

Vue综合案例开发

基于 Vue.js + Vite + Bun 技术栈

2025年04月09日

JameLouis

背景 回顾Vue 案例开发 项目部署 总结与展望 问题答疑

课程大纲

课程目标

Vue核心知识回顾

案例开发：个人作品集

项目部署

总结与展望

问题答疑

课程目标

✓ 学会使用 **Vue 3 + Vite + Bun** 开发前端项目

✓ 掌握 **组件化开发** 和 **路由管理**

✓ 完成一个 **响应式个人作品集网站**

✓ 了解 **项目打包与部署** 流程

尤雨溪个人作品集

尤雨溪

[关于我](#)[职业经历](#)[教育经历](#)[项目](#)[演讲](#)[聊天](#)

个人简介

我是 **VoidZero** 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。

在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 **Vue.js** 和前端构建工具 **Vite** 的创建者。我的工作成果大多以开源形式发布在 GitHub 上，如果您从我的开源项目中受益，可以通过 [GitHub Sponsors](#) 给予支持。

除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。

职业经历

Vue.js · Chief Procrastination Officer

谷歌 (Google) · Creative Technologist

Meteor · Core Dev

教育背景

Parsons the New School for Design · MFADT

柯蓋德大學 (Colgate University) · Art & Art History

项目





JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

背景

软件工程分析个人作品集项目

尤雨溪个人作品集信息提取与整理

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

CSS视觉优化：赋予作品集界面优雅的视觉表现

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

软件工程分析个人作品集项目

需求分析

设计

开发

测试

部署与维护

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

设计

开发

测试

部署与维护

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

功能需求：个人简介、职业经历、教育背景、项目经验等

设计

开发

测试

部署与维护

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

功能需求：个人简介、职业经历、教育背景、项目经验等

技术需求：HTML+CSS

设计

开发

测试

部署与维护

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

功能需求：个人简介、职业经历、教育背景、项目经验等

技术需求：HTML+CSS

设计

开发

HTML+CSS构建网站界面

测试

部署与维护

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

功能需求：个人简介、职业经历、教育背景、项目经验等

技术需求：HTML+CSS

设计

开发

HTML+CSS构建网站界面

测试

部署与维护

部署：将网站发布到服务器

软件工程分析个人作品集项目

需求分析

目标定义：展示个人作品、简历，技能等

功能需求：个人简介、职业经历、教育背景、项目经验等

技术需求：HTML+CSS

设计

开发

HTML+CSS构建网站界面

测试

部署与维护

部署：将网站发布到服务器

维护：定期更新内容

尤雨溪个人作品集信息提取与整理

```
1 # 个人简介
2
3 我是 VoidZero 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
4
5 在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 Vue.js 和前端构建工具 Vite 的创建者。我的工作成果大多以开源形式发布在 GitHub 上。
6
7 除了编程，我的兴趣爱好包括：电子游戏（偏爱魂类游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
8
9 ## 职业经历
10
11 - Vue.js · Chief Procrastination Officer
12 - 谷歌 (Google) · Creative Technologist
13 - Meteor · Core Dev
14
15 ## 教育经历
16
17 Parsons the New School for Design · MFADT
18 柯蓋德大學 (Colgate University) · Art & Art History
19
20 ## 项目经验
21
22 - Vue.js - 渐进式JavaScript框架
23 - Vite - 一个超快的前端构建工具，赋能下一代Web应用的发展。
24
25 ## Speaker
26
27 - [面向未来的Vue和Vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
```

尤雨溪个人作品集信息提取与整理

```
1 # 个人简介
2
3 我是 VoidZero 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
4
5 在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 Vue.js 和前端构建工具 vite 的创建者。我的工作成果大多以开源形式发布在 GitHub 上。
6
7 除了编程，我的兴趣爱好包括：电子游戏（偏爱魂类游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
8
9 ## 职业经历
10
11 - Vue.js · Chief Procrastination Officer
12 - 谷歌 (Google) · Creative Technologist
13 - Meteor · Core Dev
14
15 ## 教育经历
16
17 Parsons the New School for Design · MFADT
18 柯蓋德大學 (Colgate University) · Art & Art History
19
20 ## 项目经验
21
22 - Vue.js - 渐进式JavaScript框架
23 - Vite - 一个超快的前端构建工具，赋能下一代Web应用的发展。
24
25 ## Speaker
26
27 - [面向未来的Vue和Vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
```

尤雨溪个人作品集信息提取与整理

```
2
3 我是 VoidZero 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
4
5 在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 Vue.js 和前端构建工具 vite 的创建者。我的工作成果大多以开源形式发布在 GitHub 上。
6
7 除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
8
9 ## 职业经历
10
11 - Vue.js · Chief Procrastination Officer
12 - 谷歌 (Google) · Creative Technologist
13 - Meteor · Core Dev
14
15 ## 教育经历
16
17 Parsons the New School for Design · MFADT
18 柯盖德大學 (Colgate University) · Art & Art History
19
20 ## 项目经验
21
22 - Vue.js - 渐进式JavaScript框架
23 - Vite - 一个超快的前端构建工具，赋能下一代web应用的发展。
24
25 ## Speaker
26
27 - [面向未来的Vue和Vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
28 - [Vue的进化历程 - VueConf 2022](https://www.bilibili.com/video/BV1eK41167Dg?spm_id_from=333.1387.homepage.video_card.click)
29 - [如何开发一个Vue应用 - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
```

尤雨溪个人作品集信息提取与整理

```
2
3 我是 VoidZero 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
4
5 在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 Vue.js 和前端构建工具 vite 的创建者。我的工作成果大多以开源形式发布在 GitHub 上。
6
7 除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
8
9 ## 职业经历
10
11 - Vue.js · Chief Procrastination Officer
12 - 谷歌 (Google) · Creative Technologist
13 - Meteor · Core Dev
14
15 ## 教育经历
16
17 Parsons the New School for Design · MFADT
18 柯盖德大學 (Colgate University) · Art & Art History
19
20 ## 项目经验
21
22 - Vue.js - 渐进式JavaScript框架
23 - Vite - 一个超快的前端构建工具，赋能下一代web应用的发展。
24
25 ## Speaker
26
27 - [面向未来的Vue和vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
28 - [Vue的进化历程 - VueConf 2022](https://www.bilibili.com/video/BV1eK41167Dg?spm_id_from=333.1387.homepage.video_card.click)
29 - [面向未来的Vue和vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
```

尤雨溪个人作品集信息提取与整理

```
2
3 我是 VoidZero 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
4
5 在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 Vue.js 和前端构建工具 vite 的创建者。我的工作成果大多以开源形式发布在 GitHub 上。
6
7 除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
8
9 ## 职业经历
10
11 - Vue.js · Chief Procrastination Officer
12 - 谷歌 (Google) · Creative Technologist
13 - Meteor · Core Dev
14
15 ## 教育经历
16
17 Parsons the New School for Design · MFADT
18 柯盖德大學 (Colgate University) · Art & Art History
19
20 ## 项目经验
21
22 - Vue.js - 渐进式JavaScript框架
23 - Vite - 一个超快的前端构建工具，赋能下一代Web应用的发展。
24
25 ## Speaker
26
27 - [面向未来的Vue和Vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
28 - [Vue的进化历程 - VueConf 2022](https://www.bilibili.com/video/BV1eK41167Dg?spm_id_from=333.1387.homepage.video_card.click)
29 - [面向未来的Vue和Vite - VueConf 2024](https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click)
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
1 <body>
2   <header>
3     <h1>尤雨溪</h1>
4     <nav>
5       <a href="#about">关于我</a>
6       <a href="#work">职业经历</a>
7       <a href="#education">教育经历</a>
8       <a href="#project">项目</a>
9       <a href="#contact">演讲</a>
10    </nav>
11  </header>
12  <main>
13    <article id="about">
14      <h2>个人简介</h2>
15      <p>我是 <strong>VoidZero</strong> 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。</p>
16      <p>
17        在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 <strong>Vue.js</strong> 和前端构建工具 <strong>Vite</strong> 的创始人。
18      </p>
19      <p>
20        除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。
21      </p>
22    </article>
23    <article id="work">
24      <h2>职业经历</h2>
25      <ul>
26        <li>Vue.js · Chief Procrastination Officer </li>
27        <li>谷歌 (Google) · Creative Technologist</li>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
1
2
3
4
5     <a href="#about">关于我</a>
6     <a href="#work">职业经历</a>
7     <a href="#education">教育经历</a>
8     <a href="#project">项目</a>
9     <a href="#contact">演讲</a>
10
11 </nav>
12 </header>
13 <main>
14     <article id="about">
15         <h2>个人简介</h2>
16         <p>我是 <strong>VoidZero</strong> 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。</p>
17         <p>在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 <strong>Vue.js</strong> 和前端构建工具 <strong>Vite</strong> 的创始人。</p>
18         <p>除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。</p>
19     </article>
20     <article id="work">
21         <h2>职业经历</h2>
22         <ul>
23             <li>Vue.js · Chief Procrastination Officer </li>
24             <li>谷歌 (Google) · Creative Technologist</li>
25             <li>Meteor · Core Dev</li>
26         </ul>
27     </article>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
13 </article id="about">
14 <h2>个人简介</h2>
15 <p>我是 <strong>VoidZero</strong> 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。</p>
16 <p>
17   在创立 VoidZero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架 <strong>Vue.js</strong> 和前端构建工具 <strong>Vite</strong> 的
18   主要贡献者。
19 </p>
20 <p>除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。</p>
21 </article>
22 <article id="work">
23 <h2>职业经历</h2>
24 <ul>
25 <li>Vue.js · Chief Procrastination Officer </li>
26 <li>谷歌 (Google) · Creative Technologist</li>
27 <li>Meteor · Core Dev</li>
28 </ul>
29 </article>
30 <article id="education">
31 <h2>教育背景</h2>
32 <ul>
33 <li>Parsons the New School for Design · MFADT</li>
34 <li>柯蓋德大學 (Colgate University) · Art & Art History</li>
35 </ul>
36 </article>
37 <article id="project">
38 <h2>项目</h2>
39 <ul>
40 <li>“<strong>Vue.js</strong>” 谷歌 (Google) · Creative Technologist</li>
41 </ul>
42 </article>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
21     </p>
22   </article>
23   <article id="work">
24     <h2>职业经历</h2>
25     <ul>
26       <li>Vue.js · Chief Procrastination Officer </li>
27       <li>谷歌 (Google) · Creative Technologist</li>
28       <li>Meteor · Core Dev</li>
29     </ul>
30   </article>
31   <article id="education">
32     <h2>教育背景</h2>
33     <ul>
34       <li>Parsons the New School for Design · MFADT</li>
35       <li>柯蓋德大學 (Colgate University) · Art & Art History</li>
36     </ul>
37   </article>
38   <article id="project">
39     <h2>项目</h2>
40     <div class="project">
41       <section>
42         <div class="card">
43           <div class="card-image">
44             
47             <h2 class="card-title">Vue.js</h2>
48             <p>谷歌 (Google) · Creative Technologist</p>
49           </div>
50         </div>
51       </section>
52     </div>
53   </article>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
38     <article id="project">
39       <h2>项目</h2>
40       <div class="project">
41         <section>
42           <div class="card">
43             <div class="card-image">
44               
47               <h2 class="card-title">Vue.js</h2>
48               <p class="card-text">渐进式JavaScript框架。易学易用，性能出色，适用场景丰富的Web前端框架。</p>
49               <div class="card-meta">
50                 
51                 
52                 
53               </div>
54               <div class="card-actions">
55                 <a href="https://cn.vuejs.org/">
56                   查看更多
57                 </a>
58               </div>
59             </div>
60           </div>
61         </section>
62         <section>
63           <div class="card">
64             <h2 class="card-title">创造未来，创造技术</h2>
```

```

35     <li>柯蓋德大學 (Colgate University) · Art & Art History</li>
36 </ul>
37 </article>
38 <article id="project">
39   <h2>项目</h2>
40   <div class="project">
41     <section>
42       <div class="card">
43         <div class="card-image">
44           
47           <h2 class="card-title">Vue.js</h2>
48           <p class="card-text">渐进式JavaScript框架。易学易用，性能出色，适用场景丰富的Web前端框架。</p>
49           <div class="card-meta">
50             
51             
52             
53           </div>
54           <div class="card-actions">
55             <a href="https://cn.vuejs.org/">
56               查看更多
57             </a>
58           </div>
59         </div>
60       </div>
61     </section>
62   </div>
63 </article>
64 <li>谷歌 (Google) · Creative Technologist</li>

