2) 根据例子,可尝试在 Webpack 模板的基础上整合 Vue-router 和 Vuex。

(3) 深入理解 Virtual DOM 和“渲染函数”(Render Functions)(可选择性地使用 JSX),理解模板和渲染函数之间的对应关系,了解其使用方法和适用场景。

(4) 根据需求,了解服务端渲染的使用(需要配合 Node 服务器开发的知识)。其实,更重要的是理解它所解决的问题,并搞清楚自己是否需要它。

(5) 阅读开源的 Vue 应用、组件、插件源码,自己尝试编写 Vue 组件、插件。

(6) 参考指南阅读 Vue 源码,理解内部实现细节。

1.4 小结

在本章中,我们对 Vue.js 的产生背景和优势做了一个比较详细的讲解,也对 Vue.js 和传统的开发模式做了区分和比较,了解它们的不同之处也是比较重要的,这会让我们加深对 Vue.js 的了解。软件选择是程序员编写代码很重要的一部分,在日常开发中都会用到,本章推荐了 5 种不同的 IDE,后续将会使用 VS Code 来讲解。本章对学习路径从起步到进阶都做了介绍,将这些知识点融会贯通,可方便后续学习,开发出真正可以使用的网页,这也是学习的目的。

```

#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
#mirror_ob.select = 0
#one = bpy.context.selected_objects[0]
#bpy.data.objects[one.name].select = 1
cept:
print("please select exactly two objects, the last one gets the modifier unless 1

```

```

----- OPERATOR CLASSES -----
ool

```

```

orX(bpy.types.Operator):
s adds an X mirror to the selected object""
ame = "object.mirror_mirror_x"
el = "Mirror X"

```

```

method
11(ctx, context):
turn context.active_object into a

```

第 2 章

快速开启 一个 Vue.js 应用

简单来说，Vue.js 是一个应用框架，像 Java 中的 Spring 系列一样，能给我们提供简洁高效的语法，能够高效且优质地协助我们完成一个前端项目；同时，由于它是一个轻量级框架，我们只需要引入一个 vue.js 就可以使用它的语法了。

2.1 开发环境准备

开发环境包括硬件环境和软件环境。本节将带大家安装事件驱动 I/O 服务端 JavaScript 环境 Node.js、管理(下载/卸载/发布)第三方模块的工具 NPM、集成开发环境 VS Code、Vue CLI 和检查及调试 Vue.js 的应用工具 Vue Devtools。

2.1.1 安装 Node.js

1. Node.js 介绍

Node 是一个能让 JavaScript 运行在服务端的开发平台，使 JavaScript 成为与 PHP、Python、Perl、Ruby 等服务端语言平起平坐的脚本语言。Node 发布于 2009 年 5 月，由 Ryan Dahl 开发。

简单地说，Node.js 就是运行在服务端的 JavaScript。Node.js 是一个基于 Chrome JavaScript 运行时建立的平台，底层架构是 JavaScript，文件后缀是.js。

Node.js 是一个事件驱动 I/O 服务端 JavaScript 环境，基于 Google 的 V8 引擎，V8 引擎执行 JavaScript 的速度非常快，性能非常好。


2. 下载安装 Node.js

(1) 首先，输入下载地址 <http://nodejs.cn/download/>，打开如图 2-1 所示的下载界面。



图 2-1 下载界面

- (2) 选择 64 位的 Windows 安装包(.msi)，下载完成后默认安装即可。
- (3) 安装完成后，在 cmd 窗口中输入 `node -v`，查看是否安装成功。若看到如图 2-2 所示的输出，则证明安装成功。



```
命令提示符
Microsoft Windows [版本 10.0.19044.1766]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\刘彦祖>node -v
v16.15.1

C:\Users\刘彦祖>
```

图 2-2 检验 Node.js 是否安装成功

2.1.2 设置 NPM 镜像

1. NPM 介绍

- (1) NPM 是管理(下载、卸载、发布)第三方模块的工具。
- (2) NPM 是 Node 包管理器，包含管理 Node 包的工具。
- (3) NPM 这个工具在安装 Node 的时候就已经同时安装了。

2. NPM 的作用

- (1) 可以管理 Node 模块。
- (2) 可下载并安装第三方模块。
- (3) 可卸载第三方模块。
- (4) 可发布模块。
- (5) 可删除已发布的模块。

3. NPM 镜像安装

- (1) 通过命令配置淘宝镜像，切换为淘宝镜像：

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

查看当前使用的镜像地址：

```
npm config get registry
```

如果返回 <https://registry.npm.taobao.org>，则说明镜像配置成功。

(2) 切换回原镜像：

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

(3) 其他镜像地址查询。

安装 NRM：

```
npm install nrm -g
```

查看安装的 NRM 版本：

```
nrm -V
```

或者

```
nrm --version
```

成功时显示如下：

```
npm ---- https://registry.npmjs.org/
cnpm --- http://r.cnpmjs.org/
taobao - https://registry.npm.taobao.org/
nj ----- https://registry.nodejitsu.com/
npmMirror https://skimdb.npmjs.com/registry/
edunpm - http://registry.enpmjs.org/
```

(4) 通过配置淘宝镜像安装并使用 **cnpm**。

安装 **cnpm**：

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

使用 **cnpm**：

```
cnpm install xxx
```

2.1.3 选择合适的 IDE

IDE 的全称为 **integrated development environment**(集成开发环境)，是由一系列与软件开发有关的工具整合而成的统一软件。如果要做一个比喻，可以把 IDE 看成编程人员手中的一套“武器”。在正式编程之前，了解不同的 IDE 的优缺点，选择一个最为称手的 IDE，对我们后续的编程开发会有很大的帮助。

