



# UnoCSS 的探索和 OnuUI 的应用

🐧 现代原子化 CSS 最佳解决方案-UnoCSS

Press Space for next page →



# Evan Yang

- Core team member of [Anse](#), [OnuUI](#),.
- Creator of [magic-cli](#), [za-zi](#), [json-tsc](#).
- Contributor of [UnoCSS](#), [Vitest](#), [vite-plugin-vue2](#), [vitesse-nuxt3](#) and more.



[Evan-Yang](#)



[Mohen](#)



[evan-yang.top](#)



# Contents Toc of this shared


你可以提前学习或了解 [Atomic CSS](#), [UnoCSS](#), [OnuUI](#)


1. [What is Atomic CSS](#)
2. [Why Atomic CSS](#)
3. [How to use Atomic CSS](#)
4. [What is UnoCSS](#)
5. [Why UnoCSS](#)
6. [How to use UnoCSS](#)
7. [What is OnuUI](#)
8. [How to use OnuUI](#)
9. [Morden CSS FrameWork Usage rate](#)
10. [How to Start a UI Design System](#)


# What is Atomic CSS


Atomic CSS is the approach to CSS architecture that favors small, single-purpose classes with names based on visual function.

译文：原子化 CSS 是一种 CSS 的架构方式，它倾向于小巧且用途单一的 class，并且会以视觉效果进行命名。

 Tailwind CSS

 Windi CSS

 UnoCSS

 Open Props

 Bootstrap Utilities

 Atomizer CSS

 Primer CSS

# Why Atomic CSS?



# Why Atomic CSS?

- 📝 不用想类名 - 相信很多人在编写样式类的时候经常会纠结类名，原子化提供的类名都是能够一眼就能知道大概意思又比直接编写style更加简洁

# Why Atomic CSS?

- 📝 **不用想类名** - 相信很多人在编写样式类的时候经常会纠结类名，原子化提供的类名都是能够一眼就能知道大概意思又比直接编写style更加简洁
- 🎨 **无需离开您的HTML，即可快速建立现代网站** - 对于现在组件化开发的方式来说，单个组件文件相比以前一个html对应一个页面来说代码量要小很多，可能就几行html的组件代码，直接在html中编写样式是个更好的选择

# Why Atomic CSS?

- 📝 **不用想类名** - 相信很多人在编写样式类的时候经常会纠结类名，原子化提供的类名都是能够一眼就能知道大概意思又比直接编写style更加简洁
- 🎨 **无需离开您的HTML，即可快速建立现代网站** - 对于现在组件化开发的方式来说，单个组件文件相比以前一个html对应一个页面来说代码量要小很多，可能就几行html的组件代码，直接在html中编写样式是个更好的选择
- 📁 **强大的预设** - 利用原子化框架提供的预设原子类，极大地提高开发效率，将更多时间用在页面构造而不是重复地编写相似的代码



# Why Atomic CSS?

- 📝 **不用想类名** - 相信很多人在编写样式类的时候经常会纠结类名，原子化提供的类名都是能够一眼就能知道大概意思又比直接编写style更加简洁
- 🎨 **无需离开您的HTML，即可快速建立现代网站** - 对于现在组件化开发的方式来说，单个组件文件相比以前一个html对应一个页面来说代码量要小很多，可能就几行html的组件代码，直接在html中编写样式是个更好的选择
- 📁 **强大的预设** - 利用原子化框架提供的预设原子类，极大地提高开发效率，将更多时间用在页面构造而不是重复地编写相似的代码
- 👤 **IDE支持** - VS Code 的 Tailwind CSS 智能提示扩展涵盖了所有的类。在编辑器内既可得到智能的自动完成建议、提示及类定义等功能，而且无需配置。

# Why Atomic CSS?

- 📝 **不用想类名** - 相信很多人在编写样式类的时候经常会纠结类名，原子化提供的类名都是能够一眼就能知道大概意思又比直接编写style更加简洁
- 🎨 **无需离开您的HTML，即可快速建立现代网站** - 对于现在组件化开发的方式来说，单个组件文件相比以前一个html对应一个页面来说代码量要小很多，可能就几行html的组件代码，直接在html中编写样式是个更好的选择
- 📁 **强大的预设** - 利用原子化框架提供的预设原子类，极大地提高开发效率，将更多时间用在页面构造而不是重复地编写相似的代码
- 👤 **IDE支持** - VS Code 的 Tailwind CSS 智能提示扩展涵盖了所有的类。在编辑器内既可得到智能的自动完成建议、提示及类定义等功能，而且无需配置。

Read more about [Why Atomic CSS?](#)

# How to use Atomic CSS



"Tailwind CSS is the only framework that I've seen scale on large teams. It's easy to customize, adapts to any design, and the build size is tiny."

