

Tailwind [v4.1]

作者： -- 天气预报

日期： -- 2025-09-20

01. 官网

≡ <https://tailwindcss.com/>

- ui

The screenshot shows the TailwindCSS website homepage. At the top, there's a navigation bar with a search icon, 'Ctrl K', 'Docs', 'Blog', 'Showcase', 'Sponsor', and a 'Plus' button. Below the navigation is a large hero section with the text 'Rapidly build modern websites without ever leaving your HTML.' and a subtext about utility-first CSS. A 'Get started' button is visible. On the right side, there's a small image of a book titled 'The Anti-Patterns No. 4' with the subtitle 'Close Warfare'. A code editor window is overlaid on the bottom left, displaying a snippet of Tailwind CSS code:

```
1 <div class="flex flex-col items-center p-7 rounded-2xl">
2   <div>
3     
4   </div>
5   <div class="flex">
6     <span class="text-2xl font-medium">Class Warfare</span>
7     <span class="font-medif>The Anti-Patterns</span>
8     <span class="flex">
9       <span>No. 4</span>
10      <span>Close Warfare</span>
```

ⓘ Note

✓ Tailwind (v4.1) 推荐新项目直接使用即可

1.1 文档

≡ <https://tailwindcss.com/docs/installation/using-vite>

- ui

The screenshot shows the Tailwind CSS documentation page for 'Using Vite'. The left sidebar has sections for Documentation, Components, Templates, UI Kit, Playground, Course (NEW), and Community. Under 'GETTING STARTED', 'Installation' is selected, showing links for Editor setup, Compatibility, and Upgrade guide. The main content area is titled 'Get started with Tailwind CSS' and explains how Tailwind CSS works by scanning HTML files for class names. It highlights its fast, flexible, and reliable nature. Below this is a 'Installation' section with three steps: 1. Create your project (using 'npm create vite@latest my-project cd my-project'), 2. Install Tailwind CSS (using 'npm install tailwindcss @tailwindcss/vite'), and 3. Configure the Vite plugin (adding 'import { defineConfig } from "vite"' to vite.config.ts). The top navigation bar includes a search icon, 'Ctrl K', 'Docs', 'Blog', 'Showcase', 'Sponsor', and a 'Plus' button.

- ui

1.2 在线运行

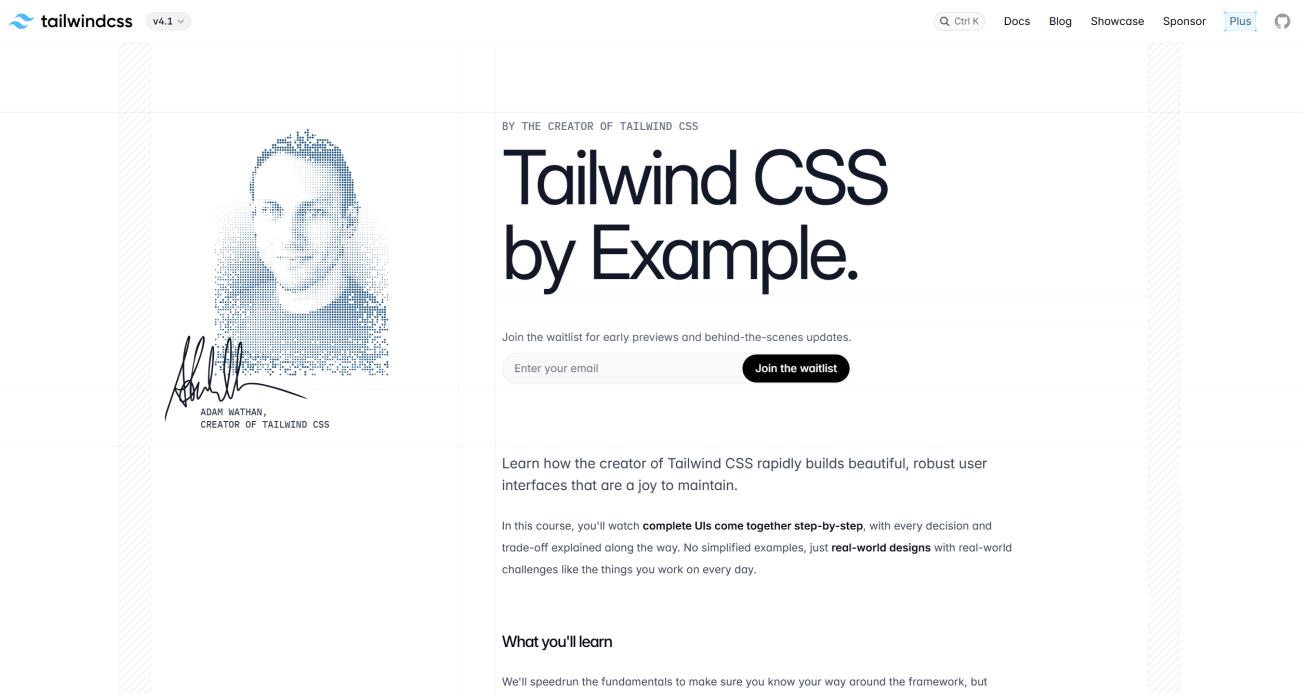
≡ <https://play.tailwindcss.com/>

● ui

1.3 福利课程

≡ <https://tailwindcss.com/course>

- ✓ 邮件报名、邮件推送油管视频 [科学上网] (* 小课程、分享特殊场景案例实现)



1.4 模版服务

≡ <https://tailwindcss.com/plus/templates?ref=sidebar>

- ui

The screenshot shows the Tailwind Plus website. At the top, there's a banner for Tailwind Plus, followed by a search bar and navigation links for UI Blocks, Templates, UI Kit, Sign in, and Get full access. Below this is a section titled "TEMPLATES BY THE MAKERS OF TAILWIND CSS" featuring a large image of a modern website template with a pink-to-yellow gradient header containing the text "Close every deal.". The template also includes sections for "Simplifying everyday business tasks." and "A snapshot of your entire sales pipeline." At the bottom left, there are buttons for "Browse templates" and "Get everything with Tailwind Plus".