第 1 章已经介绍了 5 种前端开发软件，本书选用 VS Code 来讲解。

1. 下载安装 VS Code

(1) 首先，在浏览器中输入下载地址 <https://code.visualstudio.com/>，即可打开如图 2-3 所示的下载界面。

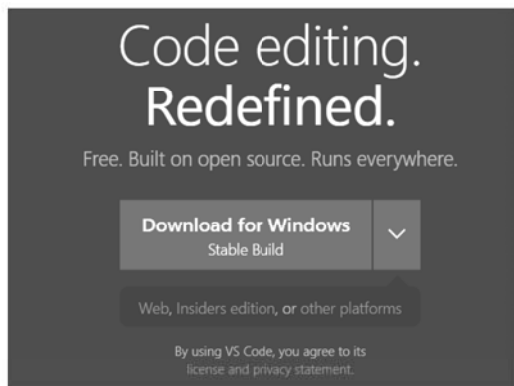


图 2-3 VS Code 下载界面

(2) 下载完成后，双击安装 VS Code，进入如图 2-4 所示的界面。



图 2-4 安装 VS Code

(3) 单击“下一步”按钮，可选择安装目录，如图 2-5 所示。



图 2-5 选择安装目录

(4) 单击“下一步”按钮，可选择创建桌面快捷方式，如图 2-6 所示。

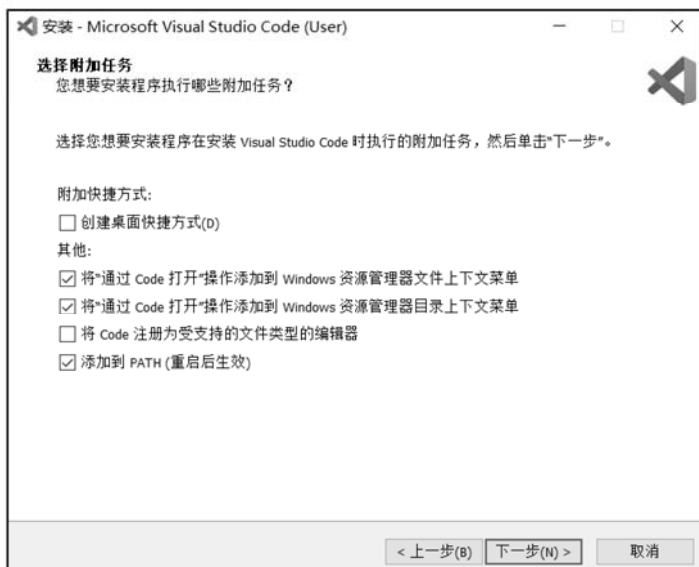


图 2-6 选择附加任务

(5) 单击“下一步”按钮，进入“准备安装”界面，如图 2-7 所示。

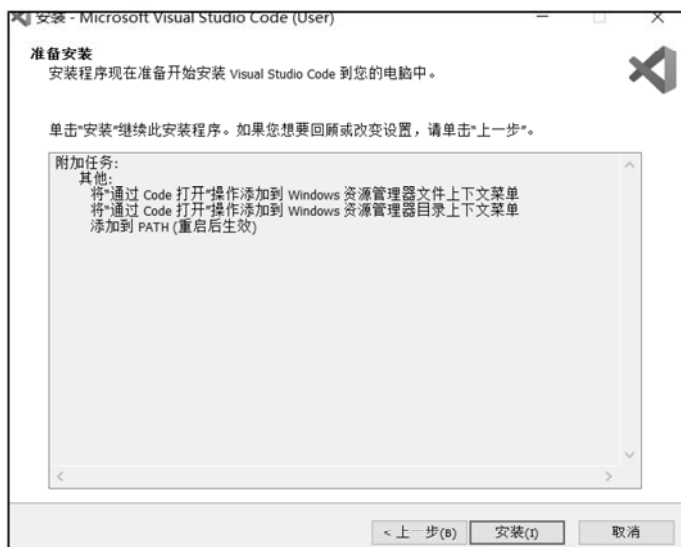


图 2-7 “准备安装”界面

(6) 单击“安装”按钮，即可安装 VS Code，如图 2-8 所示。

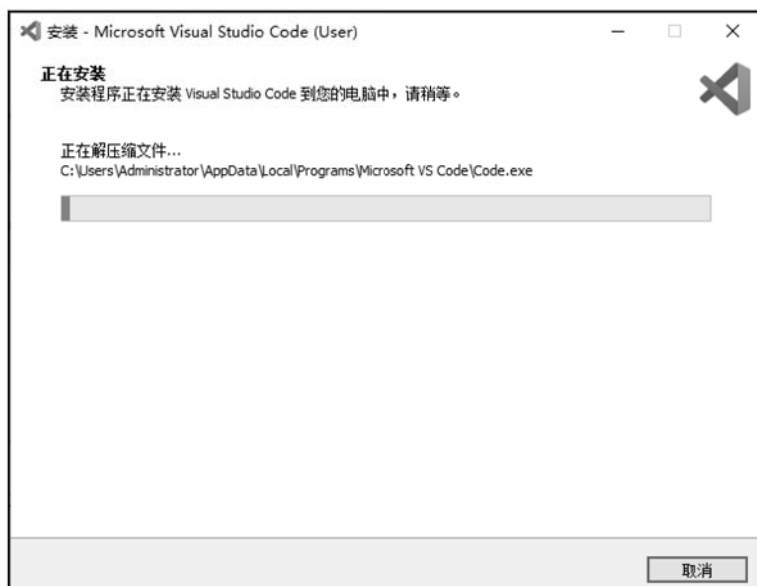


图 2-8 开始安装 VS Code

(7) 安装完毕，界面如图 2-9 所示。



图 2-9 VS Code 安装完毕界面

(8) 如果在上一步选择“运行 Visual Studio Code”复选框，单击“完成”按钮，将自动运行 VS Code，如图 2-10 所示。

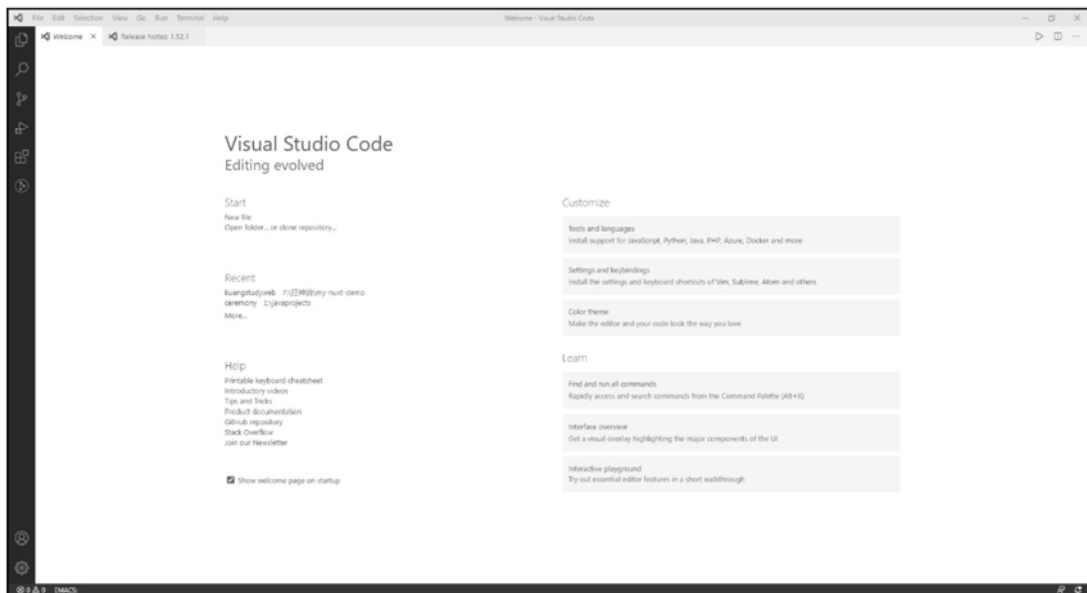


图 2-10 VS Code 初始界面

2. 配置中文界面

(1) 首先安装中文插件：Chinese (Simplified)(简体中文) Language Pack for Visual Studio Code，如图 2-11 所示。