Sarah Dayan  
Staff Engineer, Algolia


```
1 <figure class="bg-slate-100 rounded-xl dark:bg-slate-800">
2   
4     <blockquote>
5       <p class="text-lg">
6         "Tailwind CSS is the only framework that I've seen sc
7         on large teams. It's easy to customize, adapts to any
8         and the build size is tiny."
9       </p>
10    </blockquote>
11    <figcaption>
12      <div>
13        Sarah Dayan
14      </div>
15      <div>
16        Staff Engineer, Algolia
17      </div>
18    </figcaption>
19  </div>
20 </figure>
21
```

# What is UnoCSS?



Read more about [UnoCSS](#)

# What is UnoCSS?

- 完全自定义 - Uno 没有核心使用入口，所有功能都是通过预设提供 



Read more about [UnoCSS](#)

# What is UnoCSS?

- 完全自定义 - Uno 没有核心使用入口，所有功能都是通过预设提供 📄
- 轻量且高性能 - Uno 没有解析，没有 AST，没有扫描，相较 WindiCSS/TailWindcss JIT快五倍，零依赖打包产物小于6kb 📄



Read more about [UnoCSS](#)

# What is UnoCSS?

- **完全自定义** - Uno 没有核心使用入口，所有功能都是通过预设提供 📄
- **轻量且高性能** - Uno 没有解析，没有 AST，没有扫描，相较 WindiCSS/TailWindcss JIT快五倍，零依赖打包产物小于6kb 📄
- **强大生态** - Uno 提供 Vite, Webpack, PostCSS, CLI, VSCode, Eslint, CDN 一键引入等支持 📄



Read more about [UnoCSS](#)

# What is UnoCSS?

- **完全自定义** - Uno 没有核心使用入口，所有功能都是通过预设提供 📄
- **轻量且高性能** - Uno 没有解析，没有 AST，没有扫描，相较 WindiCSS/TailWindcss JIT快五倍，零依赖打包产物小于6kb 📄
- **强大生态** - Uno 提供 Vite, Webpack, PostCSS, CLI, VSCode, Eslint, CDN 一键引入等支持 📄
- **强大的语法配置** - Uno 支持配置 shortcuts, attributify mode, pure css icons, variant groups, CSS Directives 等 📄

Read more about [UnoCSS](#)





# Why UnoCSS?



UnoCSS 灵感来自于 [Anthony Fu](#) 对 [Windi CSS](#) 的大量维护工作，它完美继承了 [Windi CSS](#) 的按需加载，属性模式，快捷方式，变量组，编译模式等等特性，在此基础上做了大量的性能提升和打包优化，当然也是和 [Tailwind CSS](#) 为基础搭建的，UnoCSS 在 TailWind 的基础上构建了一个自定义设计系统，你无需像在 TailWind 那样遵守约定，用户可以自定义预设达到实现任何想要的功能。

# Why UnoCSS?



UnoCSS 灵感来自于 [Anthony Fu](#) 对 [Windi CSS](#) 的大量维护工作，它完美继承了 [Windi CSS](#) 的按需加载，属性模式，快捷方式，变量组，编译模式等等特性，在此基础上做了大量的性能提升和打包优化，当然也是和 [Tailwind CSS](#) 为基础搭建的，UnoCSS 在 TailWind 的基础上构建了一个自定义设计系统，你无需像在 TailWind 那样遵守约定，用户可以自定义预设打到实现任何想要的功能。

UnoCSS提供的灵活性，让我们能够在它之上实践很多新功能：比如

# Why UnoCSS?



UnoCSS 灵感来自于 [Anthony Fu](#) 对 [Windi CSS](#) 的大量维护工作，它完美继承了 [Windi CSS](#) 的按需加载，属性模式，快捷方式，变量组，编译模式等等特性，在此基础上做了大量的性能提升和打包优化，当然也是和 [Tailwind CSS](#) 为基础搭建的，UnoCSS 在 TailWind 的基础上构建了一个自定义设计系统，你无需像在 TailWind 那样遵守约定，用户可以自定义预设达到实现任何想要的功能。

