

Vue3 基础培训

Vue3 培训

前端基础

- 基础 html (div 标签, 表单标签(form, input, select))
- 基础 css (width, height, margin, padding, background-color, font)
- 基础 javascript 语法 (变量定义, 定义方法, 数组操作, 字符串操作)

环境准备

安装 nodejs

统一采用 **nodejs 14.x 版本**

nodejs 下载 地址: [Previous Releases | Node.js](#)

国内下载地址: [CNPM Binaries Mirror](#)

下载完成后点击安装, 选择安装位置, 点击下一步下一步

```
# 查看版本
node -v
14.x

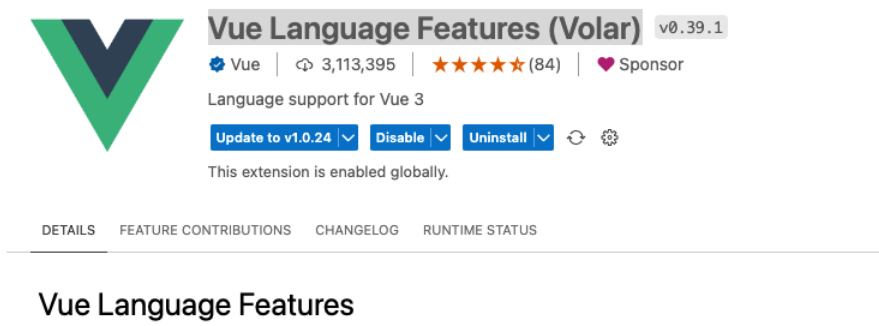
# 设置国内源
npm config set registry https://registry.npm.taobao.org

# 查看国内源
npm get registry
```

```
zwz @ zwzdeMBP in ~ [14:30:06]
$ npm get registry
https://registry.npm.taobao.org/
(base)
```

安装 vscode

安装插件: Vue Language Features (Volar)