```


HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
72         
73         
74     </div>
75     <div class="card-actions">
76         <a href="https://cn.vuejs.org/">
77             查看更多
78         </a>
79     </div>
80 </div>
81 </div>
82 </section>
83 </div>
84 </article>
85 <article>
86     <h2>演讲</h2>
87     <ul>
88         <li><a href="https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click">面向未来的Vue和Vite - Vue
89         </li>
90         <li><a href="https://www.bilibili.com/video/BV1eK41167Dg?spm_id_from=333.1387.homepage.video_card.click">Vue的进化历程 - VueConf 202
91         </li>
92     </ul>
93 </article>
94 </main>
95 <footer>
96     <p>&copy;2025 <a href="https://evanyou.me/">evanyou</a>. All rights reserved.</p>
97 </footer>
98 </body>
99 </html>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
72         
73         
74     </div>
75     <div class="card-actions">
76         <a href="https://cn.vuejs.org/">
77             查看更多
78         </a>
79     </div>
80 </div>
81 </div>
82 </section>
83 </div>
84 </article>
85 <article>
86     <h2>演讲</h2>
87     <ul>
88         <li><a href="https://www.bilibili.com/video/BV1q1421b7YR?spm_id_from=333.1387.homepage.video_card.click">面向未来的Vue和Vite - Vue
89         </li>
90         <li><a href="https://www.bilibili.com/video/BV1eK41167Dg?spm_id_from=333.1387.homepage.video_card.click">Vue的进化历程 - VueConf 202
91         </li>
92     </ul>
93 </article>
94 </main>
95 <footer>
96     <p>&copy;2025 <a href="https://evanyou.me/">evanyou</a>. All rights reserved.</p>
97 </footer>
98 </body>
99 </html>
```

HTML语义化设计：构建尤雨溪个人作品集的结构化框架

```
1 <body>
2   <header>
3     <h1>尤雨溪</h1>
4     <nav>
5       <a href="#about">关于我</a>
6       <a href="#work">职业经历</a>
7       <a href="#education">教育经历</a>
8       <a href="#project">项目</a>
9       <a href="#contact">演讲</a>
10    </nav>
11  </header>
12  <main>
13    <article id="about">
14      <h2>个人简介</h2>
15      <p>我是 <strong>VoidZero</strong> 的创始人，我们致力于为 JavaScript 和 TypeScript
16      <p>
17        在创立 voidzero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框
18      </p>
19      <p>
20        除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、
21      </p>
22    </article>
23    <article id="work">
24      <h2>职业经历</h2>
25      <ul>
26        <li>Vue.js · Chief Procrastination Officer </li>
27        <li>谷歌 (Google) · Creative Technologist</li>
```



CSS视觉优化：赋予作品集界面优雅的视觉表现

```
1 /* Reset and Base Styles */
2 * {
3   margin: 0;
4   padding: 0;
5   box-sizing: border-box;
6 }
7
8 body {
9   font-family: system-ui;
10  line-height: 1.6;
11  color: #333;
12  background-color: #f9f9f9;
13  max-width: 780px;
14  margin: 0 auto;
15 }
16
17 a {
18   text-decoration: none;
19   color: #3498db;
20   transition: color 0.3s ease;
21 }
22
23 a:hover {
24   color: #2980b9;
25 }
26
27 p {
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
5  .don-billing: before { content: 'Don Billing'; border-bottom: 1px solid #333; width: 100%; height: 1em; margin-bottom: 10px; }
6  }
7
8  body {
9    font-family: system-ui;
10   line-height: 1.6;
11   color: #333;
12   background-color: #f9f9f9;
13   max-width: 780px;
14   margin: 0 auto;
15 }
16
17 a {
18   text-decoration: none;
19   color: #3498db;
20   transition: color 0.3s ease;
21 }
22
23 a:hover {
24   color: #2980b9;
25 }
26
27 p {
28   padding-bottom: 1rem;
29 }
30
31 /* Header Styles */
32 p { margin: 0; }
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
28 padding-bottom: 1rem;
29 }
30
31 /* Header Styles */
32 header {
33   background-color: #fff;
34   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
35   padding: 0rem 2rem;
36   position: sticky;
37   top: 0;
38   z-index: 100;
39   display: flex;
40   justify-content: space-between;
41   align-items: center;
42 }
43
44 nav {
45   display: flex;
46   justify-content: space-between;
47   align-items: center;
48 }
49
50 nav a {
51   margin-left: 1.5rem;
52 }
53
54 main {
55   padding-top: 1rem;
56 }
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
50 nav a {
51   margin-left: 1.5rem;
52 }
53
54 main {
55   margin-top: 2rem;
56 }
57
58 article {
59   margin-bottom: 1rem;
60   background-color: #fff;
61   padding: 2rem;
62   border-radius: 8px;
63   box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);
64 }
65
66 article h2 {
67   margin-bottom: 1rem;
68   color: #2c3e50;
69   border-bottom: 2px solid #f1f1f1;
70   padding-bottom: 0.5rem;
71 }
72
73 ul {
74   list-style-type: none;
75 }
76
77 p {
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
65
66 article h2 {
67   margin-bottom: 1rem;
68   color: #2c3e50;
69   border-bottom: 2px solid #f1f1f1;
70   padding-bottom: 0.5rem;
71 }
72
73 ul {
74   list-style-type: none;
75 }
76
77 ul li {
78   margin-bottom: 0.5rem;
79   padding: 0.5rem;
80   background-color: #f8f9fa;
81   border-left: 4px solid #3498db;
82 }
83
84 .project {
85   display: flex;
86   justify-content: space-between;
87   flex-wrap: wrap;
88   gap: 1rem;
89 }
90
91 p {
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
73 ul {
74   list-style-type: none;
75 }
76
77 ul li {
78   margin-bottom: 0.5rem;
79   padding: 0.5rem;
80   background-color: #f8f9fa;
81   border-left: 4px solid #3498db;
82 }
83
84 .project {
85   display: flex;
86   justify-content: space-between;
87   flex-wrap: wrap;
88   gap: 1rem;
89 }
90
91 .card {
92   max-width: 320px;
93   margin: 0 auto;
94   background: white;
95   border-radius: 12px;
96   overflow: hidden;
97   box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
98   transition: transform 0.3s ease, box-shadow 0.3s ease;
99 }
100
101 p {
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

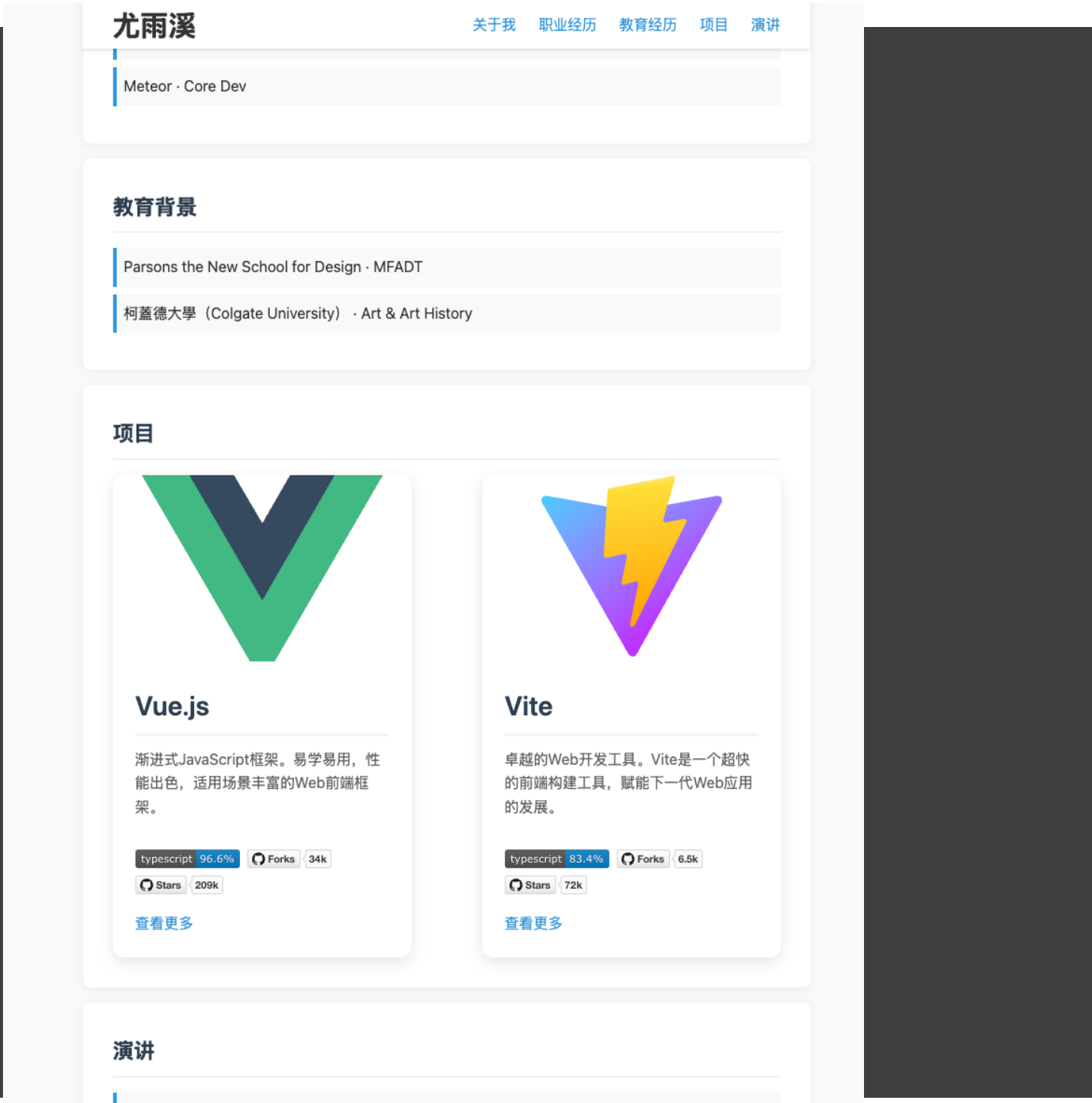
```
91 .card {
92   max-width: 320px;
93   margin: 0 auto;
94   background: white;
95   border-radius: 12px;
96   overflow: hidden;
97   box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
98   transition: transform 0.3s ease, box-shadow 0.3s ease;
99 }
100
101 .card:hover {
102   transform: translateY(-5px);
103   box-shadow: 0 8px 25px rgba(0, 0, 0, 0.15);
104 }
105
106 /* Card Image */
107 .card-image {
108   position: relative;
109   height: 200px;
110   overflow: hidden;
111 }
112
113 .card-image img {
114   width: 100%;
115   height: 100%;
116   object-fit: fit;
117 }
```