UnoCSS提供的灵活性，让我们能够在它之上实践很多新功能：比如

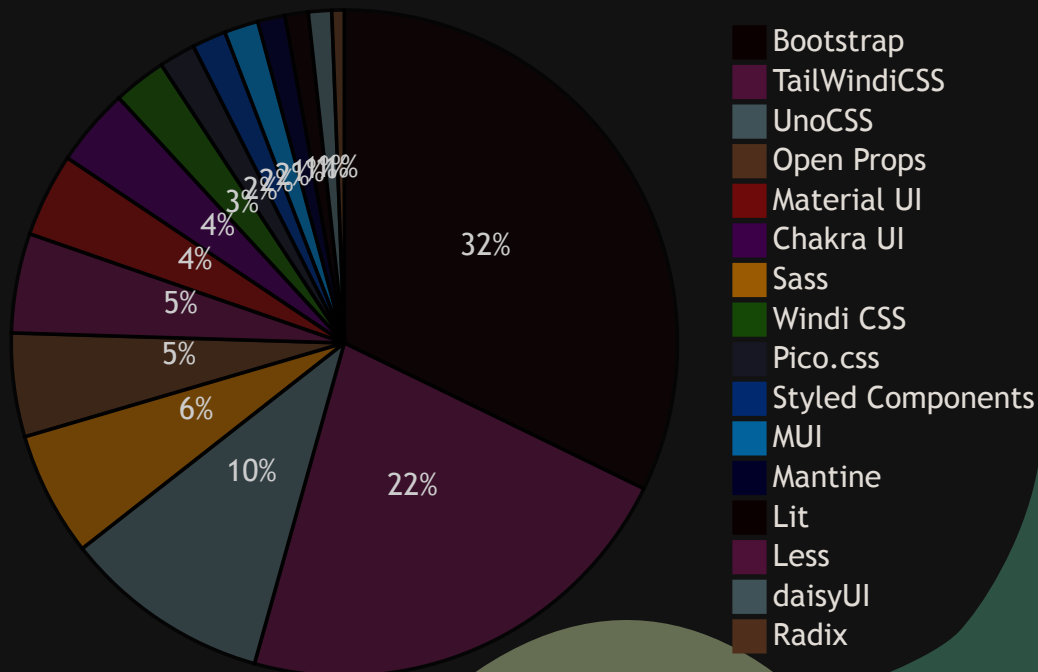
- [Pure CSS icons 纯 CSS 图标](#)
- [Attributify Mode](#)
- [Variant Groups](#)
- [Shortcuts](#)
- [Tagify](#)
- [Web fonts](#)
- [CDN Runtime](#)
- [Inspector](#)