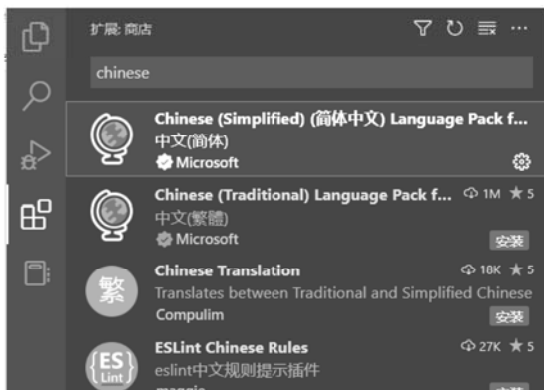


图 2-11 配置中文界面

(2) 单击 Restart 按钮，重启 VS Code。

(3) 重启后，界面如图 2-12 所示，证明安装成功。

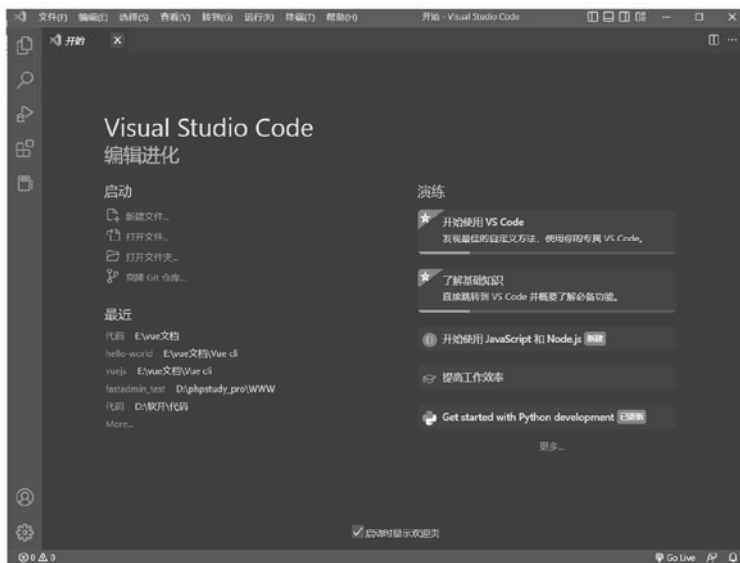


图 2-12 中文界面配置成功

- (4) 有些计算机重启后如果界面没有变化，可单击左边栏中的 **Manage**→**Command Palette** 按钮或按 **Ctrl+Shift+P** 快捷键。
- (5) 在搜索框中输入“**configure display language**”，按 **Enter** 键，如图 2-13 所示。

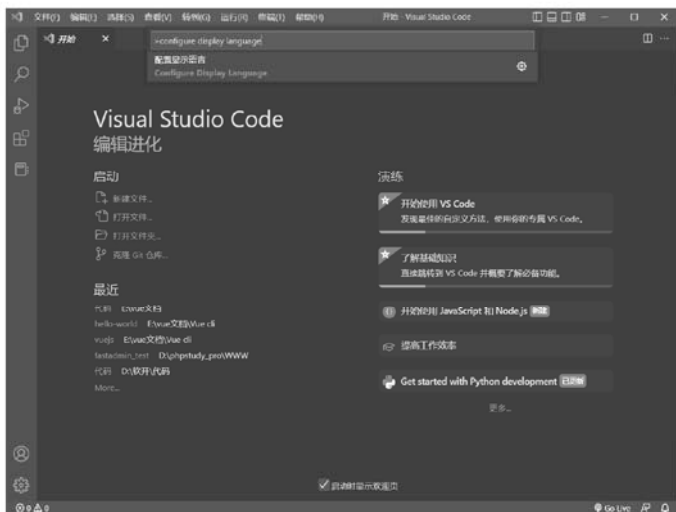


图 2-13 配置显示语言

- (6) 选择“中文(简体)(zh-cn)”，如图 2-14 所示。



图 2-14 选择“中文(简体)(zh-cn)”

(7) 最后重启 VS Code 即可。

3. 安装插件

为方便后续开发，建议安装以下插件：

- HTML CSS Support(代码补全，自动格式)。
- Auto Rename Tag(辅助标签重命名)。
- Live Server(自动刷新浏览器)。

2.1.4 安装 Vue CLI

Vue CLI 是为了让开发者能够快速地进行应用开发而研发的，秉承的是“约定大于配置”思想，简单地说，就是“以够用为配置的基本准则，在此配置下，用户可快速进行业务开发”。