1.5 组件

≡ <https://tailwindcss.com/plus/ui-blocks?ref=sidebar>

✓ 免费

● ui

[tailwind PLUS](#) Now with vanilla JavaScript support • Learn more >

PAGE SECTIONS MARKETING APPLICATION UI ECOMMERCE PRICING

Marketing

Heroes, feature sections, newsletter sign up forms — everything you need to build beautiful marketing websites.

Hero Sections 12 components

Feature Sections 15 components

CTA Sections 11 components

Bento Grids 3 components

Pricing Grids 12 components

About us 11 components

Sign up newsletter 8 components

Trusted by developers 10 components

● ui

Marketing Application UI Ecommerce Documentation

PAGE SECTIONS Hero Sections Feature Sections CTA Sections Bento Grids Pricing Sections Header Sections Newsletter Sections Stats Testimonials

Hero Sections

UI BLOCKS / MARKETING / PAGE SECTIONS

Use these Tailwind CSS hero section examples to add important messaging, product photos, and call-to-actions to the top of your website. These hero examples are designed and built by the Tailwind CSS team, and include a variety of different styles and layouts.

Simple centered

Preview Code

Data to enrich your online business

1.6 博客

≡ <https://tailwindcss.com/blog>

● ui

The screenshot shows the Tailwind CSS website's "Latest Updates" section. At the top, there's a navigation bar with links for "Ctrl K", "Docs", "Blog", "Showcase", "Sponsor", and "Plus". Below the navigation is a large, stylized illustration of a keyboard on a grid background. The main content area features two news items. The first item, dated July 25, 2025, is titled "Vanilla JavaScript support for Tailwind Plus" and discusses UI blocks that require JavaScript. The second item, dated May 14, 2025, is titled "Compass: A starter kit for online courses" and describes a Next.js starter kit for publishing online courses.

1.7 插件

≡ <https://tailscan.com/>

- ui

The screenshot shows the Tailscan website. At the top, there's a navigation bar with links for "Home", "Features", "Pricing", "Blog", "Resources", "Support", and "My Account". The main headline reads "The best devtool for Tailwind CSS". Below it, a sub-headline says "Build, design and debug your Tailwind website visually with Tailscan, right within the browser." Two buttons are present: "Try on this page" and "Get Tailscan now". A section titled "Used by" shows a grid of small user icons. Below this is a screenshot of the Tailscan interface, which looks like a browser developer tools extension. It shows a code editor with Tailwind CSS classes applied, and a sidebar with various settings and preview options. A prominent message in the interface says "Never miss the cache again!"

☰ Tailscan Core | Tailscan Core + AI

● ui

The screenshot shows the Tailscan Core pricing page. At the top, it says "Pricing" and "Level up your development experience". It notes compatibility with Chrome and Chromium-based browsers (Edge, Brave, and Arc). Below this, two plans are listed:

- Tailscan Core**: \$79/one-time. Includes Pay once, use Core forever; Includes all Core updates, forever; License for 3 devices; Permanent access to the Tailscan community discord.
- Tailscan Core + Tailscan AI**: \$15/mo (Billed \$180 annually). Includes Tailscan AI (unlimited requests*); Pay once, use Core forever; Includes all Core updates, forever; License for 3 devices; Permanent access to the Tailscan community discord.

Both plans come with a 7-day refund policy and dedicated support. A note says they are used by over 1000 developers, designers, and product managers, accompanied by small profile icons.

● ui

The screenshot shows the Tailwind CSS documentation page for the v4.1 version. The left sidebar includes links for Documentation, Components, Templates, UI Kit, Playground, Course (NEW), and Community. Under "GETTING STARTED", "Installation" is selected, with sub-links for Editor setup, Compatibility, and Upgrade guide. The main content area is titled "Get started with Tailwind CSS". It explains how Tailwind CSS works by scanning HTML files and generating static CSS. It's fast, flexible, and reliable. Below this is an "Installation" section with steps for Using Vite, Using PostCSS, Tailwind CLI, Framework Guides, and Play CDN. Step 01: "Create your project" shows a terminal command: `npm create vite@latest my-project`. Step 02: "Install Tailwind CSS" shows the command: `npm install tailwindcss @tailwindcss/vite`. Step 03: "Configure the Vite plugin" shows the addition of the plugin to `vite.config.ts` with the code: `import { defineConfig } from 'vite'`. A floating interface on the right allows for styling a selected `h1` element with various classes like `mt-2`, `text-3xl`, `font-medium`, `tracking-tight`, `text-gray-950`, and `dark:text-white`.

● ui

The screenshot shows the TailwindCSS website's documentation page. On the left, there's a sidebar with sections like Documentation, Components, Templates, UI Kit, Playground, Course (NEW), and Community. Under GETTING STARTED, there are links for Installation, Core Concepts, Colors, and Functions and directives. A central modal window contains a tip about setting up Tailwind with Vite, followed by a link to explore framework guides. To the right, there's a dark-themed interface showing a code editor with Tailwind CSS utility classes and a terminal window.

02. 环境

[≡ https://tailwindcss.com/docs/installation/using-vite](https://tailwindcss.com/docs/installation/using-vite)

● ui

This screenshot shows a detailed guide on how to use TailwindCSS with Vite. It includes five numbered steps: 1. Create your project (using Create Vite), 2. Install Tailwind CSS (via npm), 3. Configure the Vite plugin (adding the tailwindcss/vite plugin to vite.config.ts), 4. Import Tailwind CSS (adding @import "tailwindcss"; to CSS), and 5. Start your build process (running npm run dev). Each step is accompanied by a terminal command or code snippet.