# Morden CSS FrameWork Usage rate



# Morden CSS FrameWork Usage rate

Morden CSS FrameWork Usage rate

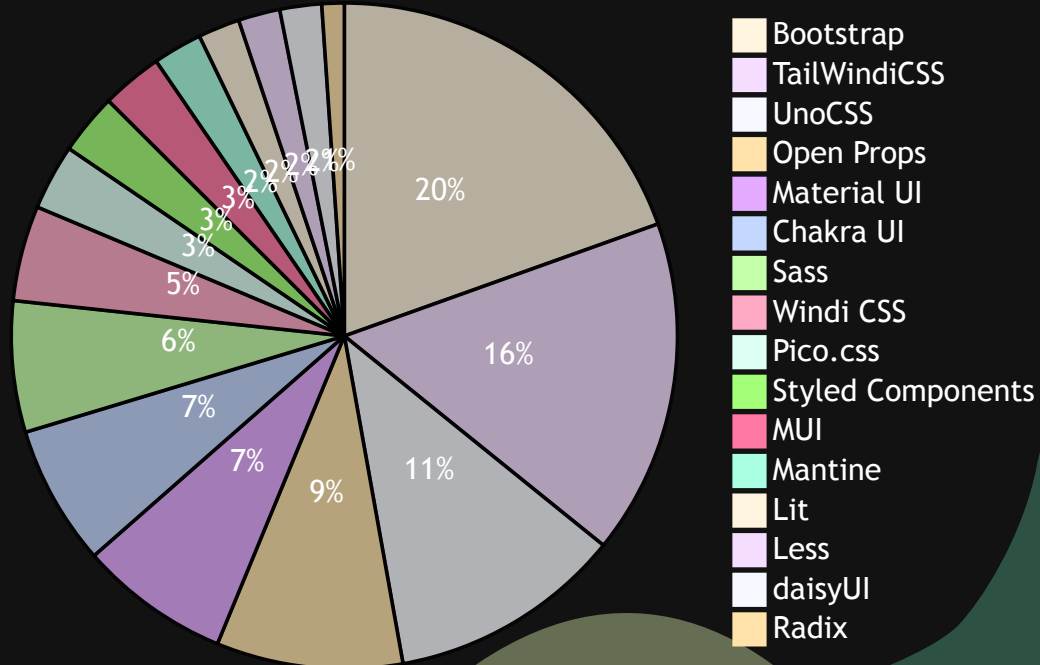


# Morden CSS FrameWork Awareness rate



# Morden CSS FrameWork Awareness rate

Morden CSS FrameWork Awareness rate



# How to use UnoCSS

Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```



# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

CDN

Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

CDN

```
<script src="https://cdn.jsdelivr.net/npm/@unocss/vite">
```

Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

## CDN

```
<script src="https://cdn.jsdelivr.net/npm/@unocss">
```

## Reset CSS CDN



# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

## CDN

```
<script src="https://cdn.jsdelivr.net/npm/@unocss" />
```

## Reset CSS CDN

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@unocss/reset" />
<!-- or -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@unocss/reset" />
```

# How to use UnoCSS

## Vue + Vite Plugin

```
// vite.config.ts
import UnoCSS from 'unocss/vite'
import { defineConfig } from 'vite'

export default defineConfig({
  plugins: [
    UnoCSS()
  ],
})

// uno.config.ts
import { defineConfig } from 'unocss'

export default defineConfig({
  // ...UnoCSS options
})

// main.ts
import 'virtual:uno.css'
```

## CDN

```
<script src="https://cdn.jsdelivr.net/npm/@unocss" data-bbox="536 186 944 216">
```

## Reset CSS CDN

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@unocss/reset" data-bbox="536 348 944 378">
<!-- or -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@unocss/reset" data-bbox="536 413 944 444">
```

Read more about [UnoCSS Integrations](#)

# What is OnuUI

OnuUI 是一个基于 Vue3 的 UnoCSS 组件库，允许您制作漂亮的 websites.它包括一个全面的预构建组件集合，这些组件可以开箱即用地在生产中使用。

组件库利用最新的技术栈 Vue3 Setup Script + Vite 4 搭建，全量使用 Typescript 编写，在打包和性能方面做了许多优化，是目前 UnoCSS 生态中为数不多的开源组件库。

- 🌈 **Components Design** - Onu provides neat & beautiful crafted UI components.
- 🔥 **On demand Import** - Provide resolver to automatically import only used components.
- 🎪 **Typescript Supported** - Support TypeScript & type checked & type inference.
- 💎 **Iconify Icons** - Use any icon from the library you love.
- 🧪 **CSS Preset** - Has UnoCSS preset package to use, rendered UI easily.
- ⚙️ **Theme Config** - Use attribute mode like unocss to design. Support theme config to customize theme.

# How to Start a UI Design System



# How to Start a UI Design System

搭建组件库必备的过程：



# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码


# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
- 支持按需引入，应该没有人愿意全量导入组件库

# How to Start a UI Design System


搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
  - 支持按需引入，应该没有人愿意全量导入组件库
  - 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
- 
- The bottom right corner of the slide features several overlapping, semi-transparent abstract shapes in shades of green and teal, creating a modern, organic design element.



# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
  - 支持按需引入，应该没有人愿意全量导入组件库
  - 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
  - 构建开发文档，至少得有个中文文档说明一下组件怎么用
- 

# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
- 支持按需引入，应该没有人愿意全量导入组件库
- 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
- 构建开发文档，至少得有个中文文档说明一下组件怎么用
- 单元测试，你写的代码得信的过

# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
- 支持按需引入，应该没有人愿意全量导入组件库
- 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
- 构建开发文档，至少得有个中文文档说明一下组件怎么用
- 单元测试，你写的代码得信的过
- 桌面端和移动端的组件预览，你得让使用者看到组件具体长什么样子

# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
- 支持按需引入，应该没有人愿意全量导入组件库
- 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
- 构建开发文档，至少得有个中文文档说明一下组件怎么用
- 单元测试，你写的代码得信的过
- 桌面端和移动端的组件预览，你得让使用者看到组件具体长什么样子
- 代码格式化和规范检测工具，毕竟是团队作案，没有规矩不成方圆