CSS视觉优化：赋予作品集界面优雅的视觉表现

```
134 }
135
136 /* Card Text */
137 .card-text {
138   color: #666;
139   margin-bottom: 1rem;
140   font-size: 1rem;
141 }
142
143 /* Card Meta */
144 .card-meta {
145   display: flex;
146   flex-wrap: wrap;
147   gap: 0.5rem;
148   color: #888;
149   font-size: 0.9rem;
150   margin-bottom: 1.2rem;
151 }
152
153 /* Footer Styles */
154 footer {
155   text-align: center;
156   padding: 2rem;
157   background-color: #2c3e50;
158   color: #ecf0f1;
159   margin-top: 3rem;
160 }
161
```

CSS视觉优化： 赋予作品集界面优雅的视觉表现

```
1  /* Reset and Base Styles */
2  * {
3    margin: 0;
4    padding: 0;
5    box-sizing: border-box;
6  }
7
8  body {
9    font-family: system-ui;
10   line-height: 1.6;
11   color: #333;
12   background-color: #f9f9f9;
13   max-width: 780px;
14   margin: 0 auto;
15 }
16
17 a {
18   text-decoration: none;
19   color: #3498db;
20   transition: color 0.3s ease;
21 }
22
23 a:hover {
24   color: #2980b9;
25 }
26
27 p {
```



JameLouis

背景 回顾Vue 案例开发 项目部署 总结与展望 问题答疑

尤雨溪作品集 - codepen

Vue核心知识回顾

响应式基础

最常用的模板语法

组件通信（父子组件）

组合式 API 核心

5个常用的指令

响应式基础

```
1 <script setup>
2 import { ref } from 'vue'
3
4 // 响应式数据
5 const count = ref(0)
6
7 // 方法
8 const increment = () => {
9   count.value++ // 注意要.value
10 }
11 </script>
12
13 <template>
14   <button @click="increment">{{ count }}</button>
15 </template>
```

响应式基础

```
1 <script setup>
2 import { ref } from 'vue'
3
4 // 响应式数据
5 const count = ref(0)
6
7 // 方法
8 const increment = () => {
9   count.value++ // 注意要.value
10 }
11 </script>
12
13 <template>
14   <button @click="increment">{{ count }}</button>
15 </template>
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

响应式基础

```
1 <script setup>
2 import { ref } from 'vue'
3
4 // 响应式数据
5 const count = ref(0)
6
7 // 方法
8 const increment = () => {
9   count.value++ // 注意要.value
10 }
11 </script>
12
13 <template>
14   <button @click="increment">{{ count }}</button>
15 </template>
```

响应式基础

```
1 <script setup>
2 import { ref } from 'vue'
3
4 // 响应式数据
5 const count = ref(0)
6
7 // 方法
8 const increment = () => {
9   count.value++ // 注意要.value
10 }
11 </script>
12
13 <template>
14   <button @click="increment">{{ count }}</button>
15 </template>
```

响应式基础

```
1 <script setup>
2 import { ref } from 'vue'
3
4 // 响应式数据
5 const count = ref(0)
6
7 // 方法
8 const increment = () => {
9   count.value++ // 注意要.value
10 }
11 </script>
12
13 <template>
14   <button @click="increment">{{ count }}</button>
15 </template>
```

最常用的模板语法

```
1 <template>
2   <!-- 插值表达式 -->
3   <p>计算值: {{ count + 1 }}</p>
4
5   <!-- 条件渲染 -->
6   <div v-if="show">显示内容</div>
7
8   <!-- 列表渲染 -->
9   <ul>
10    <li v-for="item in items" :key="item.id">
11      {{ item.name }}
12    </li>
13  </ul>
14
15  <!-- 双向绑定 -->
16  <input v-model="message">
17 </template>
```

最常用的模板语法

```
1 <template>
2   <!-- 插值表达式 -->
3   <p>计算值: {{ count + 1 }}</p>
4
5   <!-- 条件渲染 -->
6   <div v-if="show">显示内容</div>
7
8   <!-- 列表渲染 -->
9   <ul>
10    <li v-for="item in items" :key="item.id">
11      {{ item.name }}
12    </li>
13  </ul>
14
15  <!-- 双向绑定 -->
16  <input v-model="message">
17 </template>
```

最常用的模板语法

```
1 <template>
2   <!-- 插值表达式 -->
3   <p>计算值: {{ count + 1 }}</p>
4
5   <!-- 条件渲染 -->
6   <div v-if="show">显示内容</div>
7
8   <!-- 列表渲染 -->
9   <ul>
10    <li v-for="item in items" :key="item.id">
11      {{ item.name }}
12    </li>
13  </ul>
14
15  <!-- 双向绑定 -->
16  <input v-model="message">
17 </template>
```

最常用的模板语法

```
1 <template>
2   <!-- 插值表达式 -->
3   <p>计算值: {{ count + 1 }}</p>
4
5   <!-- 条件渲染 -->
6   <div v-if="show">显示内容</div>
7
8   <!-- 列表渲染 -->
9   <ul>
10    <li v-for="item in items" :key="item.id">
11      {{ item.name }}
12    </li>
13  </ul>
14
15  <!-- 双向绑定 -->
16  <input v-model="message">
17 </template>
```

最常用的模板语法

```
1 <template>
2   <!-- 插值表达式 -->
3   <p>计算值: {{ count + 1 }}</p>
4
5   <!-- 条件渲染 -->
6   <div v-if="show">显示内容</div>
7
8   <!-- 列表渲染 -->
9   <ul>
10    <li v-for="item in items" :key="item.id">
11      {{ item.name }}
12    </li>
13  </ul>
14
15  <!-- 双向绑定 -->
16  <input v-model="message">
17 </template>
```

组件通信（父子组件）

```
1 // 子组件 Child.vue
2 <script setup>
3 defineProps(['title']) // 接收父组件props
4 const emit = defineEmits(['update']) // 定义事件
5 </script>
6
7 // 父组件 Parent.vue
8 <Child :title="标题" @update="handleUpdate"></Child>
```

组件通信（父子组件）

```
1 // 子组件 Child.vue
2 <script setup>
3 defineProps(['title']) // 接收父组件props
4 const emit = defineEmits(['update']) // 定义事件
5 </script>
6
7 // 父组件 Parent.vue
8 <Child :title="标题" @update="handleUpdate"></Child>
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

组件通信（父子组件）

```
1 // 子组件 Child.vue
2 <script setup>
3 defineProps(['title']) // 接收父组件props
4 const emit = defineEmits(['update']) // 定义事件
5 </script>
6
7 // 父组件 Parent.vue
8 <Child :title="标题" @update="handleUpdate"></Child>
```

组件通信（父子组件）

```
1 // 子组件 Child.vue
2 <script setup>
3 defineProps(['title']) // 接收父组件props
4 const emit = defineEmits(['update']) // 定义事件
5 </script>
6
7 // 父组件 Parent.vue
8 <Child :title="标题" @update="handleUpdate"></Child>
```

[JameLouis](#)[背景](#)[回顾Vue](#)[案例开发](#)[项目部署](#)[总结与展望](#)[问题答疑](#)

组合式API核心

只需要掌握这4个函数：

```
1 import { ref, reactive, computed, watch } from 'vue'
2
3 const count = ref(0) // 基本类型
4 const user = reactive({ name: '张三' }) // 对象类型
5
6 const double = computed(() => count.value * 2) // 计算属性
7
8 watch(count, (newVal) => { // 监听器
9   console.log('count变化了:', newVal)
10 })
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

组合式API核心

只需要掌握这4个函数：

```
1 import { ref, reactive, computed, watch } from 'vue'
2
3 const count = ref(0) // 基本类型
4 const user = reactive({ name: '张三' }) // 对象类型
5
6 const double = computed(() => count.value * 2) // 计算属性
7
8 watch(count, (newVal) => { // 监听器
9   console.log('count变化了:', newVal)
10 })
```

组合式API核心

只需要掌握这4个函数：

```
1 import { ref, reactive, computed, watch } from 'vue'
2
3 const count = ref(0) // 基本类型
4 const user = reactive({ name: '张三' }) // 对象类型
5
6 const double = computed(() => count.value * 2) // 计算属性
7
8 watch(count, (newVal) => { // 监听器
9   console.log('count变化了:', newVal)
10 })
```

组合式API核心

只需要掌握这4个函数：

```
1 import { ref, reactive, computed, watch } from 'vue'
2
3 const count = ref(0) // 基本类型
4 const user = reactive({ name: '张三' }) // 对象类型
5
6 const double = computed(() => count.value * 2) // 计算属性
7
8 watch(count, (newVal) => { // 监听器
9   console.log('count变化了:', newVal)
10 })
```

5个最常用的指令

v-if/v-show
条件显示

v-for
列表循环

v-bind/:
Attribute绑定

v-on/@click
事件绑定

v-model
双向数据绑定

案例开发

技术栈简介

项目开发环境搭建

重构尤雨溪个人作品集 - Vue组件

重构项目组件 - 响应式 + v-for + 组件通信 + v-bind

新特性：随机一言 - watch + @click + rest api

新特性: 聊天机器人 - vue router + :class + v-if + v-model

技术栈

Vue.js 3.0

渐进式 JavaScript 框架
组合式 API (Composition API)
更快的渲染性能

Vite

极速的前端构建工具
基于原生 ES Modules
秒级启动 & 热更新

Bun

新一代 JavaScript 运行时 & 包管理工具
比 npm/yarn/pnpm 更快
内置测试、打包、脚本运行功能

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

Vite + Vue项目开发环境搭建

```
1 # Linux & macOS
2 $ curl -fsSL https://bun.sh/install | bash
3 # Windows
4 > powershell -c "irm bun.sh/install.ps1 | iex"
5
6 # Test
7 $ bun --version
8 1.2.5
9
10 # create a vue+vite project
11 $ bun create vite my-vue-app --template vue
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