安装 Vue CLI 的步骤如下。

(1) 在命令行窗口中执行如下命令，如图 2-15 所示。

```
npm install -g @vue/cli
```

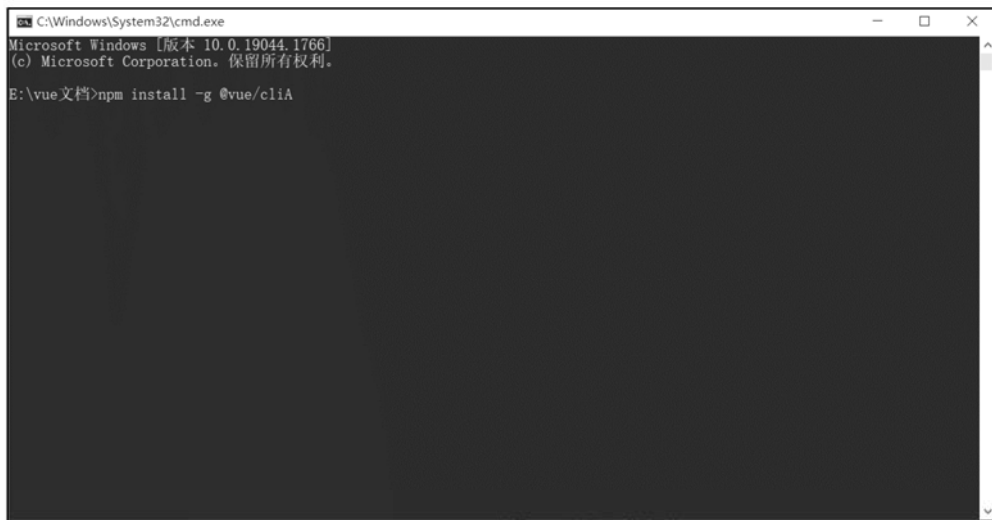


图 2-15 安装 Vue CLI

(2) 输出如图 2-16 所示，则证明安装成功。

```

C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.1766]
(c) Microsoft Corporation. 保留所有权利。

E:\vue文档\Vue cli>npm install -g @vue/cli
npm WARN config global --global, --local are deprecated. Use --location=global instead.
npm WARN config global --global, --local are deprecated. Use --location=global instead.
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated apollo-tracing@0.15.0: The 'apollo-tracing' package is no longer part of Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#tracing for details
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated graphql-extensions@0.15.0: The 'graphql-extensions' API has been removed from Apollo Server 3. Use the plugin API instead: https://www.apollographql.com/docs/apollo-server/integrations/plugins/
npm WARN deprecated apollo-cache-control@0.14.0: The functionality provided by the 'apollo-cache-control' package is built in to 'apollo-server-core' starting with Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#cachecontrol for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated subscriptions-transport-ws@0.9.10: The subscriptions-transport-ws package is no longer maintained. We recommend you use graphql-ws instead. For help migrating Apollo software to graphql-ws, see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws. For general help using graphql-ws, see https://github.com/ensidenjo/graphql-ws/blob/master/README.md
npm WARN deprecated graphql-tools@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema. We recommend it will no longer receive updates. We recommend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc. Check out https://www.graphql-tools.com to learn what package you should use instead

changed 903 packages, and audited 904 packages in 24s
91 packages are looking for funding

```

图 2-16 Vue CLI 安装成功

2.1.5 检查和调试 Vue.js 应用的工具——Vue Devtools

1. Vuex Devtools 插件的作用

跟踪记录每一次改变 state 的状态，从而知道是哪个组件修改了 state。

2. Vue Devtools 的安装

(1) 在 Microsoft Edge 浏览器中选择“扩展”选项，如图 2-17 所示。



图 2-17 选择“扩展”选项

(2) 单击“获取 Microsoft Edge 扩展”链接，如图 2-18 所示。



图 2-18 获取外部扩展

(3) 在搜索框中输入“Vue Devtools”，如图 2-19 所示。

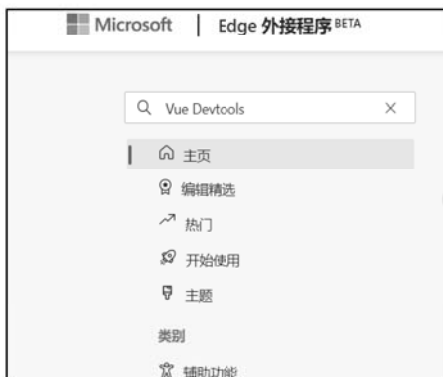


图 2-19 搜索“Vue Devtools”

(4) 找到 Vue.js devtools 并获取，如图 2-20 所示。



图 2-20 选择扩展并获取

2.2 创建 Vue.js 应用 hello-world

CLI 是 Command-Line Interface，即命令行界面，俗称脚手架。Vue CLI 是一个官方发布的 Vue.js 项目脚手架，它可以快速搭建 Vue 开发环境以及对应的 Webpack 配置。可以依靠 Vue UI 通过一套图形化界面管理所有项目。本节将讲述如何用 Vue CLI 创建并运行一个项目。

2.2.1 利用 Vue CLI 初始化 Vue.js 应用 hello-world

(1) 在要创建 Vue.js 应用的文件夹索引框中输入 cmd，如图 2-21 所示。



图 2-21 输入 cmd

(2) 在命令行窗口中执行如下命令初始化应用 hello-world，如图 2-22 所示。

```
vue create hello-world
```



图 2-22 输入命令

(3) 选择 Vue 3 模板 “Default([Vue 3] babel, eslint)”，并按 Enter 键，如图 2-23 所示。



图 2-23 选择模板

(4) 结果如图 2-24 所示，则证明 hello-world 项目已经创建完成。

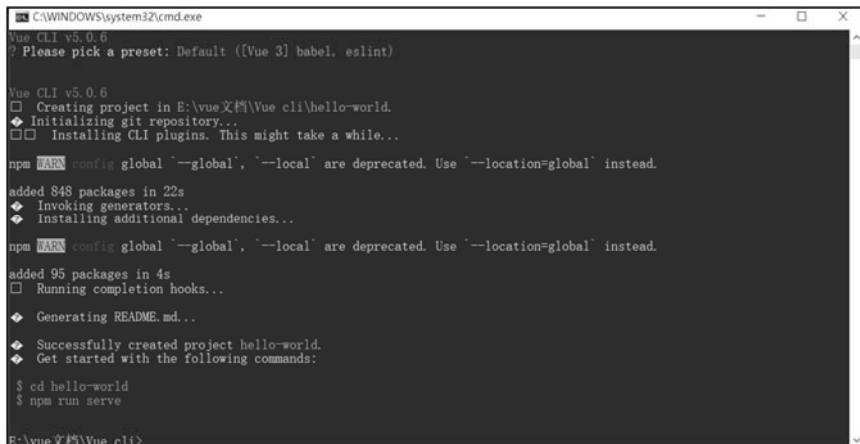


图 2-24 项目创建成功

2.2.2 运行 Vue.js 应用 hello-world

(1) Vue.js 应用需要在 Vue CLI 初始化后的目录下运行，按图 2-25 所示执行如下命令：

```
cd hello-world
npm run serve
```

```
E:\vue文档\Vue cli>cd hello-world  
E:\vue文档\Vue cli\hello-world>npm run serve
```

图 2-25 执行命令

(2) 若看到如图 2-26 所示的输出，则证明项目已经启动。

```
选择 C:\WINDOWS\system32\cmd.exe  
DONE Compiled successfully in 4043ms  
  
App running at:  
- Local: http://localhost:8081/  
- Network: http://192.168.0.101:8081/  
  
Note that the development build is not optimized.  
To create a production build, run npm run build.
```