# How to Start a UI Design System

搭建组件库必备的过程：

- 开发环境，你得起个服务去调试代码
- 支持按需引入，应该没有人愿意全量导入组件库
- 组件库编译，生成 umd 和 esm 模块的组件代码,类型提示文件等等
- 构建开发文档，至少得有个中文文档说明一下组件怎么用
- 单元测试，你写的代码得信的过
- 桌面端和移动端的组件预览，你得让使用者看到组件具体长什么样子
- 代码格式化和规范检测工具，毕竟是团队作案，没有规矩不成方圆
- 自动化的文档部署和测试流程，总不能每次发布版本都手动去部署文档和测试


# How is OnuUI implemented?



# How is OnuUI implemented?


- 开发环境: Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript

# How is OnuUI implemented?


- 开发环境: Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
  - 打包方面: Vite + Tsup + Esbuild + Vue-tsc
- 
- The bottom right corner of the slide features several overlapping, semi-transparent abstract shapes in shades of green and teal, creating a modern, organic design element.




# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
  - 打包方面： Vite + Tsup + Esbuild + Vue-tsc
  - 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
- 
- The bottom right corner of the slide features decorative abstract shapes. There are two overlapping rounded shapes: a larger, darker teal one on the right and a smaller, lighter olive-green one on the left, both with smooth, organic edges.

# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
  - 打包方面： Vite + Tsup + Esbuild + Vue-tsc
  - 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
  - 组件库playground： 借鉴 Vue 官方 playground，利用 @vue/repl 和依赖的 CDN 搭建
- 
- The bottom right corner of the slide features several overlapping, semi-transparent abstract shapes in shades of green and teal, creating a modern, organic design element.

# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
  - 打包方面： Vite + Tsup + Esbuild + Vue-tsc
  - 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
  - 组件库playground： 借鉴 Vue 官方 playground，利用 @vue/repl 和依赖的 CDN 搭建
  - 单元测试： Vitest + @vue/test-utils 组件测试细节，测试报告，覆盖率等等
- 

# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
  - 打包方面： Vite + Tsup + Esbuild + Vue-tsc
  - 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
  - 组件库playground： 借鉴 Vue 官方 playground，利用 @vue/repl 和依赖的 CDN 搭建
  - 单元测试： Vitest + @vue/test-utils 组件测试细节，测试报告，覆盖率等等
  - 文档服务： Vitepress + netlify + Vitepress Code Block plugin.
- 

# How is OnuUI implemented?

- 开发环境: Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
- 打包方面: Vite + Tsup + Esbuild + Vue-tsc
- 代码校验方面: simple-git-hook + lint-staged + eslint + eslint-config
- 组件库playground: 借鉴 Vue 官方 playground, 利用 @vue/repl 和依赖的 CDN 搭建
- 单元测试: Vitest + @vue/test-utils 组件测试细节, 测试报告, 覆盖率等等
- 文档服务: Vitepress + netlify + Vitepress Code Block plugin.
- CI, CD: Github Action + changelogithub

# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
- 打包方面： Vite + Tsup + Esbuild + Vue-tsc
- 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
- 组件库playground： 借鉴 Vue 官方 playground，利用 @vue/repl 和依赖的 CDN 搭建
- 单元测试： Vitest + @vue/test-utils 组件测试细节，测试报告，覆盖率等等
- 文档服务： Vitepress + netlify + Vitepress Code Block plugin.
- CI, CD： Github Action + changelogithub
- 工具方面： esno + rimraf + unplugin-auto-imports + unplugin-vue-components + vite-plugin-vue-setup-extend

# How is OnuUI implemented?

- 开发环境： Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
- 打包方面： Vite + Tsup + Esbuild + Vue-tsc
- 代码校验方面： simple-git-hook + lint-staged + eslint + eslint-config
- 组件库playground： 借鉴 Vue 官方 playground，利用 @vue/repl 和依赖的 CDN 搭建
- 单元测试： Vitest + @vue/test-utils 组件测试细节，测试报告，覆盖率等等
- 文档服务： Vitepress + netlify + Vitepress Code Block plugin.
- CI, CD： Github Action + changelogithub
- 工具方面： esno + rimraf + unplugin-auto-imports + unplugin-vue-components + vite-plugin-vue-setup-extend

编写一个组件的全流程

# How is OnuUI implemented?

- 开发环境: Monorepo + Pnpm workspaces + Vue3 + Vite4 + UnoCSS + Typescript
- 打包方面: Vite + Tsup + Esbuild + Vue-tsc
- 代码校验方面: simple-git-hook + lint-staged + eslint + eslint-config
- 组件库playground: 借鉴 Vue 官方 playground, 利用 @vue/repl 和依赖的 CDN 搭建
- 单元测试: Vitest + @vue/test-utils 组件测试细节, 测试报告, 覆盖率等等
- 文档服务: Vitepress + netlify + Vitepress Code Block plugin.
- CI, CD: Github Action + changelogithub
- 工具方面: esno + rimraf + unplugin-auto-imports + unplugin-vue-components + vite-plugin-vue-setup-extend