框架搭建

vite 项目初始化

vite 是一款前端脚手架工具

vite 官网文档: [Vite | 下一代的前端工具链](#)

```
# 创建 vue3 项目
npm create vite@latest

# 输入项目名

# 选择 vuec

# cd 项目名

# npm install

# npm run dev
```

```
# zwz @ zwzdeMBP in ~ [14:30:13]
$ npm create vite@latest
07]

npx: installed 1 in 1.059s
✓ Project name: ... finedo-demo
✓ Select a framework: > Vue
✓ Select a variant: > JavaScript

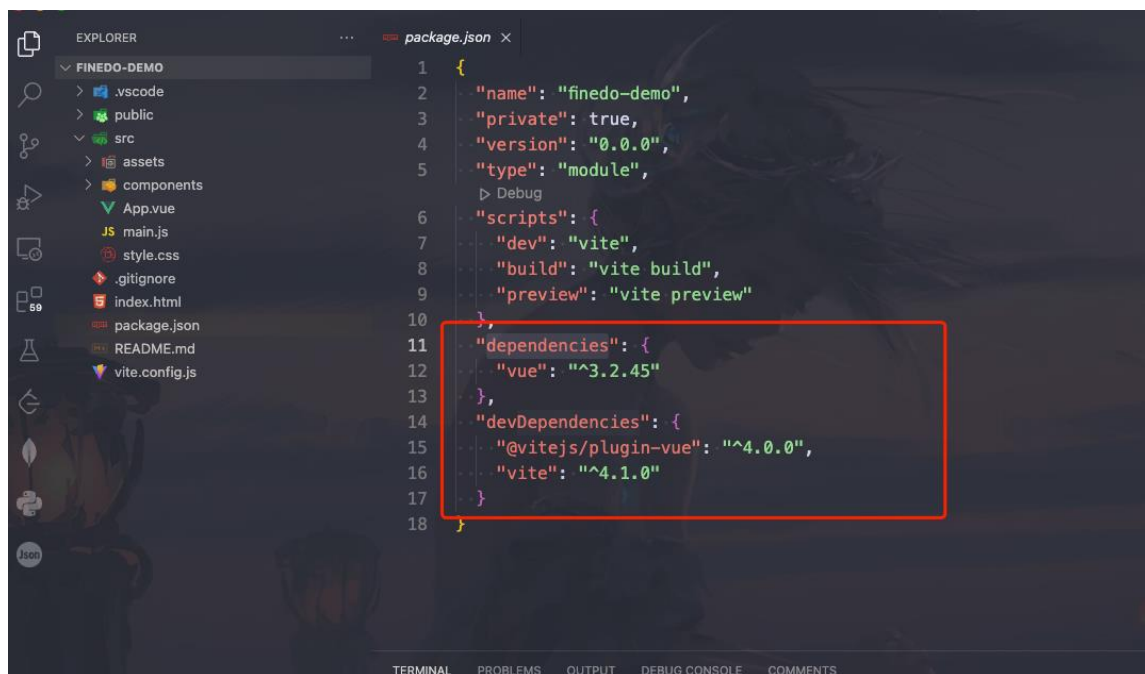
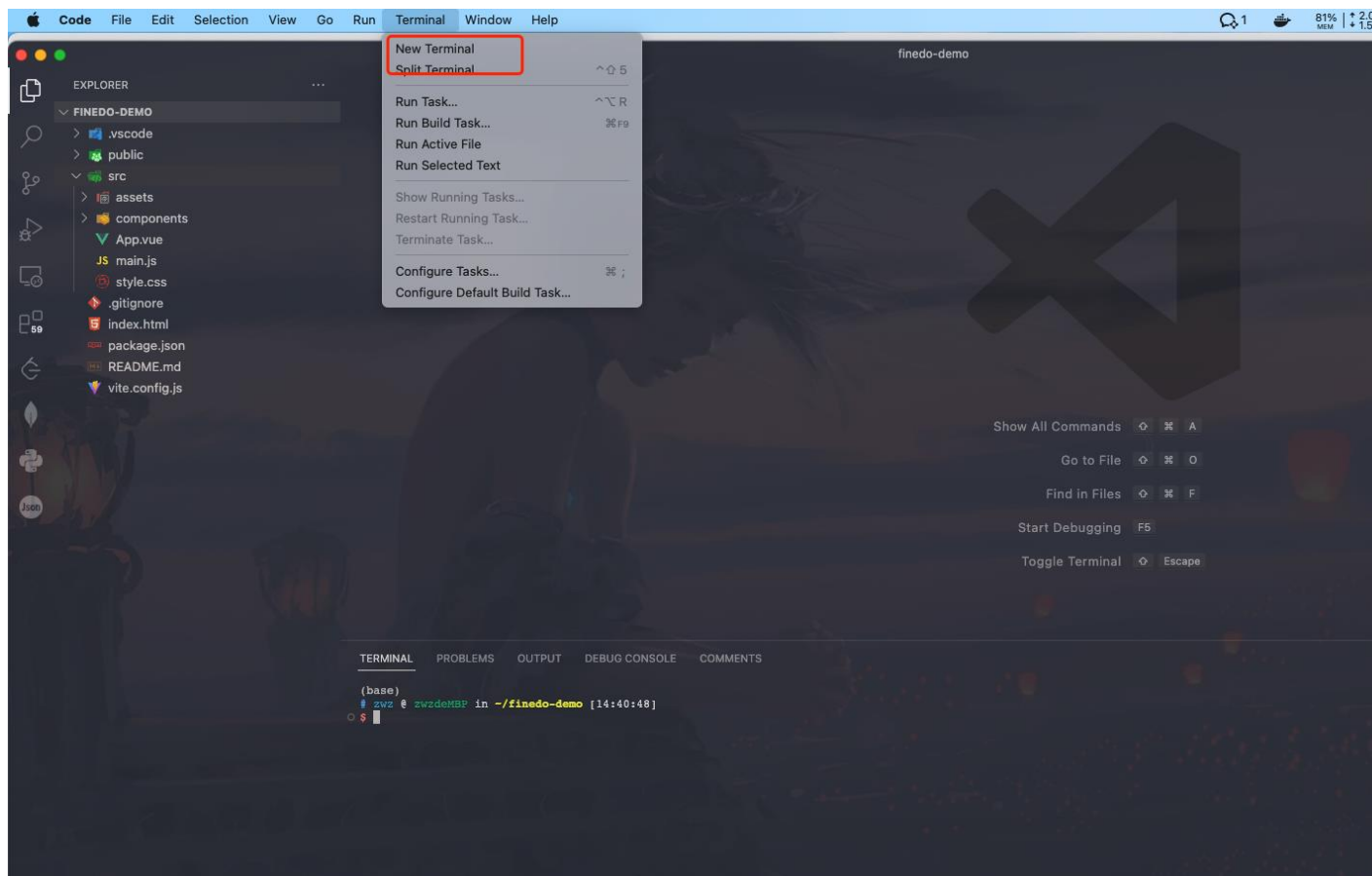
Scaffolding project in /Users/zwz/finedo-demo...

Done. Now run:

  cd finedo-demo
  npm install
  npm run dev
```

```
# zwz @ zwzdeMBP in ~/finedo-demo [14:35:57]
$ ll
total 32
-rw-r--r--  1 zwz  staff   535B Feb  7 14:35 README.md
-rw-r--r--  1 zwz  staff   357B Feb  7 14:35 index.html
-rw-r--r--  1 zwz  staff   315B Feb  7 14:35 package.json
drwxr-xr-x  3 zwz  staff    96B Feb  7 14:35 public
drwxr-xr-x  7 zwz  staff   224B Feb  7 14:35 src
-rw-r--r--  1 zwz  staff   157B Feb  7 14:35 vite.config.js
```