图 2-26 启动成功

(3) 用浏览器访问 <http://localhost:8081/>，可以看到项目的主页，如图 2-27 所示。

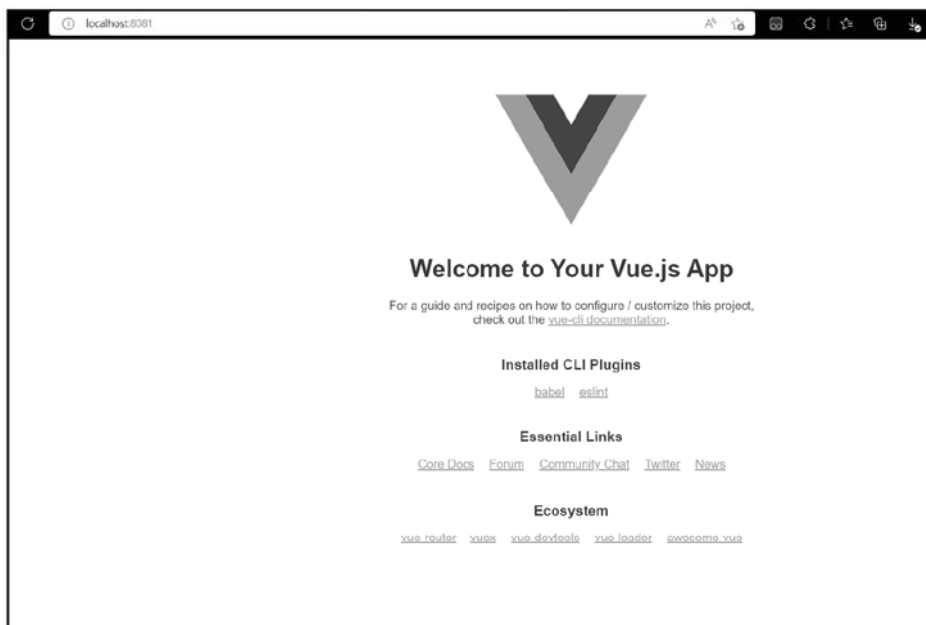


图 2-27 项目主页

2.3 探索 Vue.js 应用

2.3.1 整体项目结构

如图 2-28 所示，Vue.js 的整体项目结构包括项目根目录文件、`node_modules` 目录、`public` 目录和 `src` 目录。

名称	修改日期	类型	大小
node_modules	2022/7/5 3:43	文件夹	
public	2022/7/5 3:37	文件夹	
src	2022/7/5 3:37	文件夹	
.gitignore	2022/7/5 3:37	文本文档	1 KB
babel.config.js	2022/7/5 3:37	JavaScript 文件	1 KB
jsconfig.json	2022/7/5 3:37	JSON 文件	1 KB
package.json	2022/7/5 3:37	JSON 文件	1 KB
package-lock.json	2022/7/5 3:37	JSON 文件	758 KB
README.md	2022/7/5 3:37	MD 文件	1 KB
vue.config.js	2022/7/5 3:37	JavaScript 文件	1 KB

图 2-28 整体项目结构

2.3.2 项目根目录文件

如图 2-29 所示，主要的根目录文件说明如下。

.gitignore	2022/7/5 3:37	文本文档	1 KB
babel.config.js	2022/7/5 3:37	JavaScript 文件	1 KB
jsconfig.json	2022/7/5 3:37	JSON 文件	1 KB
package.json	2022/7/5 3:37	JSON 文件	1 KB
package-lock.json	2022/7/5 3:37	JSON 文件	758 KB
README.md	2022/7/5 3:37	MD 文件	1 KB
vue.config.js	2022/7/5 3:37	JavaScript 文件	1 KB

图 2-29 根目录文件

- (1) `.gitignore`: 用于配置哪些文件不由 `git` 管理。
- (2) `babel.config.js`: Babel 中的配置文件。Babel 是一款 JavaScript 编译器。

(3) `package.json`、`package-lock.json`: `npm` 包管理器的配置文件。`npm install` 读取 `package.json` 创建依赖项列表, 并用 `package-lock.json` 来通知要安装这些依赖项的哪个版本。如果某个依赖项在 `package.json` 中, 但是不在 `package-lock.json` 中, 运行 `npm install` 时会把这个依赖项的确定版本更新到 `package-lock.json` 中, 而不会更新其他依赖项的版本。

(4) `README.md`: 项目的说明文件。一般会详细说明项目的作用、怎么构建、怎么求助等内容。

2.3.3 node_modules 目录

`node_modules` 目录是安装 Node 后用来存放用包管理工具下载安装的包的文件夹, 其目录结构如图 2-30 所示。打开该目录, 可以看到项目所依赖的包非常多。

名称	修改日期	类型	大小
.bin	2022/7/5 3:37	文件夹	
.cache	2022/7/5 3:43	文件夹	
@achrinza	2022/7/5 3:37	文件夹	
@ampproject	2022/7/5 3:37	文件夹	
@babel	2022/7/5 3:37	文件夹	
@eslint	2022/7/5 3:37	文件夹	
@hapi	2022/7/5 3:37	文件夹	
@humanwhocodes	2022/7/5 3:37	文件夹	
@jridgewell	2022/7/5 3:37	文件夹	
@leichtgewicht	2022/7/5 3:37	文件夹	
@node-ipc	2022/7/5 3:37	文件夹	
@nodelib	2022/7/5 3:37	文件夹	
@polka	2022/7/5 3:37	文件夹	
@sideway	2022/7/5 3:37	文件夹	
@soda	2022/7/5 3:37	文件夹	
@trysound	2022/7/5 3:37	文件夹	
@types	2022/7/5 3:37	文件夹	
@vue	2022/7/5 3:37	文件夹	
@webassemblyjs	2022/7/5 3:37	文件夹	
@xtuc	2022/7/5 3:37	文件夹	
accepts	2022/7/5 3:37	文件夹	
acorn	2022/7/5 3:37	文件夹	
acorn-import-assertions	2022/7/5 3:37	文件夹	
acorn-jsx	2022/7/5 3:37	文件夹	
acorn-walk	2022/7/5 3:37	文件夹	
address	2022/7/5 3:37	文件夹	
...	2022/7/5 3:37	文件夹	

图 2-30 node_modules 目录的结构

2.3.4 public 目录

public 目录在下列情况下使用。

- (1) 当在输出中需要指定一个文件的名称时。
- (2) 当图片较多需要动态引用它们的路径时。
- (3) 有些库可能和 Webpack 不兼容，这些库放到这个目录下，以后使用时可将其用一个独立的 `<script>` 标签引入。

public 目录的结构如图 2-31 所示。

名称	修改日期	类型	大小
favicon.ico	2022/7/5 3:37	ICO 文件	5 KB
index.html	2022/7/5 3:37	Microsoft Edge ...	1 KB

图 2-31 public 目录的结构

2.3.5 src 目录

src 目录就是存放项目源码的目录。

- (1) assets: 用于放置静态文件，比如一些图片、JSON 数据等。
- (2) components: 用于放置 Vue 公共组件。目前，该目录下仅有一个 HelloWorld.vue 组件。
- (3) App.vue: 页面入口文件，也是根组件(整个应用只有一个)，可以引用其他 Vue 组件。
- (4) main.js: 程序入口文件，主要作用是初始化 Vue 实例，并使用需要的插件。

src 目录的结构如图 2-32 所示。

名称	修改日期	类型	大小
assets	2022/7/5 3:37	文件夹	
components	2022/7/5 3:37	文件夹	
App.vue	2022/7/5 3:37	VIJE 文件	1 KB
main.js	2022/7/5 3:37	JavaScript 文件	1 KB

图 2-32 src 目录的结构

2.4 在 Vue.js 应用中使用 TypeScript

目前,有基于 Vue 3 Preview 创建项目和基于 Manually select features 创建项目两种方式,可以实现在 Vue 3 应用中使用 TypeScript。

