常用的命名规范：

camelCase （小驼峰式命名法–首字母小写）

PascalCase （大驼峰式命名法–首字母大写）

kebab-case（短横线连接式）

Snake（下划线连接式）

目录命名规则：“静态文件下划线，编译文件短横线"

my-project-name/

|- BuildScript // 流水线部署文件目录

|- docs // 项目的细化文档目录（可选）

|- nginx // 部署在容器上前端项目 nginx 代理文件目录

|- node\_modules // 下载的依赖包

|- public // 静态页面目录

|- index.html // 项目入口

|- src // 源码目录

|- api // http 请求目录

|- assets // 静态资源目录，这里的资源会被wabpack构建

|- icon // icon 存放目录

|- img // 图片存放目录

|- banner\_sina.png // 采用全小写，优先选择单词命名，多个词以下划线相隔

|- js // 公共 js 文件目录

|- date-util.js // 采用全小写，优先选择单词命名，多个词以短横线相隔

|- scss // 公共样式 scss 存放目录

|- frame.scss // 入口文件

|- global.scss // 公共样式

|- input-number.scss // 采用全小写，优先选择单词命名，多个词以短横线相隔

|- components // 组件

|- plugins // 插件

|- router // 路由

|- routes // 详细的路由拆分目录（可选）

|- index.js

|- store // 全局状态管理

|- utils // 工具存放目录

|- request.js // 公共请求工具

|- views // 页面存放目录

|- App.vue // 根组件

|- main.js // 入口文件

|- tests // 测试用例

|- .browserslistrc// 浏览器兼容配置文件

|- .editorconfig // 编辑器配置文件

|- .eslintignore // eslint 忽略规则

|- .eslintrc.js // eslint 规则

|- .gitignore // git 忽略规则

|- babel.config.js // babel 规则

|- Dockerfile // Docker 部署文件

|- jest.config.js

|- package-lock.json

|- package.json // 依赖

|- README.md // 项目 README

|- vue.config.js // webpack 配置

vue组件命名：

components/

|- MyTabbar.vue // PascalCase

// 只有单个活跃实例的组件应该以The前缀命名，以示其唯一。这些组件在每个页面只用一次，并且

不接受prop，比如头部和侧边栏组件。

components/

|- TheHeading.vue

|- TheSidebar.vue

components/

|- BaseButton.vue

|- BaseTable.vue

|- BaseIcon.vue

components/

|- CustomCard.vue // 具有复用性

// 与父组件紧密耦合的子组件应以父组件名作为前缀

components/

|- TodoList.vue

|- TodoListItem.vue

router：

搜索引擎只会区分短横线分隔的命名语义

{

path: '/user-info', // 能解析成 user info

name: 'UserInfo',

component: UserInfo,

},

css属性顺序：

局定位属性：display / position / float / clear / visibility / overflow

自身属性：width / height / margin / padding / border / background

文本属性：color / font / text-decoration / text-align / vertical-align / white- space / break-word

其他属性（CSS3）： cursor / border-radius / box-shadow / text-shadow / background: linear-gradient …

规范二：

### 目录结构

- ├── public # 公共静态资源目录

- ├── src # 主目录

- │ ├── api # 接口文件

- │ ├── assets # 资源文件

- │ │ ├── icons # icon sprite 图标文件夹

- │ │ ├── images # 项目存放图片的文件夹

- │ │ └── svg # 项目存放 svg 图片的文件夹

- │ ├── components # 全局公共组件

- │ ├── directives # 自定义指令文件夹

- │ ├── layouts # 布局文件

- │ │ ├── content # 内容布局

- │ │ ├── footer # 页脚布局

- │ │ ├── navbar # 顶部布局

- │ │ └── sider # 侧边栏布局

- │ ├── main.ts # 主入口

- │ ├── router # 路由配置

- │ ├── store # 数据仓库

- │ ├── styles # 样式文件

- │ ├── types # 类型文件

- │ ├── utils # 工具类

- │ └── views # 所有页面目录

- │ └── kebab-case # 页面

- │ └── module # 页面级组件

- ├── vite.config.ts # vite 配置文件

- └── windi.config.ts # windcss 配置文件

### 项目命名规范

##### 非组件文件夹命名：

统一使用中间横杠的形式，例：kebab-case

##### 组件文件夹命名：

统一使用文件夹包含文件的形式，文件夹命名使用PascalBase风格；除了index文件，其他文件使用PascalBase风格，例：

[components]

- [Breadcrumb]

- index.vue

- [Hamburger]

- index.vue

- Item.vue

- [SvgIcon]

- index.vue

- SvgItem.vue

##### 文件命名：

\*.js/ts命名：

1. 类文件，除index外，使用PascalBase风格；

2. 其他类型文件，使用kebab-case风格；

3. 属于Api的，统一加上Api后缀，例：HomeApi；

\*.vue命名：

除index.vue之外，其他vue文件统一用PascalBase风格

\*.less/scss命名：

统一用PascalBase风格