Vite + Vue项目开发环境搭建

```
1 # Linux & macOS
2 $ curl -fsSL https://bun.sh/install | bash
3 # Windows
4 > powershell -c "irm bun.sh/install.ps1 | iex"
5
6 # Test
7 $ bun --version
8 1.2.5
9
10 # create a vue+vite project
11 $ bun create vite my-vue-app --template vue
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

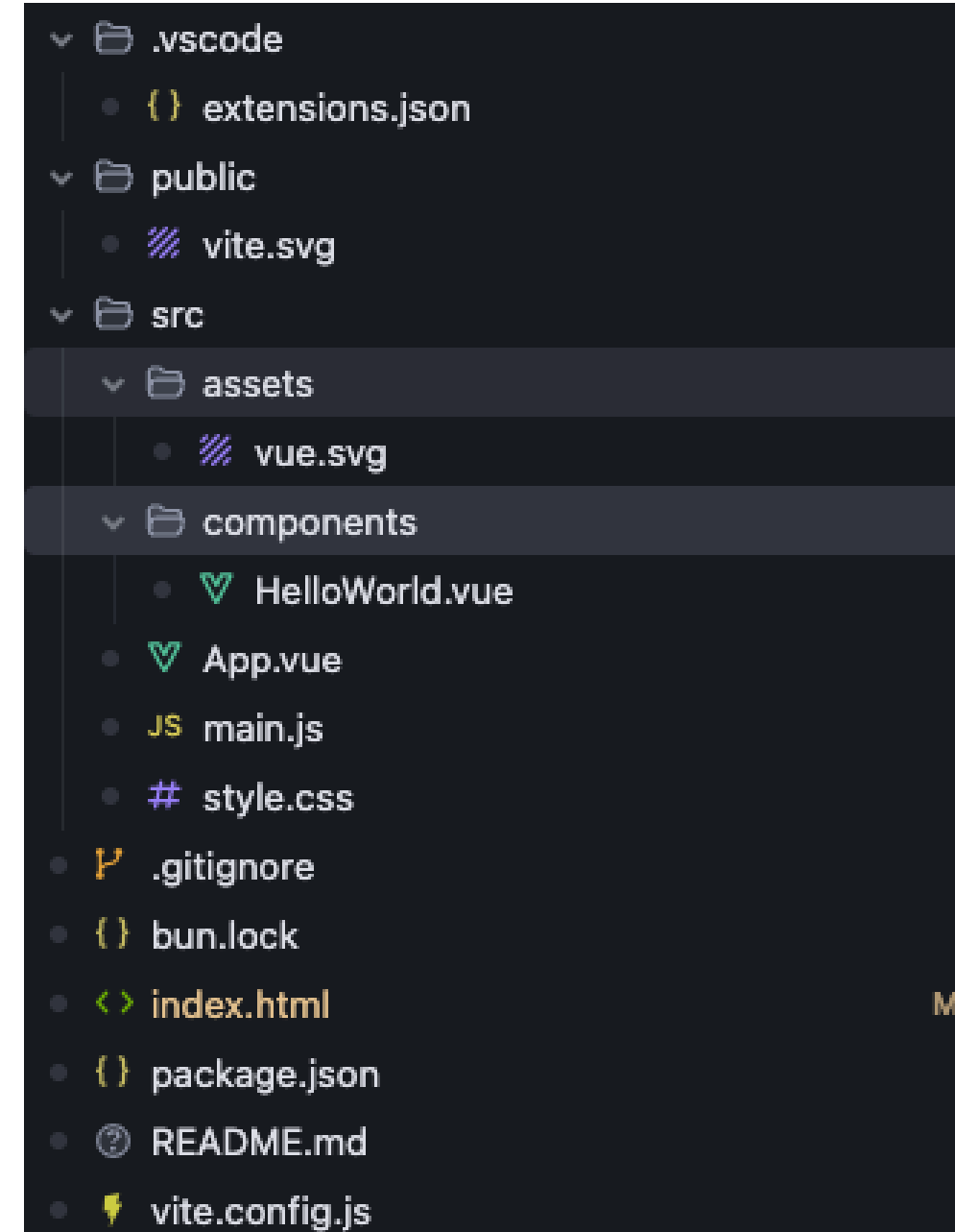
问题答疑

Vite + Vue项目开发环境搭建

```
1 # Linux & macOS
2 $ curl -fsSL https://bun.sh/install | bash
3 # Windows
4 > powershell -c "irm bun.sh/install.ps1 | iex"
5
6 # Test
7 $ bun --version
8 1.2.5
9
10 # create a vue+vite project
11 $ bun create vite my-vue-app --template vue
```

Vite + Vue项目开发环境搭建

```
1 # Linux & macOS
2 $ curl -fsSL https://bun.sh/install | bash
3 # Windows
4 > powershell -c "irm bun.sh/install.ps1 | iex"
5
6 # Test
7 $ bun --version
8 1.2.5
9
10 # create a vue+vite project
11 $ bun create vite my-vue-app --template vue
```



- .vscode
 - {} extensions.json
- public
 - vite.svg
- src
 - assets
 - vue.svg
 - components
 - HelloWorld.vue
 - App.vue
 - main.js
 - style.css
 - .gitignore
 - bun.lock
 - index.html
 - package.json
 - README.md
 - vite.config.js

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

index.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" ></meta>
5     <link rel="icon" type="image/svg+xml" href="/vite.svg" ></link>
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" ></meta>
7     <title>Vite + Vue</title> <!-- 网页标题 -->
8   </head>
9   <body>
10    <div id="app"></div> <!-- 根元素 -->
11    <script type="module" src="/src/main.js"></script> <!-- Vue 入口 -->
12  </body>
13 </html>
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

index.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" ></meta>
5     <link rel="icon" type="image/svg+xml" href="/vite.svg" ></link>
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" ></meta>
7     <title>Vite + Vue</title> <!-- 网页标题 -->
8   </head>
9   <body>
10    <div id="app"></div> <!-- 根元素 -->
11    <script type="module" src="/src/main.js"></script> <!-- Vue 入口 -->
12  </body>
13 </html>
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

main.js

```
1 import './style.css'
2 // 官方文档: 创建一个应用 - https://cn.vuejs.org/guide/essentials/application.html
3 import { createApp } from 'vue'
4 // 从一个单文件组件中导入根组件
5 import App from './App.vue'
6
7 // 创建一个应用, 并挂载应用到“容器”
8 createApp(App).mount('#app')
```

JameLouis

背景 回顾Vue 案例开发 项目部署 总结与展望 问题答疑

运行效果

```
1 # 安装依赖库
2 $ bun install
3 bun install v1.2.5 (013fdddc)
4
5 + @vitejs/plugin-vue@5.2.3
6 + vite@6.2.3
7 + vue@3.5.13
8
9 31 packages installed [333.00ms]
10 # 运行
11 $ bun run dev
12
13 VITE v6.2.3 ready in 368 ms
14
15 → Local: http://localhost:5173/evanyou-portfolio/
16 → Network: use --host to expose
17 → press h + enter to show help
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

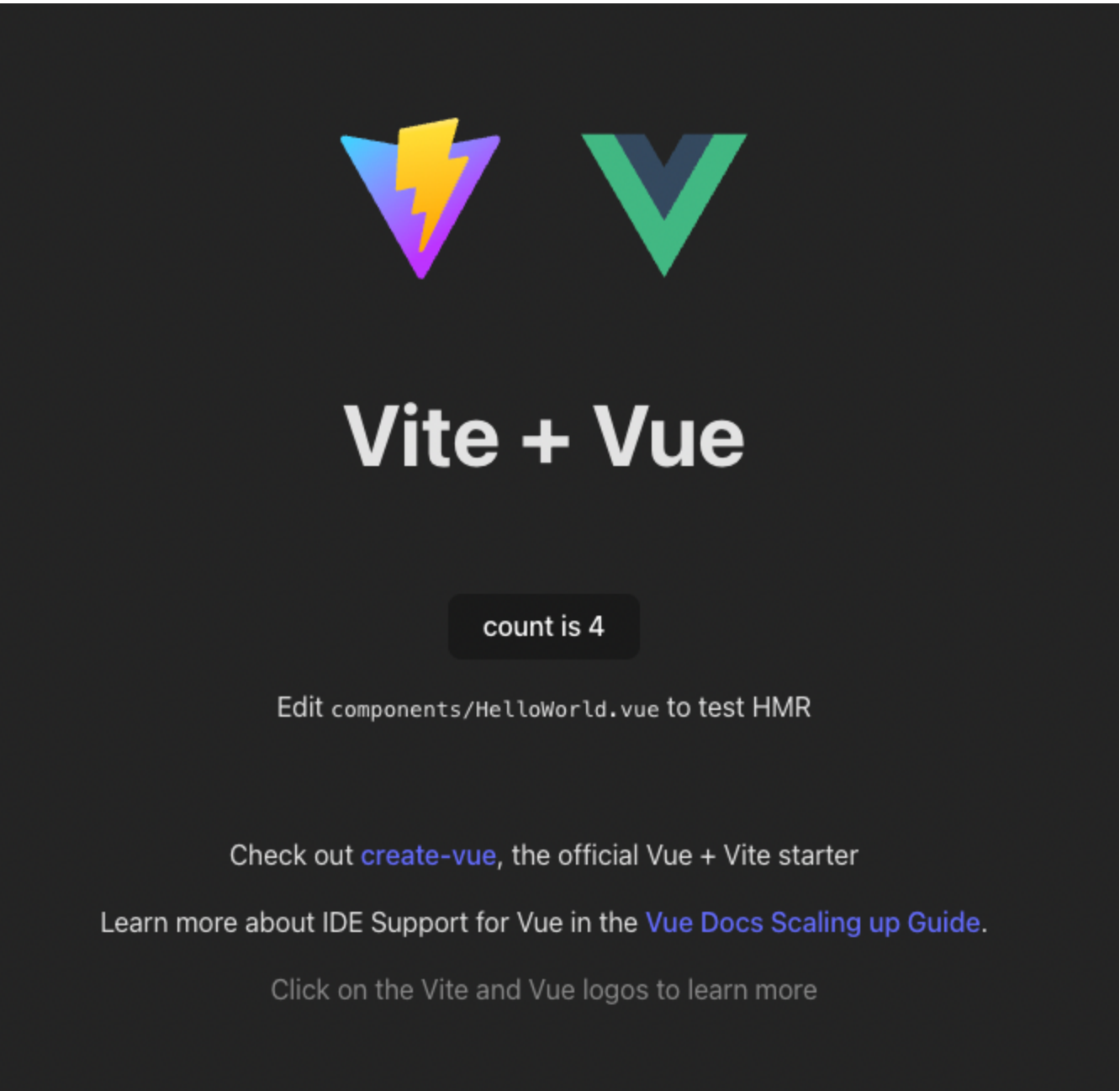
问题答疑

运行效果

```
1 # 安装依赖库
2 $ bun install
3 bun install v1.2.5 (013fdddc)
4
5 + @vitejs/plugin-vue@5.2.3
6 + vite@6.2.3
7 + vue@3.5.13
8
9 31 packages installed [333.00ms]
10 # 运行
11 $ bun run dev
12
13 VITE v6.2.3 ready in 368 ms
14
15 → Local:   http://localhost:5173/evanyou-portfolio/
16 → Network: use --host to expose
17 → press h + enter to show help
```

运行效果

```
1 # 安装依赖库
2 $ bun install
3 bun install v1.2.5 (013fdddc)
4
5 + @vitejs/plugin-vue@5.2.3
6 + vite@6.2.3
7 + vue@3.5.13
8
9 31 packages installed [333.00ms]
10 # 运行
11 $ bun run dev
12
13 VITE v6.2.3 ready in 368 ms
14
15 → Local: http://localhost:5173/evanyou-portfolio/
16 → Network: use --host to expose
17 → press h + enter to show help
```