用 vscode 打开 finedo-demo 项目,通过 open folder 打开文件夹



在终端 terminal 中执行以下命令

安装依赖包(仅第一次启动需要执行该命令), 执行成功后会在项目根目录生成 node_modules 文

文件夹, 这个目录不要删除

```
npm i 或者 npm install
```

```
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/linux-ppc64@0.16.17: wanted {"os": "linux", "arch": "ppc64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/netbsd-x64@0.16.17 (node_modules/esbuild/node_modules/@esbuild/netbsd-x64):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/netbsd-x64@0.16.17: wanted {"os": "netbsd", "arch": "x64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/sunos-x64@0.16.17 (node_modules/esbuild/node_modules/@esbuild/sunos-x64):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/sunos-x64@0.16.17: wanted {"os": "sunos", "arch": "x64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/win32-arm64@0.16.17 (node_modules/esbuild/node_modules/@esbuild/win32-arm64):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/win32-arm64@0.16.17: wanted {"os": "win32", "arch": "arm64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/linux-ppc64@0.16.17 (node_modules/esbuild/node_modules/@esbuild/linux-ppc64):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/linux-ppc64@0.16.17: wanted {"os": "linux", "arch": "ppc64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/win32-ia32@0.16.17 (node_modules/esbuild/node_modules/@esbuild/win32-ia32):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/win32-ia32@0.16.17: wanted {"os": "win32", "arch": "ia32"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @esbuild/win32-x64@0.16.17 (node_modules/esbuild/node_modules/@esbuild/win32-x64):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @esbuild/win32-x64@0.16.17: wanted {"os": "win32", "arch": "x64"}

added 33 packages from 56 contributors in 3.006s

4 packages are looking for funding
  run `npm fund` for details

(base)
```