2.4.1 基于 Vue 3 Preview 创建项目

依照 2.2 节创建 Vue.js 应用 hello-world 的流程,可以采用如下所示的步骤实现 Vue.js 对 TypeScript 的支持。

(1) 在应用的根目录下执行如下命令:

```
vue add typescript
```

此时,在命令行中会出现提示框,根据提示选择即可。这里通常选“Y”,如图 2-33 所示。

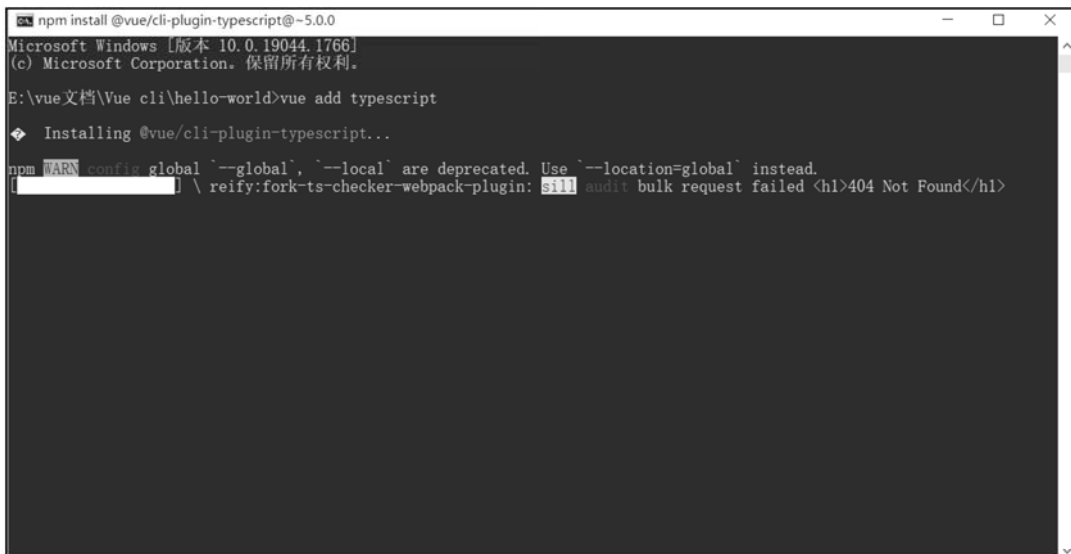
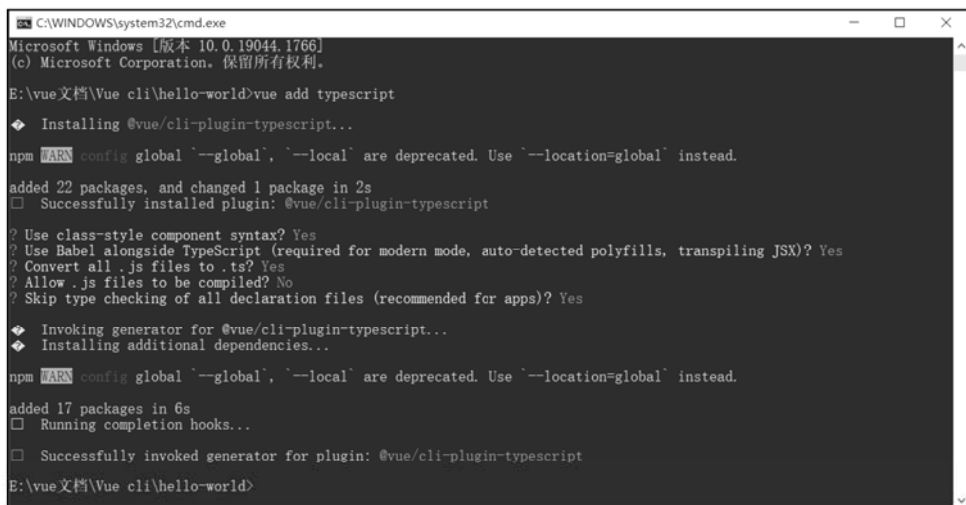


图 2-33 输入命令并选择

(2) 看到如图 2-34 所示的输出,则证明项目已经创建成功。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19044.1766]
(c) Microsoft Corporation. 保留所有权利。

E:\vue文档\Vue cli\hello-world>vue add typescript

  ◆ Installing @vue/cli-plugin-typescript...

npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 22 packages, and changed 1 package in 2s
  □ Successfully installed plugin: @vue/cli-plugin-typescript

? Use class-style component syntax? Yes
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes
? Convert all .js files to .ts? Yes
? Allow .js files to be compiled? No
? Skip type checking of all declaration files (recommended for apps)? Yes

  ◆ Invoking generator for @vue/cli-plugin-typescript...
  ◆ Installing additional dependencies...

npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 17 packages in 6s
  □ Running completion hooks...

  □ Successfully invoked generator for plugin: @vue/cli-plugin-typescript

E:\vue文档\Vue cli\hello-world>
```

图 2-34 项目创建成功

2.4.2 基于 Manually select features 创建项目

如果采用 Manually select features(手动选择)方式创建应用, 则可以直接选择 TypeScript 作为支持选项, 具体步骤如下。

(1) 在要创建项目的文件夹下输入 cmd 命令, 如图 2-35 所示。



图 2-35 输入 cmd 命令

(2) 在命令行窗口中执行如下命令, 初始化项目 “hello-ts”, 如图 2-36 所示。