2.1 CDN

≡ <https://tailwindcss.com/docs/installation/play-cdn>

- `index.html`

```
● ● ●

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Tailwind</title>
    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4">
</script>
    <style type="text/tailwindcss">
      @theme {
        --color-primary: #000;
      }
    </style>
  </head>
  <body>
    <div class="p-10 bg-primary">
      <h1 class="text-2xl text-center text-white">tailwind</h1>
    </div>
  </body>
</html>
```

2.2 Tailwind CLI

≡ <https://nodejs.org/zh-cn>

- `ui`

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

[Get Node.js®](#)

Get security support
for EOL Node.js versions

[Create an HTTP Server](#) [Write Tests](#) [Read and Hash a File](#) [Streams Pipeline](#) [Work with Threads](#)

```

1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

JavaScript

复制到剪贴板

Learn more what Node.js is able to offer with our Learning materials.

v22.19.0 Latest LTS

v24.8.0 Latest Release

商标政策

隐私政策

行为准则

安全政策

© OpenJS Foundation



≡ <https://tailwindcss.com/docs/installation/tailwind-cli>

- npm



```
npm init --yes
```

```
npm install tailwindcss @tailwindcss/cli
```

- src/input.css



```
@import "tailwindcss";
```

- src/index.html



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Tailwind</title>
    <link rel="stylesheet" href="./output.css">
  </head>
  <body>
    <div class="p-10 bg-rose-500">
      <h1 class="text-2xl text-center text-white">tailwind</h1>
    </div>
  </body>
</html>
```

● npx



```
npx @tailwindcss/cli -i ./src/input.css -o ./src/output.css --watch
```

2.3 Next

≡ <https://nextjs.org/>

✓ 内置安装引导支持、仅需创建项目时选中即可

● npm



```
npx create-next-app@latest
```

● console



```
# 项目名称
1. What is your project named? [name]

# 是否安装 TypeScript
2. Would you like to use TypeScript? [No / Yes]

# 是否安装 ESLint 代码检查工具
3. Which linter would you like to use? [ESLint / Biome / None]

# 是否安装 Tailwind
4. Would you like to use Tailwind CSS? No / Yes

# 是否需要 src 作为项目源代码存放目录
5. Would you like your code inside a `src/` directory? [No / Yes]

# 是否安装 文件路由系统
6. Would you like to use App Router? (recommended) [No / Yes]

# 是否安装 Turbopack 打包构建工具
7. Would you like to use Turbopack? (recommended) [No / Yes]

# 是否配置 /* 别名 模块导入方式
8. Would you like to customize the import alias (`/*` by default)? [No / Yes]

# 配置别名 是否使用 /*
9. What import alias would you like configured? /*
```

● page.tsx



```
export default function Page() {
    return (
        <div className="p-10 bg-rose-500">
            <h1 className="text-2xl text-white text-center">tailwind</h1>
        </div>
    )
}
```

2.4 Vite React

≡ 请认真学习本节示例过程、后续示例基于此环境、本环境无须掌握 React

2.4.1 init

≡ npm init --yes

- package.json

```
● ● ●  
{  
  "name": "tailwind",  
  "version": "1.0.0",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "description": "",  
  "type": "module"  
}
```

2.4.2 react

≡ npm install react react-dom

- package.json

```

● ● ●

{
  "name": "tailwind",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "type": "module",
  "dependencies": {
    "react": "^19.1.1",
    "react-dom": "^19.1.1"
  }
}

```

2.4.3 vite

```
≡ npm install vite -D
```

- package.json

```

● ● ●

{
  "name": "tailwind",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "type": "module",
  "dependencies": {
    "react": "^19.1.1",
    "react-dom": "^19.1.1"
  },
  "devDependencies": {
    "vite": "^7.1.6"
  }
}

```

```
    }  
}
```

2.4.4 plugin-react

≡ <https://cn.vite.dev/plugins/>

- ✓ npm install @vitejs/plugin-react -D
- ✓ npm install @vitejs/plugin-react-swc -D [<https://swc.rs/>]

- package.json

● ● ●

```
{  
  "name": "tailwind",  
  "version": "1.0.0",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\"$Error: no test specified\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "description": "",  
  "type": "module",  
  "dependencies": {  
    "react": "^19.1.1",  
    "react-dom": "^19.1.1"  
  },  
  "devDependencies": {  
    "@vitejs/plugin-react-swc": "^4.1.0",  
    "vite": "^7.1.6"  
  }  
}
```

2.4.5 vite.config.js

≡ 项目根目录创建 vite.config.js 文件

- vite.config.js

```
● ● ●  
import { defineConfig } from "vite";  
import react from "@vitejs/plugin-react-swc";  
  
export default defineConfig({  
    plugins: [  
        react(),  
    ]  
})
```

2.4.6 index.html

≡ 项目根目录创建文件 index.html

```
● ● ●  
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>React</title>  
  </head>  
  <body>  
    <div id="root"></div>  
    <script type="module" src="./src/main.jsx"></script>  
  </body>  
</html>
```

2.4.7 src/main.jsx

☰ 项目根目录创建目录及文件 src/app.jsx | src/main.jsx

- src/app.jsx



```
export function App() {
  return (
    <div>react</div>
  )
}
```

- src/main.jsx



```
import { createRoot } from "react-dom/client";
import { App } from "./app.jsx";

const root = document.getElementById("root");

createRoot(root).render(<App/>);
```

- package.json



```
{
  "name": "tailwind",
  "version": "1.0.0",
  "scripts": {
    "dev": "vite dev"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "type": "module",
```

```

"dependencies": {
    "react": "^19.1.1",
    "react-dom": "^19.1.1"
},
"devDependencies": {
    "@vitejs/plugin-react-swc": "^4.1.0",
    "vite": "^7.1.6"
}
}

```

- npm



```
npm run dev # http://localhost:5173/
```

2.4.8 tailwind

```
≡ npm install tailwindcss @tailwindcss/vite -D
```

- package.json



```
{
  "name": "tailwind",
  "version": "1.0.0",
  "scripts": {
    "dev": "vite dev"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "type": "module",
  "dependencies": {
    "react": "^19.1.1",
    "react-dom": "^19.1.1"
  },
  "devDependencies": {
    "@tailwindcss/vite": "^4.1.13",
    "@vitejs/plugin-react-swc": "^4.1.0",
  }
}
```

```
        "tailwindcss": "^4.1.13",
        "vite": "^7.1.6"
    }
}
```

● vite.config.js

```
● ● ●

import { defineConfig } from "vite";
import react from "@vitejs/plugin-react-swc";
import tailwindcss from "@tailwindcss/vite";

export default defineConfig({
    plugins: [
        react(),
        tailwindcss(),
    ]
})
```