编写一个组件的全流程

提 Feture Request => 讨论同意 => 原型设计 => 组件功能和配置确定 => 按照 Contrubiting md 开发 => 单元测试 => 编写文档 => 提 MR => MR Request => 发版本



# How to use OnuUI

## Full Import

```
pnpm add onu-ui && pnpm add -D unocss
```

```
// main.ts
import OnuUI from 'onu-ui'
import 'uno.css'
import 'onu-ui/dist/style.css'

createApp(App).use(OnuUI).mount('#app')
```

```
// Get Volar Support tsconfig.json
{
  "compilerOptions": {
    // ...
    "types": ["onu-ui/volar"]
  }
}
```

# How to use OnuUI

On-demand Import

## Full Import

```
pnpm add onu-ui && pnpm add -D unocss
```

```
// main.ts
import OnuUI from 'onu-ui'
import 'uno.css'
import 'onu-ui/dist/style.css'

createApp(App).use(OnuUI).mount('#app')
```

```
// Get Volar Support tsconfig.json
{
  "compilerOptions": {
    // ...
    "types": ["onu-ui/volar"]
  }
}
```

# How to use OnuUI

## On-demand Import

```
pnpm add -D unplugin-vue-components unplugin-au
```

## Full Import

```
pnpm add onu-ui && pnpm add -D unocss
```

```
// main.ts
import OnuUI from 'onu-ui'
import 'uno.css'
import 'onu-ui/dist/style.css'

createApp(App).use(OnuUI).mount('#app')
```

```
// Get Volar Support tsconfig.json
{
  "compilerOptions": {
    // ...
    "types": ["onu-ui/volar"]
  }
}
```

# How to use OnuUI

## Full Import

```
pnpm add onu-ui && pnpm add -D unocss
```

```
// main.ts
import OnuUI from 'onu-ui'
import 'uno.css'
import 'onu-ui/dist/style.css'

createApp(App).use(OnuUI).mount('#app')
```

```
// Get Volar Support tsconfig.json
{
  "compilerOptions": {
    // ...
    "types": ["onu-ui/volar"]
  }
}
```

## On-demand Import

```
pnpm add -D unplugin-vue-components unplugin-auto-import
```

```
// vite.config.ts
import { defineConfig } from 'vite'
import AutoImport from 'unplugin-auto-import/vite'
import Components from 'unplugin-vue-components/vite'
import { OnuResolver } from 'onu-ui'

export default defineConfig({
  // ...
  plugins: [
    // ...
    AutoImport({
      resolvers: [OnuResolver()],
    }),
    Components({
      resolvers: [OnuResolver()],
    }),
  ],
})
```

# How to use OnuUI

## Full Import

```
pnpm add onu-ui && pnpm add -D unocss
```

```
// main.ts
import OnuUI from 'onu-ui'
import 'uno.css'
import 'onu-ui/dist/style.css'

createApp(App).use(OnuUI).mount('#app')
```

```
// Get Volar Support tsconfig.json
{
  "compilerOptions": {
    // ...
    "types": ["onu-ui/volar"]
  }
}
```

## On-demand Import

```
pnpm add -D unplugin-vue-components unplugin-auto-import
```

```
// vite.config.ts
import { defineConfig } from 'vite'
import AutoImport from 'unplugin-auto-import/vite'
import Components from 'unplugin-vue-components'
import { OnuResolver } from 'onu-ui'

export default defineConfig({
  // ...
  plugins: [
    // ...
    AutoImport({
      resolvers: [OnuResolver()],
    }),
    Components({
      resolvers: [OnuResolver()],
    }),
  ],
})
```

# Awesome UnoCSS



## Community presets

- Community presets: A collection of community-made presets for UnoCSS



## Community frameworks

- Community frameworks: A collection of community-made UI frameworks for UnoCSS



## Awesome Projects

- ChatGPT-demo: Minimal UI for ChatGPT, Powered by unocss + solid + astro + ssr + openAI
- Anse-app: Supercharged experience for ChatGPT, Powered by unocss + solid + astro + ssr + openAI
- UnoCSS Animates: Preview all animations in UnoCSS.

# Speak freely

聊聊最近学的新的技术或者看过的最新的技术等等