JameLouis

背景

回顾Vue

案例开发

项目部署

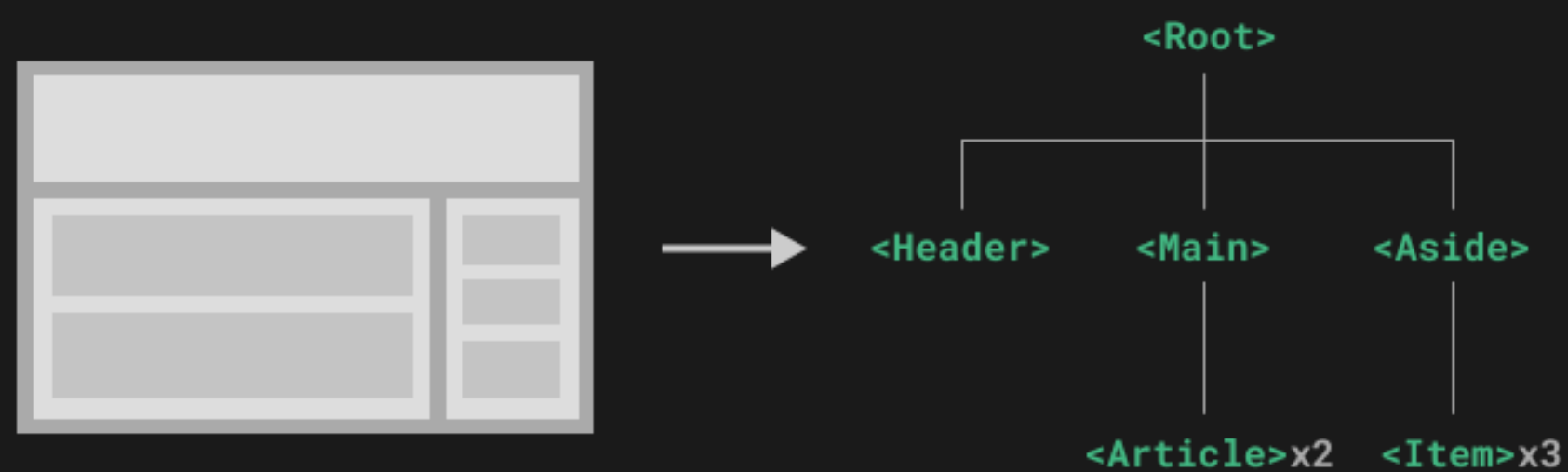
总结与展望

问题答疑

重构尤雨溪个人作品集 - Vue组件

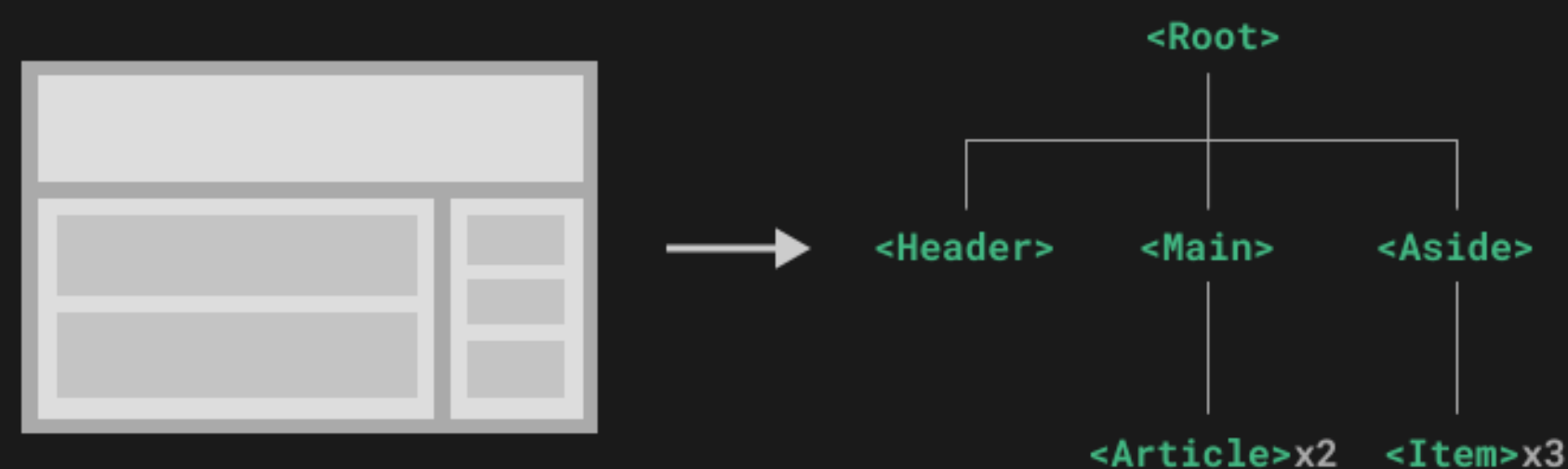
重构尤雨溪个人作品集 - Vue组件

组件允许我们将 UI 划分为独立的、可重用的部分，并且可以对每个部分进行单独的思考。在实际应用中，组件常常被组织成一个层层嵌套的树状结构：



重构尤雨溪个人作品集 - Vue组件

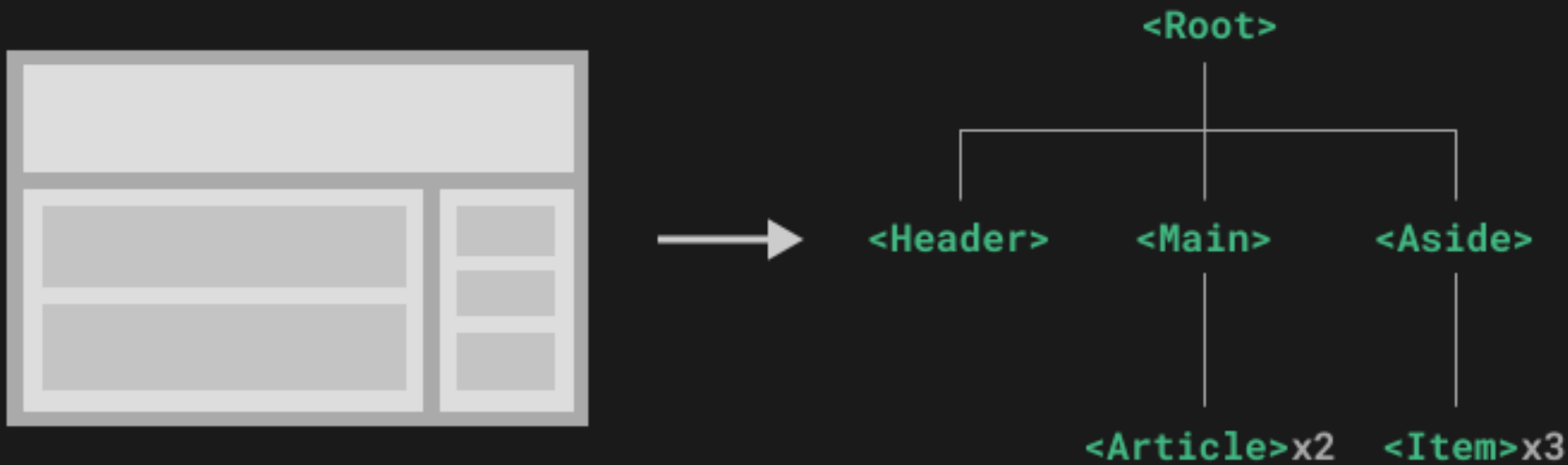
组件允许我们将 UI 划分为独立的、可重用的部分，并且可以对每个部分进行单独的思考。在实际应用中，组件常常被组织成一个层层嵌套的树状结构：



```
1 // ButtonCounter.vue
2 <script setup>
3 import { ref } from 'vue'
4 const count = ref(0)
5 </script>
6
7 <template>
8   <button @click="count++">You clicked me {{ count }} times.</button>
9 </template>
10
11 <style scoped>
12 </style>
```

重构尤雨溪个人作品集 - Vue组件

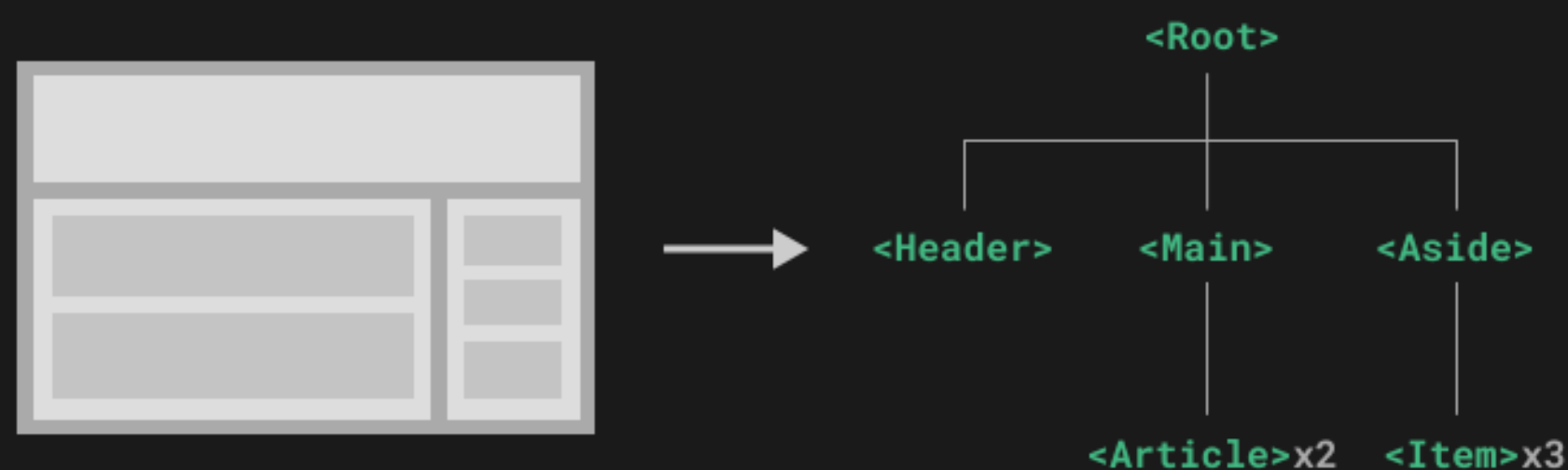
组件允许我们将 UI 划分为独立的、可重用的部分，并且可以对每个部分进行单独的思考。在实际应用中，组件常常被组织成一个层层嵌套的树状结构：



```
1 // ButtonCounter.vue
2 <script setup>
3 import { ref } from 'vue'
4 const count = ref(0)
5 </script>
6
7 <template>
8   <button @click="count++">You clicked me {{ count }} times.</button>
9 </template>
10
11 <style scoped>
12 </style>
```