● src/globals.css

```
● ● ●

@import "tailwindcss";
```

● src/main.jsx

```
● ● ●

import { createRoot } from "react-dom/client";
import { App } from "./app.jsx";
import "./globals.css";

const root = document.getElementById("root");
createRoot(root).render(<App/>)
```

- **src/app.jsx**

```
● ● ●

export function App() {
  return (
    <div className="p-32 bg-rose-500">
      <h1 className="font-mono text-center text-white">
        <span className="text-4xl md:text-6xl lg:text-7xl duration-300">
          tailwind
        </span>
      </h1>
    </div>
  )
}
```

03. 开发工具

≡ <https://tailwindcss.com/docs/editor-setup>

3.1 IDE

- **ui**

 tailwindcss v4.1

[Documentation](#)

- Components
- Templates
- UI Kit
- Playground
- Course NEW
- Community

[GETTING STARTED](#)

- Installation
- Editor setup**
- Compatibility
- Upgrade guide

[CORE CONCEPTS](#)

- Styling with utility classes
- Hover, focus, and other states
- Responsive design
- Dark mode
- Theme variables
- Colors
- Adding custom styles
- Detecting classes in source files
- Functions and directives

GETTING STARTED

Editor setup

Tooling to improve the developer experience when working with Tailwind CSS.

Syntax support

Tailwind CSS uses custom CSS syntax like `@theme`, `@variant`, and `@source`, and in some editors this can trigger warnings or errors where these rules aren't recognized.

If you're using VS Code, our official [Tailwind CSS IntelliSense](#) plugin includes a dedicated Tailwind CSS language mode that has support for all of the custom at-rules and functions Tailwind uses.

In some cases, you may need to disable native CSS linting/validations if your editor is very strict about the syntax it expects in your CSS files.

IntelliSense for VS Code

The official [Tailwind CSS IntelliSense](#) extension for Visual Studio Code enhances the Tailwind development experience by providing users with advanced features such as autocomplete, syntax highlighting, and linting.



ON THIS PAGE

Syntax support

- IntelliSense for VS Code
- Class sorting with Prettier
- JetBrains IDEs
- Zed

From the creators of Tailwind CSS

Make your ideas look awesome, without relying on a designer.

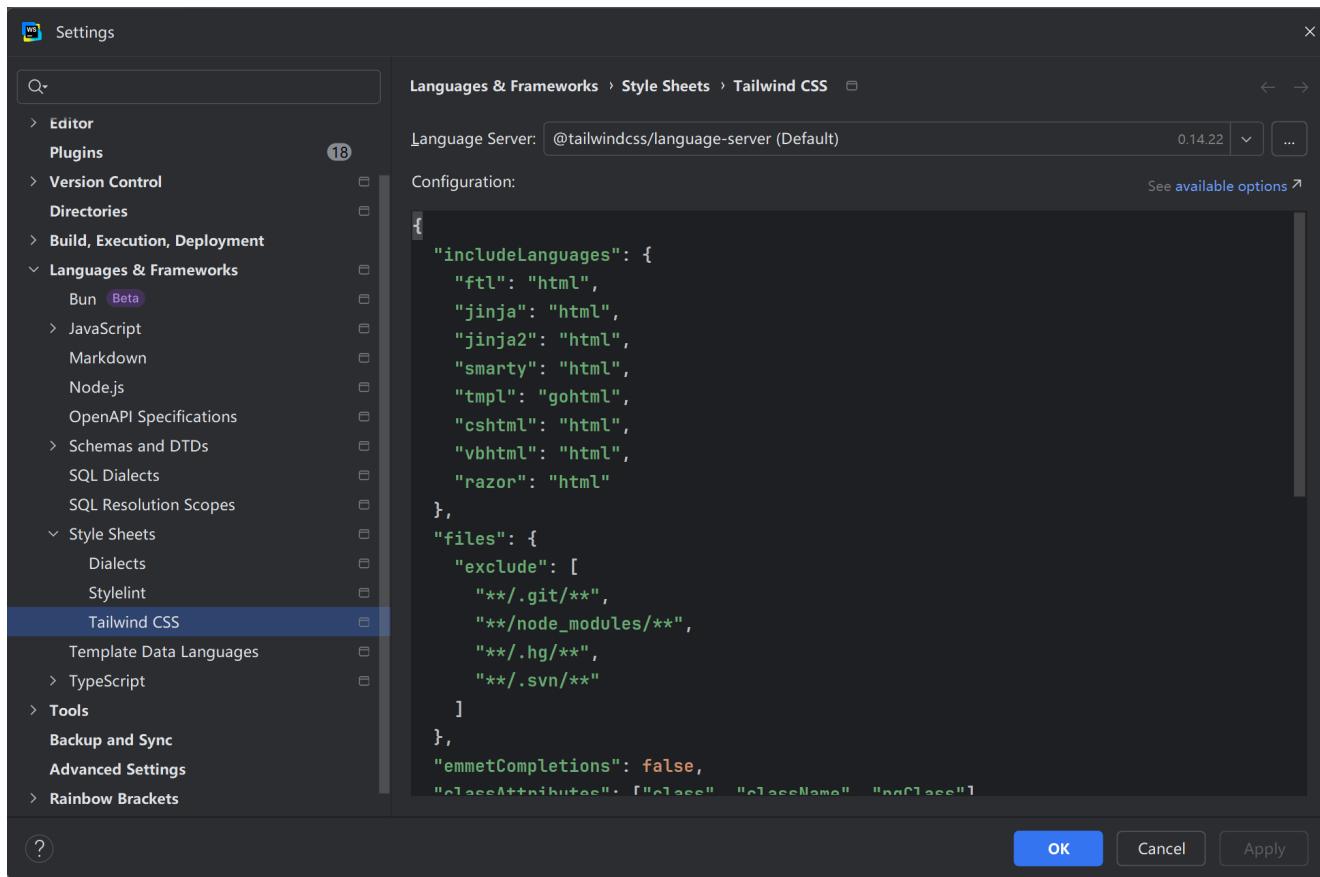
"This is the survival kit I wish I had when I started building apps."