启动 finedo-demo 项目

```
npm run dev
```

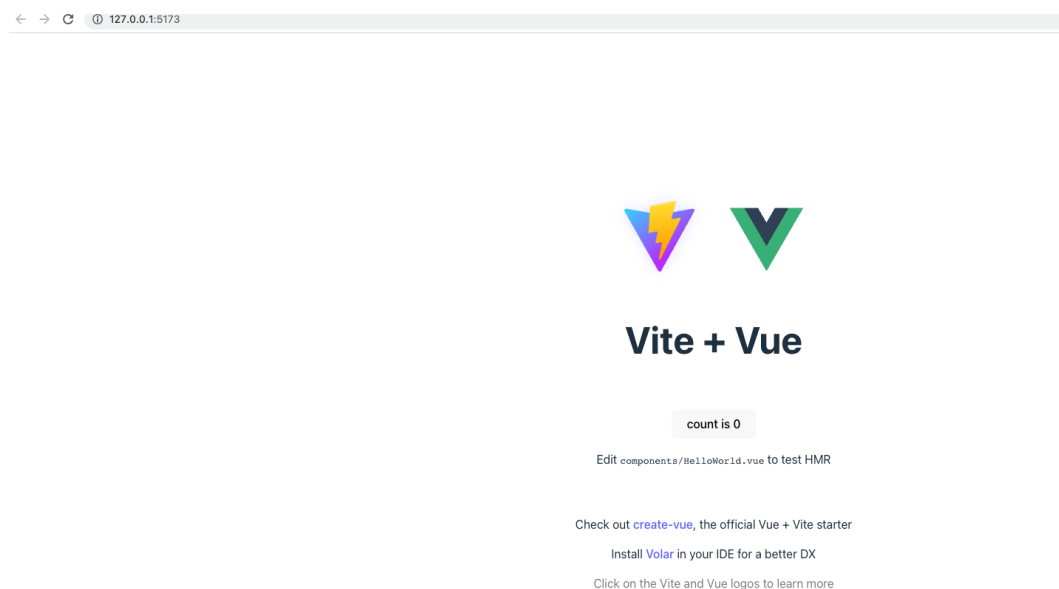
```
$ npm run dev

> finedo-demo@0.0.0 dev /Users/zwz/finedo-demo
> vite

VITE v4.1.1 ready in 701 ms

+ Local:   http://127.0.0.1:5173/
+ Network: use --host to expose
+ press h to show help
```

访问: <http://127.0.0.1:5173>



项目目录介绍

```
finedo-demo
-----node_modules 第三库依赖
-----public 公共目录
-----vite.svg 公共资源
-----src 源代码
-----assets 存放静态资源文件(图片、图标)
-----components 存放 vue 组件
-----views/pages 项目页面
-----router 路由
-----store 状态管理
-----utils 工具类
-----HelloWorld.vue 组件
App.vue vue 主文件
main.js js 入口文件
style.css 样式文件
.gitignore git 忽略文件
index.html 前端入口文件
package-lock.json
package.json 管理项目第三库依赖
README.md 项目说明文件
vite.config.js vite 配置文件(配置后端代理服务器, 部署打包优化)
```

1. Vue3 启动后访问 index.html
2. index.html 有一个 div 和一个 script
3. 加载 main.js 将 App.vue 渲染成 html 代码
4. 将渲染成 html 代码挂载在 div 节点

整合 ElementPlus 组件库

elementplus 组件地址: [一个 Vue 3 UI 框架 | Element Plus](#)

依赖安装: <https://element-plus.gitee.io/zh-CN/guide/installation.html#%E7%8E%AF%E5%A2%83%E6%94%AF%E6%8C%81>

帮助指南



基础

设计

导航

安装

快速开始

进阶

国际化

从 Element UI 升级

主题

栅格模式 (2.2.0)

自定义命名空间 (2.2.0)

内置过渡动画

更新日志

开发

安装

环境支持

Element Plus 可以在支持 [ES2018](#) 和 [ResizeObserver](#) 的浏览器上运行。如果您确实需要支持旧版本的浏览器，请自行添加 [Babel](#) 和相应的 Polyfill。

由于 Vue 3 不再支持 IE11，Element Plus 也不再支持 IE 浏览器。



Edge ≥ 79



Firefox ≥ 78



Chrome ≥ 64



Safari ≥ 12

版本

Element Plus 目前还处于快速开发迭代中。

[🔗](#)

使用包管理器

我们建议您使用包管理器（如 [NPM](#)、[Yarn](#) 或 [pnpm](#)）安装 Element Plus，然后您就可以使用打包工具，例如 [Vite](#) 或 [webpack](#)。