```
vue create hello-ts
```

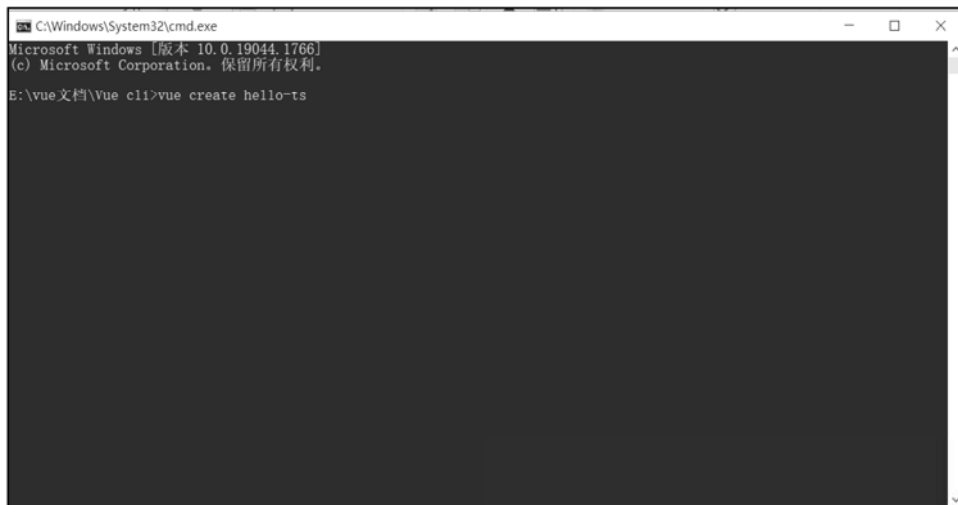


图 2-36 初始化项目

(3) 选择 Manually select features 选项，并按 Enter 键，如图 2-37 所示。



图 2-37 配置 Manually select features

(4) 选择 TypeScript 选项，而后按 Enter 键，如图 2-38 所示。

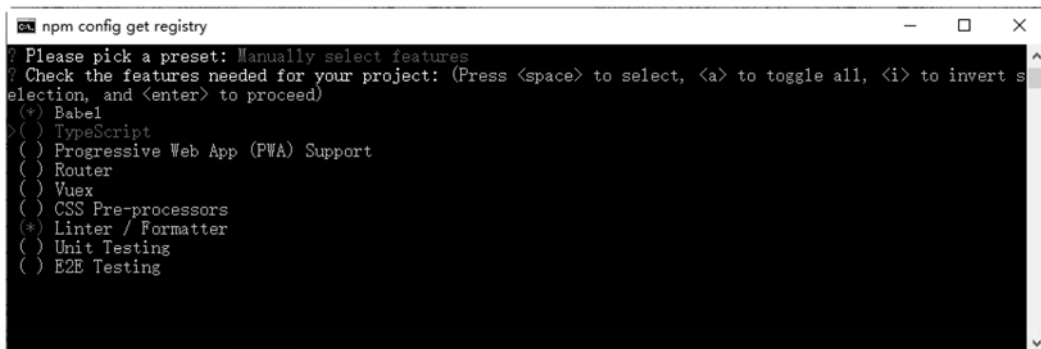


图 2-38 配置 TypeScript

(5) 选择 3.x 选项，而后按 Enter 键，如图 2-39 所示。

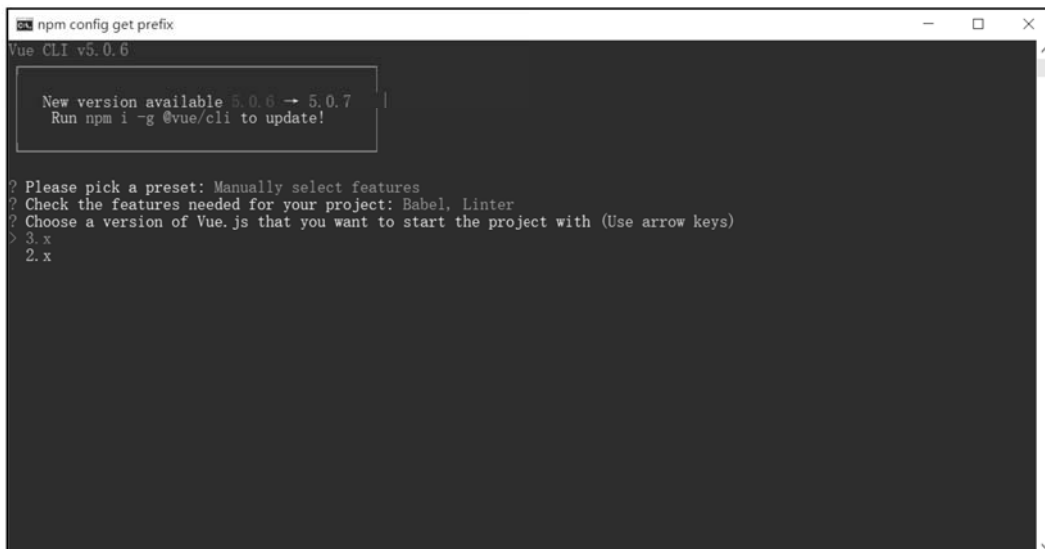


图 2-39 配置 3.x

(6) 选择 ESLint with error prevention only 选项，而后按 Enter 键，如图 2-40 所示。

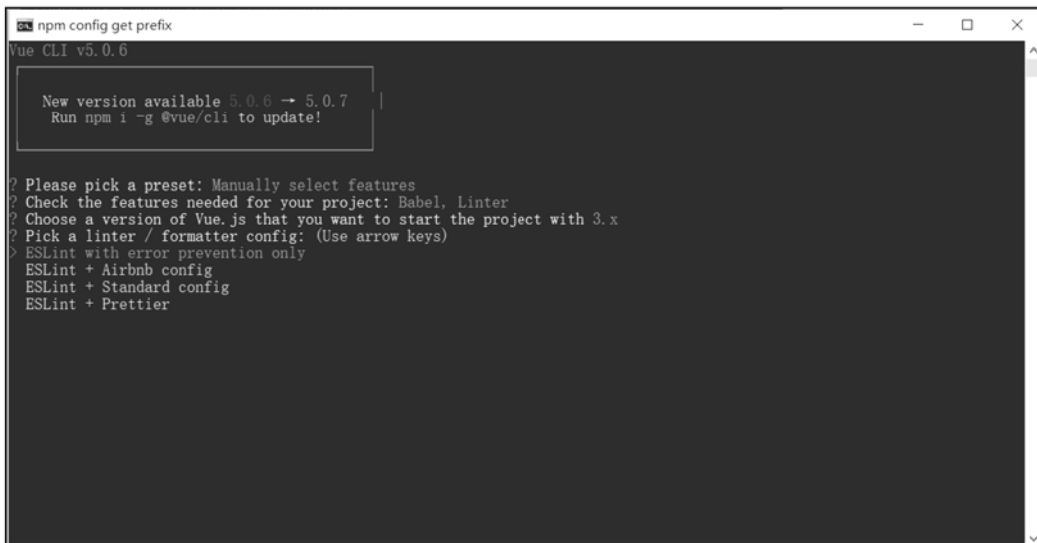


图 2-40 配置 ESLint with error prevention only

(7) 选择 Lint on save 选项，而后按 Enter 键，如图 2-41 所示。

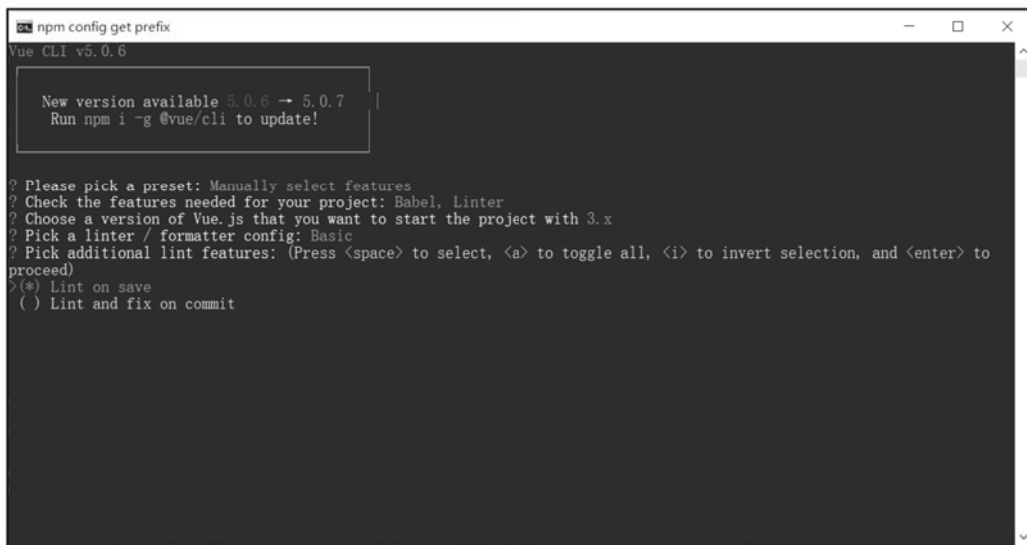


图 2-41 配置 Lint on save

(8) 选择 In dedicated config files 选项，而后按 Enter 键，如图 2-42 所示。

(9) 看到如图 2-43 所示的输出，则证明项目已经创建成功。

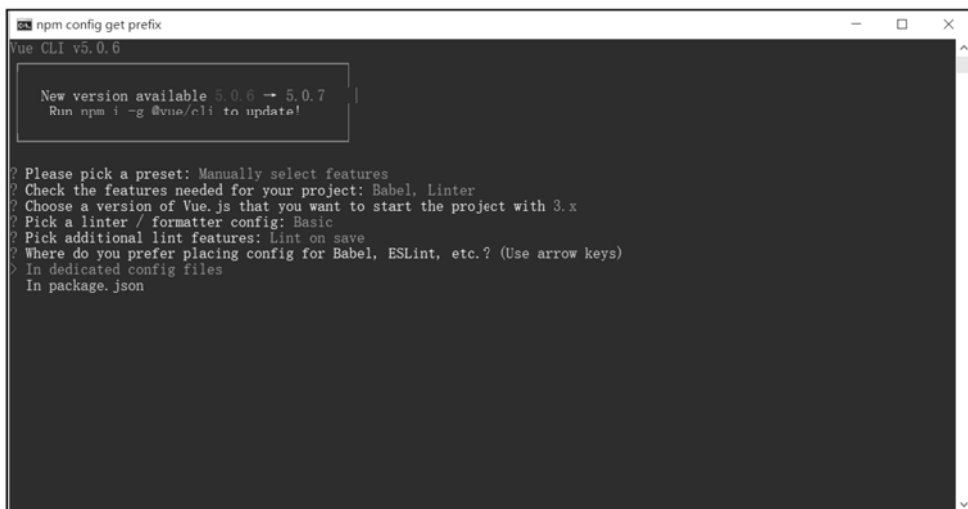


图 2-42 配置 In dedicated config files

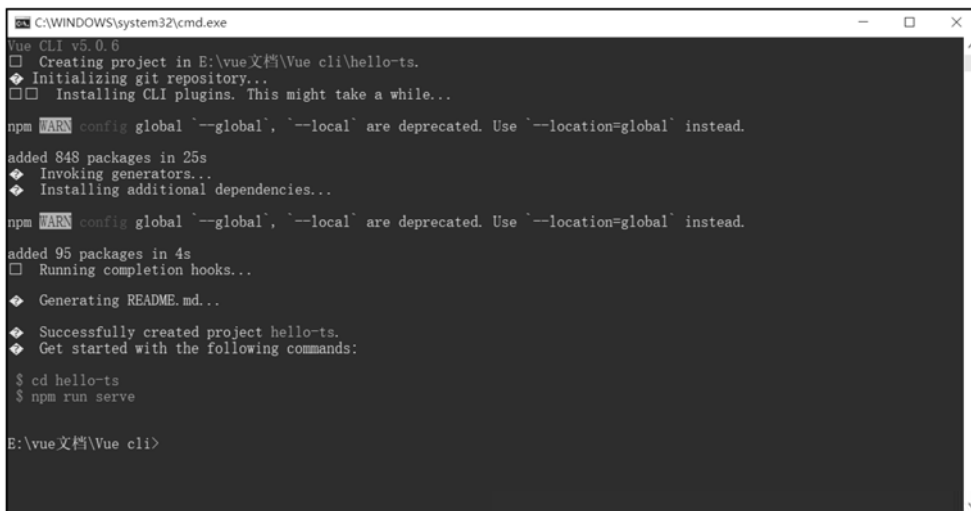


图 2-43 初始化成功

2.4.3 TypeScript 与 JavaScript 应用的差异

相比于 JavaScript 的应用，TypeScript 应用的目录结构如图 2-44 和图 2-45 所示。多了 TypeScript 语言的配置文件 `jsconfig.json`，`package.json` 和 `package-lock.json` 中多了对 TypeScript 等依赖的描述，`main.js` 改为了 `main.ts`，多了 `shims-vue.d.ts` 文件，所有在 Vue 组