Derrick Reimer, SavvyCal

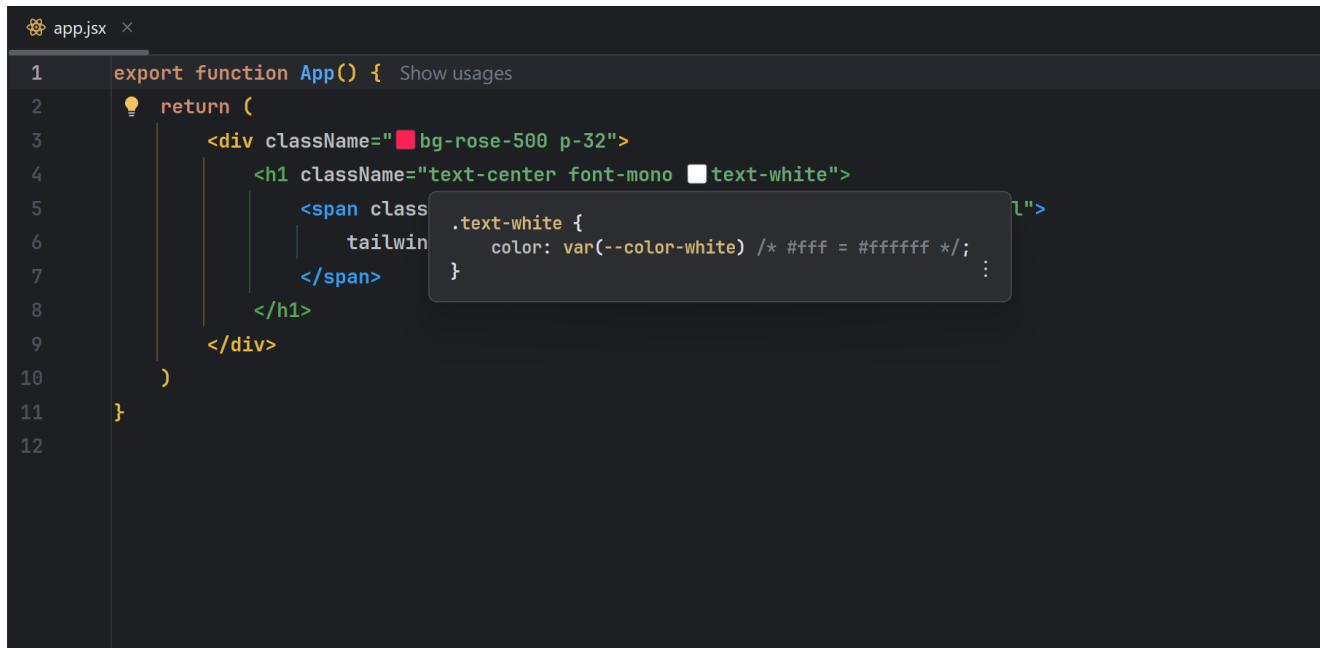
≡ WebStorm 内置 Tailwind Language Server 语言服务引擎

- ✓ 推荐最新版本：<https://www.jetbrains.com/webstorm/download/other.html>
- ✓ 引擎自动开启、无需其它配置、可自动提示 Tailwind、可检测类选择器

● ui



● UI



3.2 格式化

<https://github.com/tailwindlabs/prettier-plugin-tailwindcss>

- ui

A Prettier v3+ plugin for Tailwind CSS v3.0+ that automatically sorts classes based on our recommended class order.

Installation

To get started, install `prettier-plugin-tailwindcss` as a dev-dependency:

```
npm install -D prettier prettier-plugin-tailwindcss
```

Then add the plugin to your [Prettier configuration](#):

```
// .prettierrc
{
  "plugins": ["prettier-plugin-tailwindcss"]
}
```

When using a JavaScript config, you can import the types for IntelliSense:

```
// prettier.config.js
/** @type {import('prettier').Config & import('prettier-plugin-tailwindcss').PluginOptions} */
export default {
  plugins: ["prettier-plugin-tailwindcss"],
}
```

- npm

`npm install prettier prettier-plugin-tailwindcss -D`

- `prettier.config.js`

```
export default {
  plugins: ["prettier-plugin-tailwindcss"],
}
```

- npx

● ● ●

```
npx prettier . --write
```

≡ 推荐人为控制书写顺序 [位置 | 盒子模型 | 排版 | 视觉 | 其它]

- src/app.jsx

● ● ●

```
export function App() {
  return (
    <div className="p-32 bg-slate-200">
      <div className="relative size-48 bg-rose-500">
        <div className="absolute left-1/2 top-1/2 -translate-1/2
          size-1/2
          bg-purple-700
          animate-spin">
        </div>
      </div>
    )
}
```

① Note

✓ <https://codeguide.bootcss.com/>

- ui

声明顺序

相关的属性声明应当归为一组，并按照下面的顺序排列：

1. Positioning
2. Box model
3. Typographic
4. Visual
5. Misc

由于定位（positioning）可以从正常的文档流中移除元素，并且还能覆盖盒模型（box model）相关的样式，因此排在首位。盒模型排在第二位，因为它决定了组件的尺寸和位置。

其他属性只是影响组件的 内部 或者是不影响前两组属性，因此排在后面。

完整的属性列表及其排列顺序请参考 [Bootstrap property order for Stylelint](#)。

```
.declaration-order {
  /* Positioning */
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 100;

  /* Box-model */
  display: block;
  float: right;
  width: 100px;
  height: 100px;

  /* Typography */
  font: normal 13px "Helvetica Neue", sans-serif;
  line-height: 1.5;
  color: #333;
  text-align: center;

  /* Visual */
  background-color: #f5f5f5;
  border: 1px solid #e5e5e5;
  border-radius: 3px;

  /* Misc */
  opacity: 1;
}
```

3.3 兼容性

≡ <https://tailwindcss.com/docs/compatibility>

● ui

GETTING STARTED

Compatibility

Learn about browser support and compatibility with other tooling.

Browser support

Tailwind CSS v4.0 is designed for and tested on modern browsers, and the core functionality of the framework specifically depends on these browser versions:

- = **Chrome 111** (released March 2023)
- = **Safari 16.4** (released March 2023)
- = **Firefox 128** (released July 2024)

Tailwind also includes support for many bleeding-edge platform features like `field-sizing: content`, `@starting-style`, and `text-wrap: balance` that have limited browser support. It's up to you if you want to use these modern features in your projects — if the browsers you're targeting don't support them, simply don't use those utilities and variants.