选择一个您喜欢的包管理器

安装

```
npm install element-plus --save
```

修改 main.js 配置文件


```
// main.ts
import { createApp } from 'vue'
import ElementPlus from 'element-plus'
// elementplus 对应的样式
import 'element-plus/dist/index.css'
import App from './App.vue'

const app = createApp(App)

app.use(ElementPlus)
app.mount('#app')
```

```

<!--
  * @Author: zhuwenzhi
  * @Date: 2023-02-07 14:35:30
  * @LastEditTime: 2023-02-07 16:48:02
  * @LastEditors: zhuwenzhi
  * @FilePath: /finedo-demo/src/App.vue
  * @Description:
-->
<script setup>
import HelloWorld from './components/HelloWorld.vue'

// 获取 HelloWorld 中 user.name 的值
function getHelloWorldName(name) {
  console.log('getHelloWorldName = ', name);
}
</script>

```

```

<template>
  <el-row class="mb-4">
    <el-button>Default</el-button>
    <el-button type="primary">Primary</el-button>
    <el-button type="success">Success</el-button>
    <el-button type="info">Info</el-button>
    <el-button type="warning">Warning</el-button>
    <el-button type="danger">Danger</el-button>
  </el-row>
<!--

```

App.vue 是一个父组件

HelloWorld 相当于一个子组件

怎么使用组件

1. script 中引入 HelloWorld 组件, `import HelloWorld from './components/HelloWorld.vue'`
2. 接着在 template 中, 增加 `<HelloWorld msg="finedo Vue" />` 标签

怎么进行组件传参?

传参数有二种

1. 父传子
2. 子传父

-->

← → ↻ ⓘ 127.0.0.1:5173

Default

Primary

Success

Info

Warning

Danger

finedo Vue

parent to child: do finedo, fiendo do

name: xiaoming -- age: 20

count is 0

count2 is 0

Edit `components/HelloWorld.vue` to test HMR

Check out [create-vue](#), the official Vue + Vite starter 1111

Install [Volar](#) in your IDE for a better DX

Click on the Vite and Vue logos to learn more

axios 接口调用

安装

```
npm i axios -S
```

使用

VueRouter 路由

官网地址: [安装 | Vue Router](#)

```

/*
 * @Author: zhuwenzhi
 * @Date: 2023-06-01 10:47:53
 * @LastEditTime: 2023-06-01 11:30:09
 * @LastEditors: zhuwenzhi
 * @FilePath: /vue3-crm-demo/src/router/index.js
 * @Description:
 *
 *
 */
import * as VueRouter from 'vue-router';

// 引入路由组件
import HelloWorld from '../components/HelloWorld.vue';
import TestUI from '../components/TestUI.vue';

// 定义 VueRouter 路由
const routes = [
  {
    name: 'hello',
    path: '/HelloWorld',
    component: HelloWorld,
  },
  { name: 'testui', path: '/TestUI', component: TestUI },
];

// 3. 创建路由实例并传递 `routes` 配置
// 你可以在这里输入更多的配置，但我们在这里
// 暂时保持简单
const router = VueRouter.createRouter({
  // 4. 内部提供了 history 模式的实现。为了简单起见，我们在这里使用 hash
  模式。
  history: VueRouter.createWebHashHistory(),
  routes: routes, // `routes: routes` 的缩写
});

export default router;

/**
 *
 */

```

/#/abc/adfda hash 路由模式
 /abc/adfda history 路由模式 需要后端配合 nginx 404

修改 main.js

```
/*
 * @Author: zhuwenzhi
 * @Date: 2023-06-01 09:48:19
 * @LastEditTime: 2023-06-01 10:54:09
 * @LastEditors: zhuwenzhi
 * @FilePath: /vue3-crm-demo/src/main.js
 * @Description:
 */
import { createApp } from 'vue';
import './style.css';
import App from './App.vue';
import ElementPlus from 'element-plus';
// elementplus 对应的样式
import 'element-plus/dist/index.css';
import router from './router/index.js';

const app = createApp(App);
app.use(ElementPlus);
app.use(router);
app.mount('#app');
```