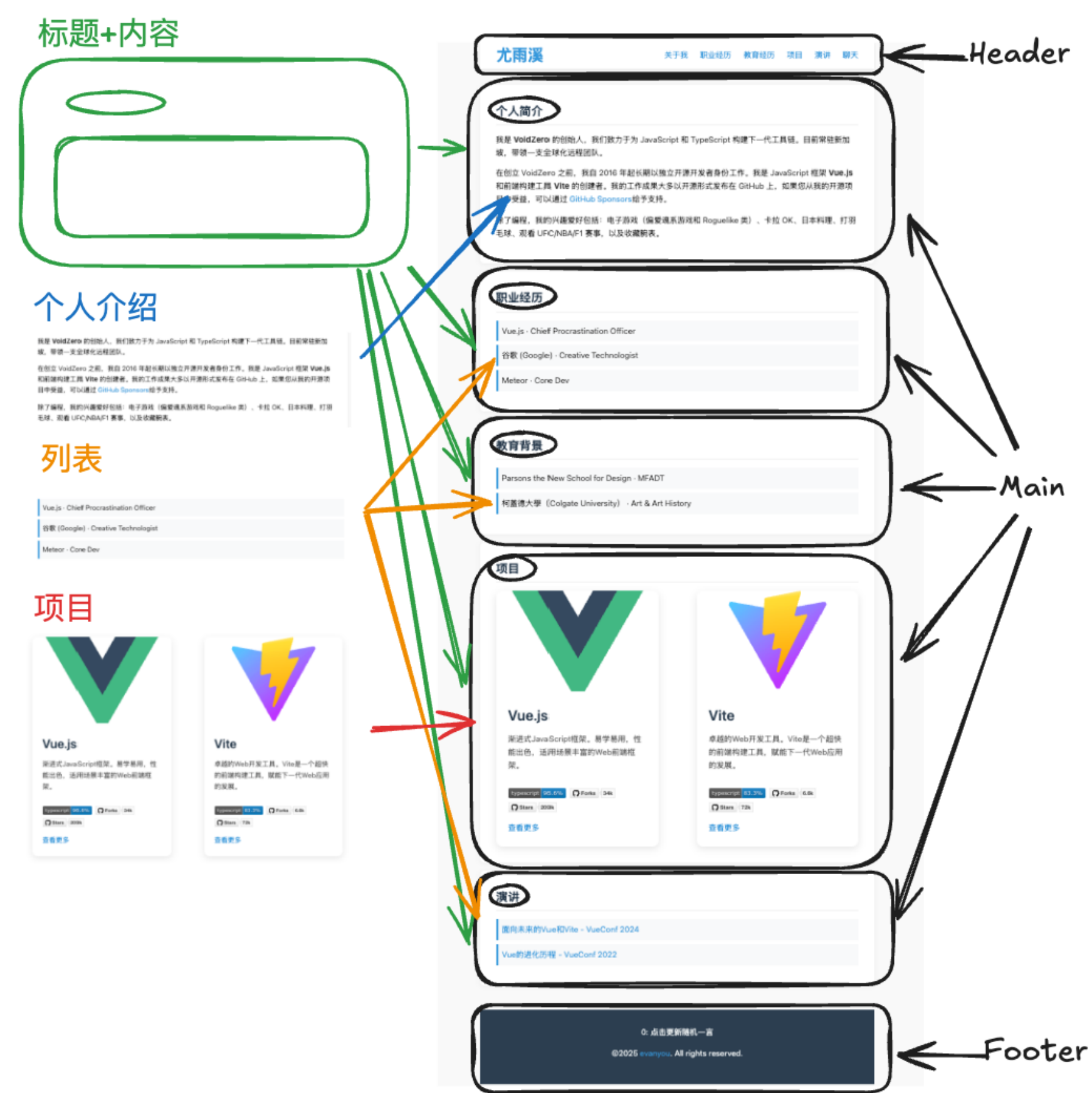
重构尤雨溪个人作品集 - Vue组件

组件允许我们将 UI 划分为独立的、可重用的部分，并且可以对每个部分进行单独的思考。在实际应用中，组件常常被组织成一个层层嵌套的树状结构：



```
1 // ButtonCounter.vue
2 <script setup>
3 import { ref } from 'vue'
4 const count = ref(0)
5 </script>
6
7 <template>
8   <button @click="count++">You clicked me {{ count }} times.</button>
9 </template>
10
11 <style scoped>
12 </style>
```

布局 and 组件



JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

清理项目代码和文件

删除HelloWorld.vue组件文件

App.vue

- 清空template里的内容
- 清空style里的内容

清空style.css里的内容

Header组件

```
<script setup>
</script>

<template>
  <header>
    <h1>尤雨溪</h1>
    <nav>
      <a href="#about">关于我</a>
      <a href="#work">职业经历</a>
      <a href="#education">教育经历</a>
      <a href="#project">项目</a>
      <a href="#speak">演讲</a>
    </nav>
  </header>
</template>

<style scoped>
header {
  background-color: #fff;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  padding: 0rem 2rem;
  position: sticky;
  top: 0;
  z-index: 100;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

Footer组件

```
<script setup>
</script>

<template>
  <footer>
    <p>
      &copy;2025
      <a href="https://evanyou.me/">evanyou</a>. All rights reserved.
    </p>
  </footer>
</template>

<style scoped>
footer {
  text-align: center;
  padding: 2rem;
  background-color: #2c3e50;
  color: #ecf0f1;
  margin-top: 3rem;
}
</style>
```

TextContent组件

```
<script setup>
import { ref } from 'vue'

defineProps({
  id: String,
  title: String
})
</script>

<template>
  <article :id="id">
    <h2>{{ title }}</h2>
    <slot></slot>
  </article>
</template>

<style scoped>

article {
  margin-bottom: 1rem;
  background-color: #fff;
  padding: 2rem;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);
}

article h2 {
```

Intro组件

```
<template>
  <p>
    我是
    <strong>VoidZero</strong> 的创始人，我们致力于为 JavaScript 和 TypeScript 构建下一代工具链。目前常驻新加坡，带领一支全球化远程团队。
  </p>
  <p>
    在创立 voidzero 之前，我自 2016 年起长期以独立开源开发者身份工作。我是 JavaScript 框架
    <strong>Vue.js</strong> 和前端构建工具
    <strong>Vite</strong> 的创建者。我的工作成果大多以开源形式发布在 GitHub 上，如果您从我的开源项目中受益，可以通过
    <a href="https://github.com/sponsors/yyx990803">GitHub Sponsors</a>给予支持。
  </p>
  <p>除了编程，我的兴趣爱好包括：电子游戏（偏爱魂系游戏和 Roguelike 类）、卡拉 OK、日本料理、打羽毛球、观看 UFC/NBA/F1 赛事，以及收藏腕表。</p>
</template>
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

工作、教育、演讲组件

```
<template>
  <ul>
    <li>Vue.js · Chief Procrastination Officer</li>
    <li>谷歌 (Google) · Creative Technologist</li>
    <li>Meteor · Core Dev</li>
  </ul>
</template>
```

项目组件

```
<template>
  <div class="project">
    <section>
      <div class="card">
        <div class="card-image">
          </img>
        </div>
        <div class="card-content">
          <h2 class="card-title">Vue.js</h2>
          <p class="card-text">渐进式JavaScript框架。易学易用，性能出色，适用场景丰富的Web前端框架。</p>
          <div class="card-meta">
            </img>
            </img>
            </img>
          </div>
        </div>
      </div>
    </section>
  </div>
```

添加CSS样式

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: system-ui;
  line-height: 1.6;
  color: #333;
  background-color: #f9f9f9;
  max-width: 780px;
  margin: 0 auto;
}

a {
  text-decoration: none;
  color: #3498db;
  transition: color 0.3s ease;
}

a:hover {
  color: #2980b9;
}

p {
  padding-bottom: 1rem;
}
```

Vue组件下的效果

尤雨溪

关于我 职业经历 教育经历 项目 演讲

Meteor · Core Dev

教育背景

Parsons the New School for Design · MFADT

柯盖德大學 (Colgate University) · Art & Art History

项目



Vue.js

渐进式JavaScript框架。易学易用，性能出色，适用场景丰富的Web前端框架。

typescript 96.6%

Forks 34k

Stars 209k

查看更多



Vite

卓越的Web开发工具。Vite是一个超快的前端构建工具，赋能下一代Web应用的发展。

typescript 83.4%

Forks 6.5k

Stars 72k

查看更多

演讲

重构项目组件 - 响应式 + v-for + 组件通信 + v-bind

```
1 <script setup>
2 import Card from './Card.vue';
3 import { reactive } from 'vue';
4 const projects = reactive([
5   {
6     "id": 0,
7     "title": "Vue.js",
8     "description": "渐进式JavaScript框架。易学易用，性能出色，适
9     "github_language": "https://img.shields.io/github/lang
10    "github_forks": "https://img.shields.io/github/forks/v
11    "github_stars": "https://img.shields.io/github/stars/v
12    "cover": "data:image/svg+xml,%3csvg%20xmlns='http://ww
13  },
14  {
15    "id": 1,
16    "title": "Vite",
17    "description": "卓越的Web开发工具。Vite是一个超快的前端构建工
18    "github_language": "https://img.shields.io/github/lang
19    "github_forks": "https://img.shields.io/github/forks/v
20    "github_stars": "https://img.shields.io/github/stars/v
21    "cover": "https://cn.vite.dev/logo.svg"
22  }
23 ]);
24 </script>
25
26 <template>
27   <div class="project">
```

```
1 <script setup>
2 import { ref } from 'vue'
3
4 defineProps({
5   title: String,
6   description: String,
7   github_language: String,
8   github_forks: String,
9   github_stars: String,
10  cover: String
11 })
12 </script>
13
14 <template>
15   <div class="card">
16     <div class="card-image">
17       </img>
21     </div>
22     <div class="card-content">
23       <h2 class="card-title">{{ title }}</h2>
24       <p class="card-text">{{ description }}</p>
25       <div class="card-meta">
26         
25
26 <template>
27   <div class="project">
28     <section v-for="project in projects" :key="project.id"
29       <Card
30         :title="project.title"
31         :description="project.description"
32         :github_language="project.github_language"
33         :github_forks="project.github_forks"
34         :github_stars="project.github_stars"
35         :cover="project.cover"
36       ></Card>
37     </section>
38   </div>
39 </template>
40
41
42 <style scoped>
43 .project {
44   display: flex;
45   justify-content: space-between;
46 </div class="project">
```

```
1 <script setup>
2 import { ref } from 'vue'
3
4 defineProps({
5   title: String,
6   description: String,
7   github_language: String,
8   github_forks: String,
9   github_stars: String,
10  cover: String
11 })
12 </script>
13
14 <template>
15   <div class="card">
16     <div class="card-image">
17       </img>
21     </div>
22     <div class="card-content">
23       <h2 class="card-title">{{ title }}</h2>
24       <p class="card-text">{{ description }}</p>
25       <div class="card-meta">
26         
2 import Card from "./Card.vue";
3 import { reactive } from "vue";
4 const projects = reactive([
5   {
6     "id": 0,
7     "title": "Vue.js",
8     "description": "渐进式JavaScript框架。易学易用，性能出色，适
9     "github_language": "https://img.shields.io/github/lang
10    "github_forks": "https://img.shields.io/github/forks/v
11    "github_stars": "https://img.shields.io/github/stars/v
12    "cover": "data:image/svg+xml,%3csvg%20xmlns='http://ww
13  },
14  {
15    "id": 1,
16    "title": "Vite",
17    "description": "卓越的Web开发工具。vite是一个超快的前端构建工
18    "github_language": "https://img.shields.io/github/lang
19    "github_forks": "https://img.shields.io/github/forks/v
20    "github_stars": "https://img.shields.io/github/stars/v
21    "cover": "https://cn.vite.dev/logo.svg"
22  }
23 ]);
24 </script>
25
26 <template>
27   <div class="project">
```

```
1 <script setup>
2 import { ref } from 'vue'
3
4 defineProps({
5   title: String,
6   description: String,
7   github_language: String,
8   github_forks: String,
9   github_stars: String,
10  cover: String
11 })
12 </script>
13
14 <template>
15   <div class="card">
16     <div class="card-image">
17       </img>
21     </div>
22     <div class="card-content">
23       <h2 class="card-title">{{ title }}</h2>
24       <p class="card-text">{{ description }}</p>
25       <div class="card-meta">
26         
25
26 <template>
27   <div class="project">
28     <section v-for="project in projects" :key="project.id">
29       <Card
30         :title="project.title"
31         :description="project.description"
32         :github_language="project.github_language"
33         :github_forks="project.github_forks"
34         :github_stars="project.github_stars"
35         :cover="project.cover"
36       ></Card>
37     </section>
38   </div>
39 </template>
40
41
42 <style scoped>
43 .project {
44   display: flex;
45   justify-content: space-between;
46 </div class="project">
```

```
1 <script setup>
2 import { ref } from 'vue'
3
4 defineProps({
5   title: String,
6   description: String,
7   github_language: String,
8   github_forks: String,
9   github_stars: String,
10  cover: String
11 })
12 </script>
13
14 <template>
15   <div class="card">
16     <div class="card-image">
17       </img>
21     </div>
22     <div class="card-content">
23       <h2 class="card-title">{{ title }}</h2>
24       <p class="card-text">{{ description }}</p>
25       <div class="card-meta">
26         
25
26 <template>
27   <div class="project">
28     <section v-for="project in projects" :key="project.id">
29       <Card
30         :title="project.title"
31         :description="project.description"
32         :github_language="project.github_language"
33         :github_forks="project.github_forks"
34         :github_stars="project.github_stars"
35         :cover="project.cover"
36       ></Card>
37     </section>
38   </div>
39 </template>
40
41
42 <style scoped>
43 .project {
44   display: flex;
45   justify-content: space-between;
46 </div class="project">
```