件中使用 JavaScript 的地方，都改为了 TypeScript。

名称	修改日期	类型	大小
node_modules	2022/7/5 20:47	文件夹	
public	2022/7/5 20:47	文件夹	
src	2022/7/5 20:47	文件夹	
.browserslistrc	2022/7/5 20:47	BROWSERSLIST...	1 KB
.eslintrc.js	2022/7/5 20:47	JavaScript 文件	1 KB
.gitignore	2022/7/5 20:47	文本文档	1 KB
babel.config.js	2022/7/5 20:47	JavaScript 文件	1 KB
jsconfig.json	2022/7/5 20:47	JSON 文件	1 KB
package.json	2022/7/5 20:47	JSON 文件	1 KB
package-lock.json	2022/7/5 20:47	JSON 文件	758 KB
README.md	2022/7/5 20:47	MD 文件	1 KB
vue.config.js	2022/7/5 20:47	JavaScript 文件	1 KB

图 2-44 TypeScript 应用的目录结构

名称	修改日期	类型	大小
assets	2022/7/5 3:37	文件夹	
components	2022/7/5 3:37	文件夹	
App.vue	2022/7/5 20:32	VUE 文件	1 KB
main.ts	2022/7/5 20:32	TS 文件	1 KB
shims-vue.d.ts	2022/7/5 20:32	TS 文件	1 KB

图 2-45 TypeScript 应用的资源文件

2.5 小结

在本章中，我们对 Vue.js 的 IDE 安装和 Vue CLI 的目录结构有了一个比较详细的了解，足以满足日常使用。本章也对 VS Code 常用的插件进行了介绍，良好的插件可以为我们减轻很多操作负担。Vue CLI 作为 Vue.js 官方推出的框架，内容较多，在日常开发中基本上都会用到。本章从 Vue CLI 的多个版本安装到 Vue CLI 的运行，再到 Vue CLI 的目录结构都做了详细的介绍，将这些知识点融会贯通，会方便我们的理解，这也就是学习的目的。