基本使用

路由定义

1. 引入路由文件
2. 定义路由

```
import HelloWorld from '../components/HelloWorld.vue';

const routes = [
  {
    name: 'hello', // 路由名称
    path: '/HelloWorld', // 路由访问路径 ,eg:
    http://127.0.0.1:5173/#/HelloWorld
    component: HelloWorld, // 对应路由组件, 静态
  },
  {
    name: 'hello',
    path: '/HelloWorld',
    component: import('../components/HelloWorld.vue'), // 动态路由方式
  },
];
```

路由传参 query 方式

在 script 标签中, 添加以下内容

```
import { useRoute } from 'vue-router';

// http://127.0.0.1:5173/#/HelloWorld?name=123123
const route = useRoute();
// query 方式
const queryParams = route.query;
let { name } = queryParams; // 123123
const newName = ref(name)
```

404 路由

```
{
  // 找不到路由重定向到 404 页面
  path: '/*',
  redirect: '/404', // 重定向
  hideMenu: true,
},
```

pinia 全局状态管理

介绍

pinia 是新一代状态管理框架, 用于项目多组件之间做数据共享

[Pinia 中文文档](#)

VITE 项目配置

安装 vite 插件

```
npm install @vitejs/plugin-vue -D
```

Vue3 基础知识

- 双向绑定 ref, reactive
- Vue 事件, click 事件 v-on:click
- 组件封装 (父组件传递参数给子组件, 子组件传参给父组件) defineProps, defineEmits
- watch、watchEffect 监听
- computed 计算属性

一个 vue 文件,有三部分组件

```

<!--
  * @Author: zhuwenzhi
  * @Date: 2023-02-07 14:35:30
  * @LastEditTime: 2023-02-07 16:35:39
  * @LastEditors: zhuwenzhi
  * @FilePath: /finedo-demo/src/components/HelloWorld.vue
  * @Description:
-->
<!--

```

setup 是 vue3.2 提供的语法糖，可以简化 vue 代码编写

```

-->
<script>
// export default {
//   data () {
//     return {
//       user2: {
//         name: 'xiaoli',
//         age: 18
//       }
//     }
//   },
//   methods: {

//   }
// }

</script>
<script setup>
// 引入 ref , watch
//
https://24kcs.github.io/vue3\_study/chapter4/01\_Composition%20API\_%E5%B8%B8%E7%94%A8%E9%83%A8%E5%88%86.html#\_1-setup

import { ref, reactive, watch } from 'vue'
/**
   ref 设置双向绑定的变量 , const count = ref(0)
   怎么在 js 中获取 count 的值呢?
   这里通过 count.value 获取 count 的值
* */

```

打开 App.vue

```

<!--
  * @Author: zhuwenzhi
  * @Date: 2023-02-07 14:35:30
  * @LastEditTime: 2023-02-07 16:29:21
  * @LastEditors: zhuwenzhi
  * @FilePath: /finedo-demo/src/App.vue
  * @Description:
-->
<script setup>
import HelloWorld from './components/HelloWorld.vue'

// 获取 HelloWorld 中 user.name 的值
function getHelloWorldName(name) {
  console.log('getHelloWorldName = ', name);
}
</script>

<template>
  <div>
    <a href="https://vitejs.dev" target="_blank">
      
    </a>
    <a href="https://vuejs.org/" target="_blank">
      
  </div>
</div>

```

<!--

App.vue 是一个父组件

HelloWorld 相当于一个子组件

怎么使用组件

1. script 中引入 HelloWorld 组件, `import HelloWorld from './components/HelloWorld.vue'`
2. 接着在 template 中, 增加 `<HelloWorld msg="finedo Vue" />` 标签

怎么进行组件传参?