If you're unsure about the support for a modern platform feature, the [Can I use](#) database is a great resource.

Sass, Less, and Stylus

Tailwind CSS v4.0 is a full-featured CSS build tool designed for a specific workflow, and is not designed to be used with CSS preprocessors like Sass, Less, or Stylus.

ON THIS PAGE

Browser support

- Sass, Less, and Stylus
- Build-time imports
- Variables
- Nesting
- Loops
- Color and math functions
- CSS modules
- Scoping concerns
- Performance
- Explicit context sharing
- Vue, Svelte, and Astro



5-day mini-course

Build UIs that don't suck.

Short, tactical video lessons from the creator of Tailwind CSS, delivered directly to your inbox every day for a week.

[Get the free course →](#)

04. 样式初始化

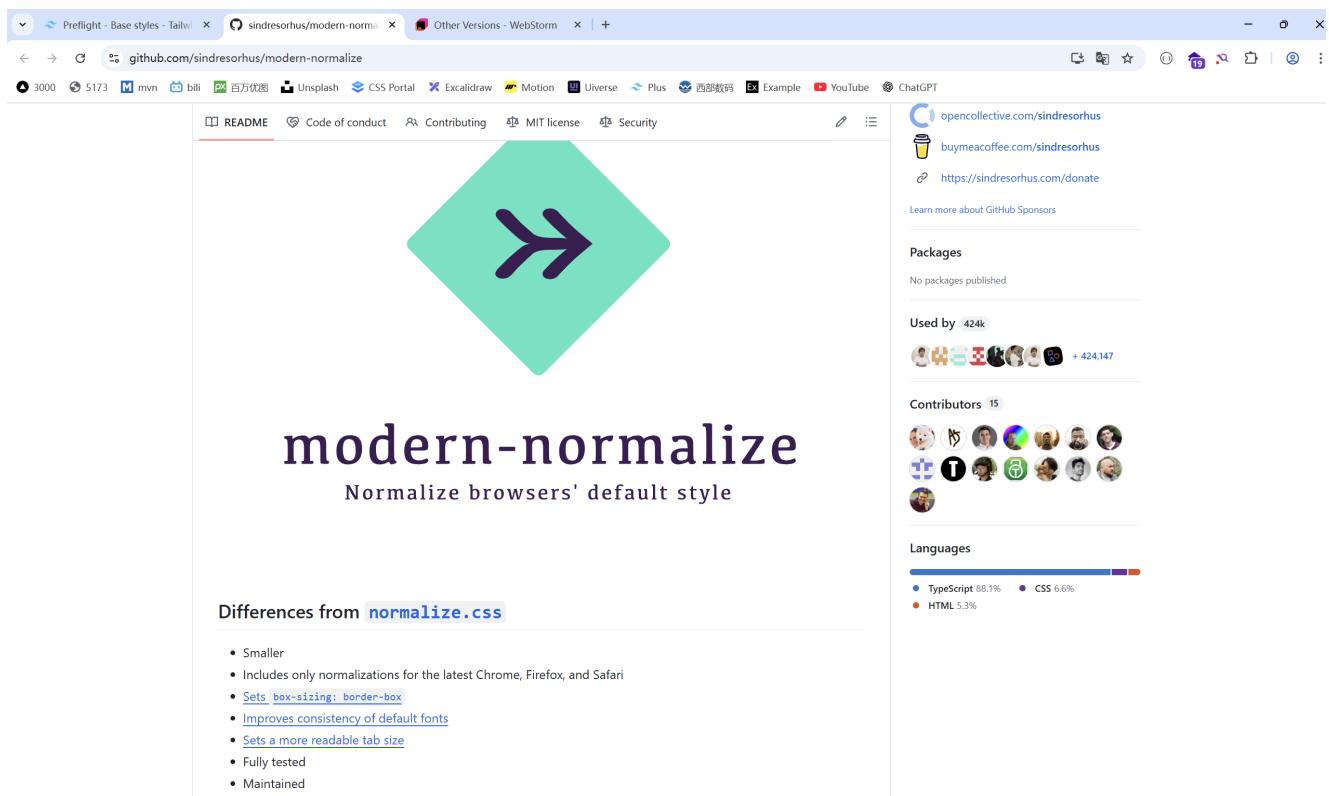
≡ <https://tailwindcss.com/docs/preflight>

4.1 modern-normaliz

≡ <https://github.com/sindresorhus/modern-normalize>

- ✓ Tailwind 使用 modern-normaliz 初始化 HTML 及常用元素样式初始化
- ✓ node_modules/tailwindcss/preflight.css

● ui



- `preflight.css`

```
● ● ●  
*,  
::after,  
::before,  
::backdrop,  
::file-selector-button {  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
    border: 0 solid;  
}  
  
h1,  
h2,  
h3,  
h4,  
h5,  
h6 {  
    font-size: inherit;  
    font-weight: inherit;  
}  
  
img,  
svg,  
video,  
canvas,  
audio,  
iframe,  
embed,  
object {  
    display: block; /* 1 */  
    vertical-align: middle; /* 2 */  
}
```

4.2 @layer

≡ Tailwind 内置 theme | base | components | utilities 分层全局管理样式

- `globals.css`



```
@import "tailwindcss";
```

- **globals.css**



```
@layer theme, base, components, utilities;

@import "tailwindcss/theme.css" layer(theme);
@import "tailwindcss/preflight.css" layer(base);
@import "tailwindcss/utilities.css" layer(utils);
```

4.3 @layer theme

≡ 主题层定义 Tailwind 内置各种全局变量、颜色、动画、间距 等 ...