```
1 <script setup>
2 import { ref } from 'vue'
3
4 defineProps({
5   title: String,
6   description: String,
7   github_language: String,
8   github_forks: String,
9   github_stars: String,
10  cover: String
11 })
12 </script>
13
14 <template>
15   <div class="card">
16     <div class="card-image">
17       </img>
21     </div>
22     <div class="card-content">
23       <h2 class="card-title">{{ title }}</h2>
24       <p class="card-text">{{ description }}</p>
25       <div class="card-meta">
26         
3 import { ref, watch, computed } from 'vue';
4 const count = ref(0);
5 const quote = ref("点击更新随机一言");
6
7 watch(count, async (newCount) => {
8   try {
9     const res = await fetch("https://uapis.cn/api/say")
10     quote.value = await (res.text());
11   } catch (error) {
12   } finally {
13   }
14 })
15
16 const msg = computed(()=>{
17   return count.value + ": " + quote.value;
18 })
19 </script>
20
21 <template>
22   <footer>
23     <p class="clickable" @click="count++">
24       {{ msg }}
25     </p>
26     <p>
27       2025
```

新特性：随机一言 - watch + @click + rest api + computed

```
11     } catch (error) {
12     } finally {
13     }
14 })
15
16 const msg = computed(()=>{
17     return count.value + ": " + quote.value;
18 })
19 </script>
20
21 <template>
22     <footer>
23         <p class="clickable" @click="count++">
24             {{ msg }}
25         </p>
26         <p>
27             &copy;2025
28             <a href="https://evanyou.me/">evanyou</a>. All rights res
29         </p>
30     </footer>
31 </template>
32
33 <style scoped>
34 footer {
35     text-align: center;
36     padding: 2rem;
37     background-color: #2c3e50;
```

新特性：随机一言 - watch + @click + rest api + computed

```
1 // Footer.vue
2 <script setup>
3 import { ref, watch, computed } from 'vue';
4 const count = ref(0);
5 const quote = ref("点击更新随机一言");
6
7 watch(count, async (newCount) => {
8   try {
9     const res = await fetch("https://uapis.cn/api/say")
10     quote.value = await (res.text());
11   } catch (error) {
12   } finally {
13   }
14 })
15
16 const msg = computed(()=>{
17   return count.value + ": " + quote.value;
18 })
19 </script>
20
21 <template>
22   <footer>
23     <p class="clickable" @click="count++">
24       {{ msg }}
25     </p>
26     <p>
27       2025
```

新特性：随机一言 - watch + @click + rest api + computed

```
4 const count = ref(0);
5 const quote = ref("点击更新随机一言");
6
7 watch(count, async (newCount) => {
8   try {
9     const res = await fetch("https://uapis.cn/api/say")
10    quote.value = await (res.text());
11   } catch (error) {
12   } finally {
13   }
14 })
15
16 const msg = computed(()=>{
17   return count.value + ": " + quote.value;
18 })
19 </script>
20
21 <template>
22   <footer>
23     <p class="clickable" @click="count++">
24       {{ msg }}
25     </p>
26     <p>
27       &copy;2025
28       <a href="https://evanyou.me/">evanyou</a>. All rights res
29     </p>
30   </footer>
31 </template>
```

新特性：随机一言 - watch + @click + rest api + computed

```
1 // Footer.vue
2 <script setup>
3 import { ref, watch, computed } from 'vue';
4 const count = ref(0);
5 const quote = ref("点击更新随机一言");
6
7 watch(count, async (newCount) => {
8   try {
9     const res = await fetch("https://uapis.cn/api/say")
10    quote.value = await (res.text());
11   } catch (error) {
12   } finally {
13   }
14 })
15
16 const msg = computed(()=>{
17   return count.value + ": " + quote.value;
18 })
19 </script>
20
21 <template>
22   <footer>
23     <p class="clickable" @click="count++">
24       {{ msg }}
25     </p>
26   </p>
27   </template>
```

0: 点击更新随机一言

©2025 [evanyou](#). All rights reserved.

1: 对! 错的不是我, 是世界!

©2025 [evanyou](#). All rights reserved.

2: 要享受有趣的事情, 首先就是不能违背自己的感情。

©2025 [evanyou](#). All rights reserved.

JameLouis

背景 回顾Vue 案例开发 项目部署 总结与展望 问题答疑

Vue Router + 聊天机器人

Vue Router安装

使用Vue Router实现主页和聊天机器人页面

聊天机器人 - :class + v-model

Vue Router安装

```
$ bun install vue-router@4
bun add v1.2.5 (013fdddc)

installed vue-router@4.5.0

2 packages installed [2.18s]
```



JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

使用Vue Router实现主页和聊天机器人页面

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.'
4 import Footer from './components/Footer.'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView ></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.'
4 import Footer from './components/Footer.'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView ></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue';
import Work from './Work.vue';
import Education from './Education.vue';
import Project from './Project.vue';
import Speak from './Speak.vue';
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>

```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView ></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>
```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
1 import { createMemoryHistory, createRouter } from 'vue-router'
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7  routes 选项定义了一组路由，把URL路径映射到组件
8  */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes
17 })
18
```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>
```

```
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7  routes 选项定义了一组路由，把URL路径映射到组件
8  */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView },
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes,
17 })
```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView /></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
5
6 /*
7 routes 选项定义了一组路由，把URL路径映射到组件
8 */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes,
17 })
18
19 export default router

```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <!-- 教育经历 -->
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <!-- 项目经历 -->
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <!-- 演讲经历 -->
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
1 import { createMemoryHistory, createRouter } from 'vue-router'
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7 routes 选项定义了一组路由，把URL路径映射到组件
8 */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes
17 })

```

```
1 <script setup>
2 </script>
3
4 <template>
5   <header>
6     <h1><RouterLink to="/">尤雨溪</RouterLink></h1>
7     <nav>
8       <a href="#about">关于我</a>
9       <a href="#work">职业经历</a>
10      <a href="#education">教育经历</a>
11      <a href="#project">项目</a>
12      <a href="#speak">演讲</a>
13      <RouterLink to="/chat">聊天</RouterLink>
14    </nav>
15  </header>
16 </template>

```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <!-- 教育经历 -->
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <!-- 项目经历 -->
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <!-- 演讲经历 -->
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
1 import { createMemoryHistory, createRouter } from 'vue-router'
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7 routes 选项定义了一组路由，把URL路径映射到组件
8 */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes
17 })
18
```

```
6 <h1><RouterLink to="/">尤雨溪</RouterLink></h1>
7 <nav>
8   <a href="#about">关于我</a>
9   <a href="#work">职业经历</a>
10  <a href="#education">教育经历</a>
11  <a href="#project">项目</a>
12  <a href="#speak">演讲</a>
13  <RouterLink to="/chat">聊天</RouterLink>
14 </nav>
15 </header>
16 </template>
17
18 <style scoped>
19 header {
20   background-color: #fff;
21 }
22
```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <!-- 教育经历 -->
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <!-- 项目经历 -->
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <!-- 演讲经历 -->
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
1 import { createMemoryHistory, createRouter } from 'vue-router'
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7 routes 选项定义了一组路由，把URL路径映射到组件
8 */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes
17 })

```

```
1 <script setup>
2 </script>
3
4 <template>
5   <header>
6     <h1><RouterLink to="/">尤雨溪</RouterLink></h1>
7   <nav>
8     <a href="#about">关于我</a>
9     <a href="#work">职业经历</a>
10    <a href="#education">教育经历</a>
11    <a href="#project">项目</a>
12    <a href="#speak">演讲</a>
13    <RouterLink to="/chat">聊天</RouterLink>
14  </nav>
15 </header>
16 </template>

```

```
1 // main.js
2 import './style.css'
3 import { createApp } from 'vue'
4 import router from './router'
5 import App from './App.vue'
6
7 createApp(App)
8   .use(router)
9   .mount('#app')

```

使用Vue Router实现主页和聊天机器人页面

```
1 // App.vue
2 <script setup>
3 import Header from './components/Header.vue'
4 import Footer from './components/Footer.vue'
5 </script>
6
7 <template>
8   <Header>
9   </Header>
10  <main>
11    <RouterView></RouterView>
12  </main>
13  <Footer></Footer>
14 </template>
15
```

```
// HomeView.vue
<script setup>
import TextContent from './TextContent.vue'
import Intro from './Intro.vue'
import Work from './Work.vue'
import Education from './Education.vue'
import Project from './Project.vue'
import Speak from './Speak.vue'
</script>
<template>
  <!-- 个人简介 -->
  <TextContent id="about" title="个人简介">
    <Intro></Intro>
  </TextContent>
  <!-- 职业经历 -->
  <TextContent id="work" title="职业经历">
    <Work></Work>
  </TextContent>
  <!-- 教育经历 -->
  <TextContent id="education" title="教育经历">
    <Education></Education>
  </TextContent>
  <!-- 项目经历 -->
  <TextContent id="project" title="项目经历">
    <Project></Project>
  </TextContent>
  <!-- 演讲经历 -->
  <TextContent id="speak" title="演讲经历">
    <Speak></Speak>
  </TextContent>
</template>

```

```
// ChatView.vue
<template>
  <h2>你好，我是聊天机器人</h2>
</template>

```

```
1 import { createMemoryHistory, createRouter } from 'vue-router'
2
3 import HomeView from './components/HomeView.vue'
4 import ChatView from './components/ChatView.vue'
5
6 /*
7 routes 选项定义了一组路由，把URL路径映射到组件
8 */
9 const routes = [
10   { path: '/', component: HomeView },
11   { path: '/chat', component: ChatView }
12 ]
13
14 const router = createRouter({
15   history: createMemoryHistory(),
16   routes
17 })

```

```
1 <script setup>
2 </script>
3
4 <template>
5   <header>
6     <h1><RouterLink to="/">尤雨溪</RouterLink></h1>
7   <nav>
8     <a href="#about">关于我</a>
9     <a href="#work">职业经历</a>
10    <a href="#education">教育经历</a>
11    <a href="#project">项目</a>
12    <a href="#speak">演讲</a>
13    <RouterLink to="/chat">聊天</RouterLink>
14  </nav>
15 </header>
16 </template>

```

```
1 // main.js
2 import './style.css'
3 import { createApp } from 'vue'
4 import router from './router'
5 import App from './App.vue'
6
7 createApp(App)
8   .use(router)
9   .mount('#app')

```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

聊天机器人页面 - :class + v-if + v-model

聊天机器人页面 - :class + v-if + v-model

```
1 <template>
2   <div class="chat-container">
3     <div class="chat-header">
4       <h2>Vue & Vite Chat</h2>
5     </div>
6     <div class="chat-messages">
7       <div
8         v-for="(message, index) in conversationHistory"
9         :key="index"
10        :class="['message', message.role === 'user' ? 'user' : 'system']"
11      >
12        <div v-if="message.role !== 'system'" class="message">
13          </div>
14        </div>
15      <div class="chat-input">
16        <label>API Key: </label>
17        <input
18          type="password"
19          v-model="apiKey"
20          placeholder="API Key"
21        ></input>
22      </div>
23      <div class="chat-input">
24        <input
25          v-model="userInput"
26          placeholder="输入你的问题..."
27        ></input>
28      </div>
29    </div>
30  </div>
31</template>
```

聊天机器人页面 - :class + v-if + v-model