传参数有二种

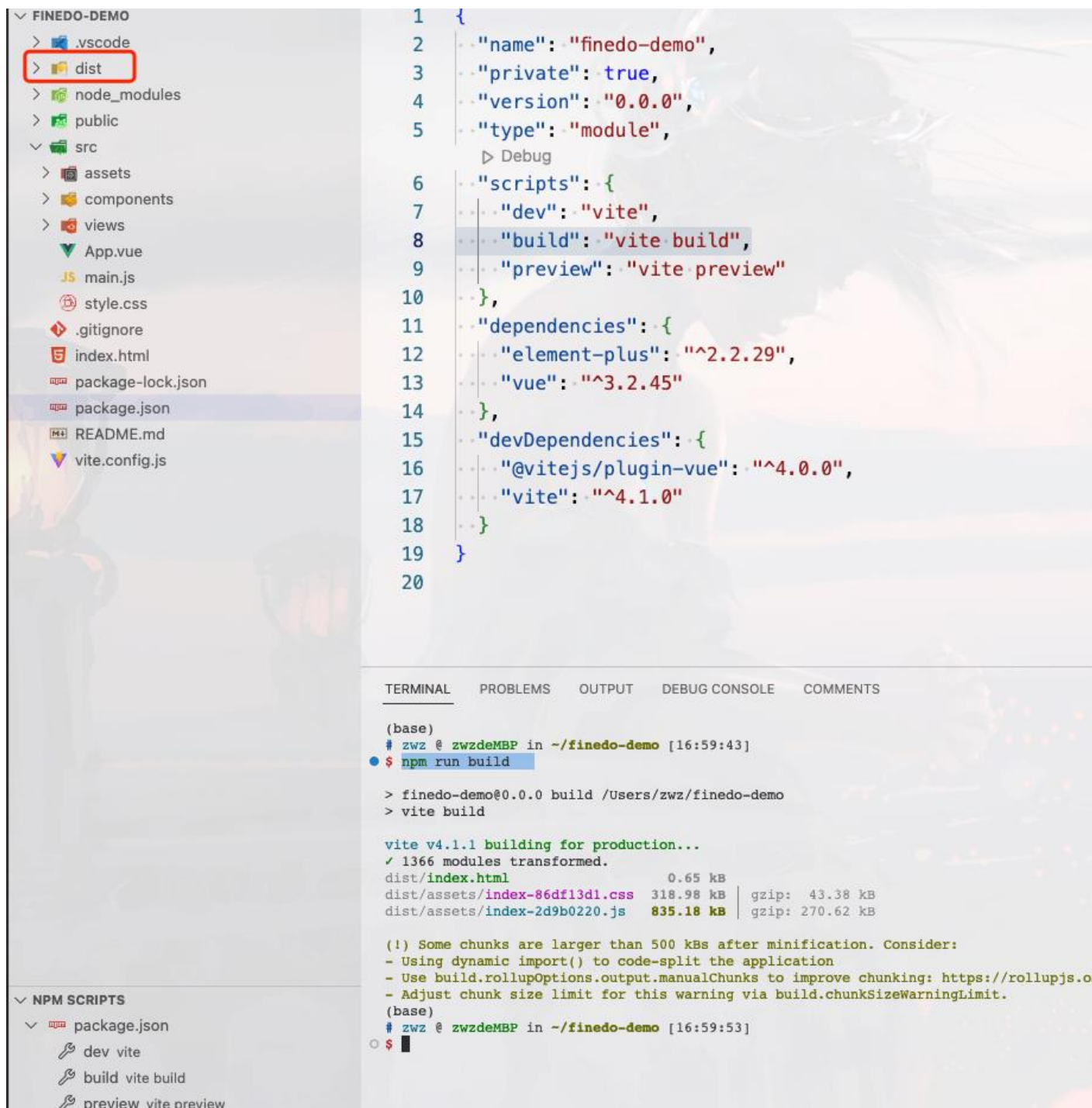
1. 父传子
2. 子传父

前端项目打包

打开命令行输入以下命令

```
npm run build
```

打包成功后自动在当前根目录生成一个 dist 文件夹



参考

- [简介 | Vue.js](#)
- [Ant Design Vue](#)
- [入门 | Vue Router](#)
- [Vue3 新特性和 Ts 基础语法](#)
- ElementPlus 组件库: <https://element-plus.gitee.io/zh-CN/guide/design.html>

- [重新介绍 JavaScript \(JS 教程\) - JavaScript | MDN](#)