- **node_modules/tailwindcss/theme.css**



```
@theme default {

  --color-red-50: oklch(97.1% 0.013 17.38);
  --color-red-100: oklch(93.6% 0.032 17.717);
  --color-red-200: oklch(88.5% 0.062 18.334);
  --color-red-300: oklch(80.8% 0.114 19.571);
  --color-red-400: oklch(70.4% 0.191 22.216);
  --color-red-500: oklch(63.7% 0.237 25.331);
  --color-red-600: oklch(57.7% 0.245 27.325);
  --color-red-700: oklch(50.5% 0.213 27.518);
  --color-red-800: oklch(44.4% 0.177 26.899);
  --color-red-900: oklch(39.6% 0.141 25.723);
  --color-red-950: oklch(25.8% 0.092 26.042);

  --blur-xs: 4px;
  --blur-sm: 8px;
  --blur-md: 12px;
  --blur-lg: 16px;
```

```
--blur-xl: 24px;
--blur-2xl: 40px;
--blur-3xl: 64px;

--spacing: 0.25rem; // html 16px * 0.25 ==> 4px
// w-5 ==> width: 20px
// p-5 ==> padding: 20px

@keyframes spin {
  to {
    transform: rotate(360deg);
  }
}

--animate-spin: spin 1s linear infinite;

...
}
```

- node_modules/tailwindcss/index.css



```
@layer theme, base, components, utilities;

@layer theme {
  @theme default {
    ...
    --color-red-100: oklch(93.6% 0.032 17.717);
    --color-red-200: oklch(88.5% 0.062 18.334);
    --color-red-300: oklch(80.8% 0.114 19.571);
    ...
  }
}
```

- globals.css



```
@import "tailwindcss";  
  
@theme {  
  --font-serif: ui-sans-serif, system-ui, sans-serif;  
}
```

4.4 @Layer base

≡ 完成 HTML 样式初始化、全局样式默认内外边距、盒子模型 等 ...

- node_modules/tailwindcss/index.css



```
@layer theme, base, components, utilities;  
  
@layer theme {  
  ...  
}  
  
@layer base {  
  *,  
  ::after,  
  ::before,  
  ::backdrop,  
  ::file-selector-button {  
    box-sizing: border-box; /* 1 */  
    margin: 0; /* 2 */  
    padding: 0; /* 2 */  
    border: 0 solid; /* 3 */  
  }  
  
  ...  
}
```

- `globals.css`

```
● ● ●

@import "tailwindcss";

@layer base {
  h1 {
    font-size: 36px;
  }
}
```

4.5 @layer utilities

≡ utilities 表示内置程序类 p-5 | pt-2 | text-white 等 ... [通过解析生成]

✓ @layer utilities 应当定义 Tailwind 还没有内置的实用程序类

- `node_modules/tailwindcss/index.css`

```
● ● ●

@layer theme, base, components, utilities;

@layer theme {
  ...
}

@layer base {
  *,
  ::after,
  ::before,
  ::backdrop,
  ::file-selector-button {
    box-sizing: border-box; /* 1 */
    margin: 0; /* 2 */
    padding: 0; /* 2 */
    border: 0 solid; /* 3 */
  }
}

...
```

```
@layer utilities {  
    @tailwind utilities;  
}
```

- `node_modules/tailwindcss/utilities.css`



```
@tailwind utilities;
```

- `globals.css`



```
@import "tailwindcss";  
  
@utility animmrate-paused {  
    animation-play-state: paused;  
}  
  
@utility animmrate-delay-1500 {  
    animation-delay: 3000ms;  
}  
  
@utility animmrate-delay-3000 {  
    animation-delay: 3000ms;  
}
```

4.5 @layer components

`@layer components {}` 层是预留给开发者编写的自定义类、在入口文件编写

- `src/globals.css`

```
● ● ●  
import "tailwindcss";  
  
@layer components {  
  .btn {  
    @apply px-3 py-2 bg-rose-500 text-white;  
  }  
  
  .btn-developer {  
    background-color: blue;  
    padding: 10px;  
    color: white;  
  }  
}
```

ⓘ Note

- ✓ `@components` 自定义类选择器、不具有任何特殊性、也不可 `hover:btn` 方式使用
- ✓ `utilities` 视作为 Tailwind 自身的一部分、可 `hover:animmate-paused` 使用

05. 核心机制

≡ <https://tailwindcss.com/docs/styling-with-utility-classes>

- ui

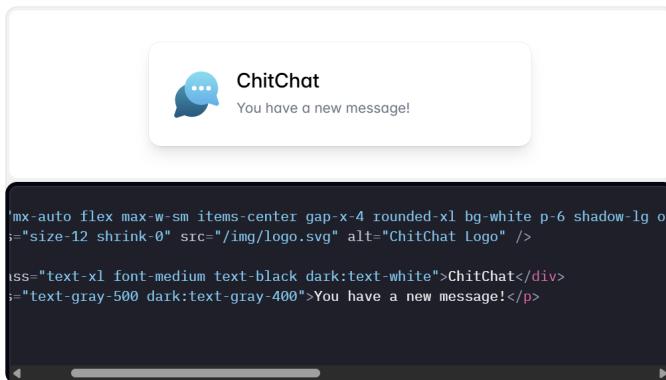
CORE CONCEPTS

Styling with utility classes

Building complex components from a constrained set of primitive utilities.

Overview

You style things with Tailwind by combining many single-purpose presentational classes (*utility classes*) directly in your markup:



For example, in the UI above we've used:

- = The **display** and **padding** utilities ('**flex**', '**shrink-0**', and '**p-6**') to control the overall layout

ON THIS PAGE

Overview

Why not just use inline styles?

Thinking in utility classes

Styling hover and focus states

Media queries and breakpoints

Targeting dark mode

Using class composition

Using arbitrary values

Complex selectors

When to use inline styles

Managing duplication

Using loops

Using multi-cursor editing

Using components

Using custom CSS

Managing style conflicts

Conflicting utility classes

Using the important modifier

Using the important flag

Using the prefix option

4.1 规则

≡ 通常一个 css property 对应一个类选择器、采用 [属性 + 对应值] 方式

- ✓ **p-5** \Rightarrow padding: 20px;
- ✓ **mt-5** \Rightarrow margin-top: 20px;
- ✓ **text-white** \Rightarrow color: white;
- ✓ **bg-white** \Rightarrow background-color: white;