```
1 <template>
2   <div class="chat-container">
3     <div class="chat-header">
4       <h2>Vue & Vite Chat</h2>
5     </div>
6     <div class="chat-messages">
7       <div
8         v-for="(message, index) in conversationHistory"
9         :key="index"
10        :class="['message', message.role === 'user' ? 'user' : 'system']"
11      >
12        <div v-if="message.role !== 'system'" class="message">
13          </div>
14        </div>
15      <div class="chat-input">
16        <label>API Key: </label>
17        <input
18          type="password"
19          v-model="apiKey"
20          placeholder="API Key"
21        ></input>
22      </div>
23      <div class="chat-input">
24        <input
25          v-model="userInput"
26          placeholder="输入你的问题..."
27        ></input>
28      </div>
29    </div>
30  </div>
31</template>
```

聊天机器人页面 - :class + v-if + v-model

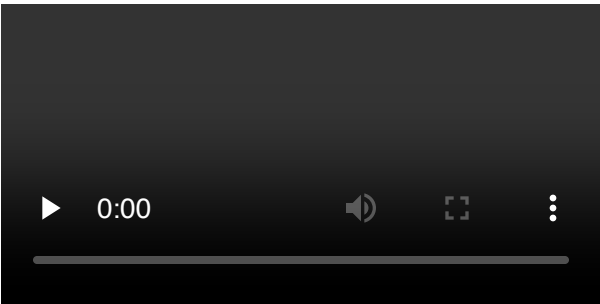
```
6      <div class="chat-messages">
7        <div
8          v-for="(message, index) in conversationHistory"
9          :key="index"
10         :class="['message', message.role === 'user' ? 'user' : 'system']"
11        >
12          <div v-if="message.role !== 'system'" class="message">
13            </div>
14          </div>
15          <div class="chat-input">
16            <label>API Key: </label>
17            <input
18              type="password"
19              v-model="apiKey"
20              placeholder="API Key"
21            ></input>
22          </div>
23          <div class="chat-input">
24            <input
25              v-model="userInput"
26              placeholder="输入你的问题..."
27              @keypress.enter="sendMessage"
28            ></input>
29            <button @click="sendMessage">发送</button>
30          </div>
31        </div>
32      </template>
33      @keypress.enter="sendMessage"
```

聊天机器人页面 - :class + v-if + v-model

```
12     <div v-if= message.role !== 'system' class= message
13     </div>
14 </div>
15 <div class="chat-input">
16   <label>API Key: </label>
17   <input
18     type="password"
19     v-model="apiKey"
20     placeholder="API Key"
21   ></input>
22 </div>
23 <div class="chat-input">
24   <input
25     v-model="userInput"
26     placeholder="输入你的问题..."
27     @keypress.enter="sendMessage"
28   ></input>
29   <button @click="sendMessage">发送</button>
30 </div>
31 </div>
32 </template>
33
34 <script setup>
35 import { ref } from 'vue';
36 import axios from 'axios';
37
38 const systemMessage = {
39   @keypress.enter="sendMessage"
```

聊天机器人页面 - :class + v-if + v-model

```
1 <template>
2   <div class="chat-container">
3     <div class="chat-header">
4       <h2>Vue & Vite Chat</h2>
5     </div>
6     <div class="chat-messages">
7       <div
8         v-for="(message, index) in conversationHistory"
9         :key="index"
10        :class="['message', message.role === 'user' ? 'user' : 'system']"
11      >
12        <div v-if="message.role !== 'system'" class="message">
13          </div>
14        </div>
15      <div class="chat-input">
16        <label>API Key: </label>
17        <input
18          type="password"
19          v-model="apiKey"
20          placeholder="API Key"
21        ></input>
22      </div>
23      <div class="chat-input">
24        <input
25          v-model="userInput"
26          placeholder="输入你的问题..."
27          @keyup.enter="sendMessage">
```



JameLouis

背景 回顾Vue 案例开发 项目部署 总结与展望 问题答疑

项目部署

构建应用

本地测试

部署到Github Pages

构建应用

```
1 $ bun run build
2 $ vite build
3 vite v6.2.3 building for production...
4 ✓ 25 modules transformed.
5 dist/index.html          0.51 kB | gzip:  0.32 kB
6 dist/assets/index-Xlrkv5kA.css  1.98 kB | gzip:  0.78 kB
7 dist/assets/index-DmeE7fDD.js 65.25 kB | gzip: 26.46 kB
8 ✓ built in 253ms
```

构建应用

```
1 $ bun run build
2 $ vite build
3 vite v6.2.3 building for production...
4 ✓ 25 modules transformed.
5 dist/index.html          0.51 kB | gzip:  0.32 kB
6 dist/assets/index-Xlrkv5kA.css  1.98 kB | gzip:  0.78 kB
7 dist/assets/index-DmeE7fDD.js 65.25 kB | gzip: 26.46 kB
8 ✓ built in 253ms
```

构建应用

```
1 $ bun run build
2 $ vite build
3 vite v6.2.3 building for production...
4 ✓ 25 modules transformed.
5 dist/index.html          0.51 kB | gzip: 0.32 kB
6 dist/assets/index-Xlrkv5kA.css 1.98 kB | gzip: 0.78 kB
7 dist/assets/index-DmeE7fDD.js 65.25 kB | gzip: 26.46 kB
8 ✓ built in 253ms
```

构建应用

```
1 $ bun run build
2 $ vite build
3 vite v6.2.3 building for production...
4 ✓ 25 modules transformed.
5 dist/index.html          0.51 kB | gzip:  0.32 kB
6 dist/assets/index-Xlrkv5kA.css  1.98 kB | gzip:  0.78 kB
7 dist/assets/index-DmeE7fDD.js  65.25 kB | gzip: 26.46 kB
8 ✓ built in 253ms
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

本地测试

```
1 $ bun run preview
2 $ vite preview
3   → Local:   http://localhost:4173/evanyou-portfolio/
4   → Network: use --host to expose
5   → press h + enter to show help
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

本地测试

```
1 $ bun run preview
2 $ vite preview
3   → Local:    http://localhost:4173/evanyou-portfolio/
4   → Network: use --host to expose
5   → press h + enter to show help
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

本地测试

```
1 $ bun run preview
2 $ vite preview
3   → Local:    http://localhost:4173/evanyou-portfolio/
4   → Network: use --host to expose
5   → press h + enter to show help
```

部署到Github Pages

```
1 # 将静态内容部署到 GitHub Pages 的简易工作流程
2 name: Deploy Vite + Vue + Bun Site
3
4 on:
5   # 仅在推送到默认分支时运行。
6   push:
7     branches: [main]
8
9   # 这个选项可以使你手动在 Action tab 页面触发 workflow
10  workflow_dispatch:
11
12 # 设置 GITHUB_TOKEN 的权限, 以允许部署到 GitHub Pages。
13 permissions:
14   contents: write
15
16 jobs:
17   # 单次部署的工作描述
18   deploy:
19     runs-on: ubuntu-latest
20     steps:
21       - name: Checkout repository
22         uses: actions/checkout@v4
23
24       - name: Setup Bun
25         uses: oven-sh/setup-bun@v1
26         with:
27           bun-version: latest
```

部署到Github Pages

```
9   # 这个选项可以使你手动在 Action tab 页面触发 workflow
10  workflow_dispatch:
11
12  # 设置 GITHUB_TOKEN 的权限, 以允许部署到 GitHub Pages。
13  permissions:
14    contents: write
15
16  jobs:
17    # 单次部署的工作描述
18    deploy:
19      runs-on: ubuntu-latest
20      steps:
21        - name: Checkout repository
22          uses: actions/checkout@v4
23
24        - name: Setup Bun
25          uses: oven-sh/setup-bun@v1
26          with:
27            bun-version: latest
28
29        - name: Install dependencies
30          run: bun install
31
32        - name: Build project
33          run: bun run build
34
35        bun-version: latest -
```

部署到Github Pages

```
12 # 设置 GITHUB_TOKEN 的权限，允许部署到 Github Pages
13 permissions:
14   contents: write
15
16 jobs:
17   # 单次部署的工作描述
18   deploy:
19     runs-on: ubuntu-latest
20     steps:
21       - name: Checkout repository
22         uses: actions/checkout@v4
23
24       - name: Setup Bun
25         uses: oven-sh/setup-bun@v1
26         with:
27           bun-version: latest
28
29       - name: Install dependencies
30         run: bun install
31
32       - name: Build project
33         run: bun run build
34
35       - name: Deploy to GitHub Pages
36         uses: peaceiris/actions-gh-pages@v3
37         with:
38           github_token: ${ secrets.GITHUB_TOKEN }
39           bun-version: latest
```

部署到Github Pages

```
13 permissions:
14   contents: write
15
16 jobs:
17   # 单次部署的工作描述
18   deploy:
19     runs-on: ubuntu-latest
20     steps:
21       - name: Checkout repository
22         uses: actions/checkout@v4
23
24       - name: Setup Bun
25         uses: oven-sh/setup-bun@v1
26         with:
27           bun-version: latest
28
29       - name: Install dependencies
30         run: bun install
31
32       - name: Build project
33         run: bun run build
34
35       - name: Deploy to GitHub Pages
36         uses: peaceiris/actions-gh-pages@v3
37         with:
38           github_token: ${ secrets.GITHUB_TOKEN }
39           publish_dir: ./dist
40           bun-version: latest
```

部署到Github Pages

```
13 permissions:
14   contents: write
15
16 jobs:
17   # 单次部署的工作描述
18   deploy:
19     runs-on: ubuntu-latest
20     steps:
21       - name: Checkout repository
22         uses: actions/checkout@v4
23
24       - name: Setup Bun
25         uses: oven-sh/setup-bun@v1
26         with:
27           bun-version: latest
28
29       - name: Install dependencies
30         run: bun install
31
32       - name: Build project
33         run: bun run build
34
35       - name: Deploy to GitHub Pages
36         uses: peaceiris/actions-gh-pages@v3
37         with:
38           github_token: ${ secrets.GITHUB_TOKEN }
39           publish_dir: ./dist
40           bun-version: latest
```

部署到Github Pages

```
13 permissions:
14   contents: write
15
16 jobs:
17   # 单次部署的工作描述
18   deploy:
19     runs-on: ubuntu-latest
20     steps:
21       - name: Checkout repository
22         uses: actions/checkout@v4
23
24       - name: Setup Bun
25         uses: oven-sh/setup-bun@v1
26         with:
27           bun-version: latest
28
29       - name: Install dependencies
30         run: bun install
31
32       - name: Build project
33         run: bun run build
34
35       - name: Deploy to GitHub Pages
36         uses: peaceiris/actions-gh-pages@v3
37         with:
38           github_token: ${ secrets.GITHUB_TOKEN }
39           publish_dir: ./dist
40       bun-version: latest
```

JameLouis

背景

回顾Vue

案例开发

项目部署

总结与展望

问题答疑

尤雨溪作品集-Github Page

总结与展望

总结

了解 Vue + Vite + Bun技术栈搭建网站项目

基于Vue重构纯HTML+CSS的网页

掌握完整开发流程：项目初始化->组件开发->页面布局->部署

展望

重构工作、教育、演讲组件 - v-for + v-if

在Vue项目中使用tailwind css

Vue+Vite项目+Github部署项目模板

问题答疑

参考资料

Vue官网

- [创建一个Vue应用](#)
- [组件基础](#)
- [响应式基础](#)
- [列表渲染](#)
- [Vue Router](#)

Vite官网

- [部署静态站点](#)

Bun官网

[尤雨溪-个人网页](#)

[尤雨溪-知乎](#)