ⓘ Note

- ✓ 原子选择器 有更高的重复利用率、减少打包体积 如 **p-5** 可反复独立使用
- ✓ 原子选择器 避免大量的自定义类选择器增加命名负担、结构控制负担、避免优先级问题
- ✓ 原子选择器 易于开发者之间、社区之间进行样式复制
- ✓ 原子选择器 带有轻微的人为公共语义、极易在开发者之间达成心智共识
- ✓ 内联选择器 属性及值需完整给出、较多属性会使得声明较长
- ✓ 内联选择器 难以肉眼可见理解属性值含义
- ✓ 内联选择器 无法完成 hover 聚焦、媒体查询、等浏览器交互效果

4.1.1 内联样式

≡ Tailwind 主张特殊情况使用内联样式、如条件变量拼接样式、没有内置的实用类

- `src/app.jsx`



```
export function App() {  
  
  const styles = {  
    backgroundImage: "linear-gradient(#fff 1px, transparent 1px), linear-  
    gradient(to right, #fff 1px, transparent 1px)",  
    backgroundSize: "20px 20px",  
    backgroundPosition: "center center"  
  };  
  
  return (  
    <div className="p-32 bg-slate-300">  
      <div className="w-1/2 h-80 bg-purple-700" style={{ ...styles }}>  
        <div className="size-full flex justify-center items-center">  
          <p className="font-mono text-6xl text-white">Tailwind</p>  
        </div>  
      </div>  
    )  
}
```

4.1.2 样式重复

≡ 可采用 循环 | 列编辑 | 抽象组件 等思考控制样式重复

- `src/app.jsx`



```
export function App() {
    return (
        <div className="p-32 bg-slate-300">
            <div className="flex gap-x-5">
                <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">tailwind</button>
                <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">react</button>
                <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">motion</button>
                <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">next</button>
                <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">zustand</button>
            </div>
        </div>
    )
}
```

● src/app.jsx [loop]

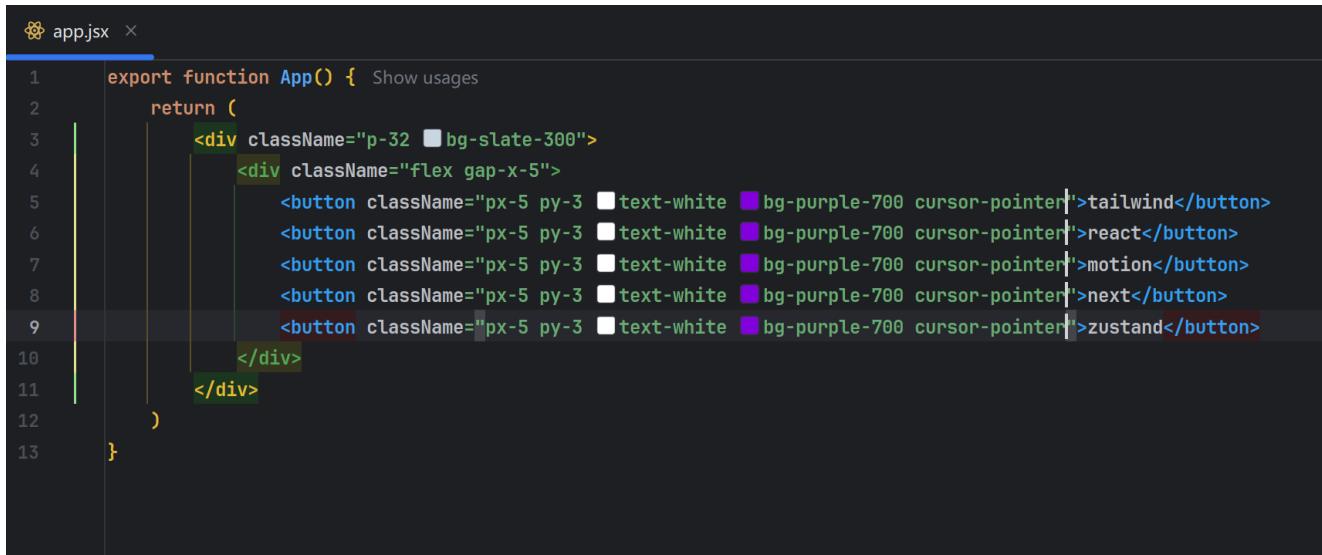


```
export function App() {
    const titles = [
        "tailwind",
        "react",
        "motion",
        "next",
        "zustand",
    ];
    return (
        <div className="p-32 bg-slate-300">
            <div className="flex gap-x-5">
                {
                    titles.map((title) => (
                        <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer"
                            key={ title }
                            { title }
                        </button>
                    ))
                }
            </div>
        </div>
    )
}
```

}

≡ 列编辑 [ALT + SHIFT + ALT]

- ui



```

1  export function App() { Show usages
2
3      return (
4          <div className="p-32 bg-slate-300">
5              <div className="flex gap-x-5">
6                  <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">tailwind</button>
7                  <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">react</button>
8                  <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">motion</button>
9                  <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">next</button>
10                 <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">zustand</button>
11             </div>
12         )
13     }

```

≡ 抽象组件

- src/app.jsx



```

function Button({ title }) {
    return (
        <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">
            { title }
        </button>
    )
}

export function App() {
    return (
        <div className="p-32 bg-slate-300">
            <div className="flex gap-x-5">
                <Button title={"tailwind"} />
                <Button title={"react"} />
            </div>
        </div>
    )
}

```

```

        <Button title={"motion"} />
        <Button title={"next"} />
        <Button title={"zustand"} />
    </div>
</div>
)
}

```

● src/app.jsx

```

● ● ●

function Button({ title }) {
    return (
        <button className="px-5 py-3 text-white bg-purple-700 cursor-pointer">
            { title }
        </button>
    )
}

export function App() {

    const titles = [
        "tailwind",
        "react",
        "motion",
        "next",
        "zustand",
    ];

    return (
        <div className="p-32 bg-slate-300">
            <div className="flex gap-x-5">
                {
                    titles.map((title) => (
                        <Button title={ title } key={ title } />
                    ))
                }
            </div>
        </div>
    )
}

```

06. 盒子大小

07. 背景

08. 间距

09. 弹性网格

10. 布局

11. 排版

12. 边框

13. 效果

14. 移动

15. 过渡动画

16. 互动

17. 表格

18. 作用

19. SVG
