

TAILWIND CSS

Utility-First

Functions & Directives

- @tailwind 用来插入 tailwinds 的 base, components, utilities, screen 样式。
- @apply 在 css 中 inline 集成已存在的 utility classes
- @layer
 - 表明自定义的样式属于
 - base
 - components
 - utilities
 - @layer 里面的样式也会被 purge 进行优化处理
- @variants 生成 hover, focus, 等 variants
- @responsive
- @screen sm {
/* ... */
}
创建媒体查询，可以引用断点名称。
- screen()
 - 把 tailwind 的 breakpoint 转换为传统的 media query
 - 和一些不理解 @screen 的其他工具配合使用
- theme()
 - 用来访问定义在 config 里面的值

Cons

- 无需为 class 命名
- 几乎不需要写 custom css, 避免 css 文件膨胀
- CSS 有可能是全局的，但是 Template 所有的 class 一定是 local 的。对 local 进行修改更加安全。

Why not use inline styles?

- 设计有约束，因为设计理念可以很容易的体现在 Utility 中，避免零散的 Magic Number
- Responsive Design
 - inline style 不能进行媒体查询
- Hover, focus, 和其他 state 可以很容易的进行 styling

Maintainability

- 对于一组特定的 tailwind className 的组合，如何进行复用？

抽取为 Template

```
<!-- Using utilities -->
<button class="py-2 px-4 font-semibold rounded-lg shadow-md text-white bg-green-500 hover:bg-g
Click me
</button>

<!-- Extracting classes using @apply -->
<button class="btn btn-green">
  Button
</button>

<style>
  .btn {
    @apply py-2 px-4 font-semibold rounded-lg shadow-md;
  }
  .btn-green {
    @apply text-white bg-green-500 hover:bg-green-700;
  }
</style>
```

在 CSS 中使用 @apply

Responsive Design

Every utility class can be applied conditionally at different breakpoints.

- sm: 640px @media (min-width: 640px) { ... }
- md: 768px @media (min-width: 768px) { ... }
- lg: 1024px @media (min-width: 1024px) { ... }
- xl: 1280px @media (min-width: 1280px) { ... }
- 2xl: 1536px @media (min-width: 1536px) { ... }

```
// tailwind.config.js
module.exports = {
  theme: {
    screens: {
      'tablet': '640px',
      // => @media (min-width: 640px) { ... }

      'laptop': '1024px',
      // => @media (min-width: 1024px) { ... }

      'desktop': '1280px',
      // => @media (min-width: 1280px) { ... }
    },
  },
}
```

customizing breakpoints

这意味着无前缀的工具类应用到全部尺寸的屏幕上，一旦添加了 md:uppercase 这样的工具类，其会对 md 这个断点以及其上的断点起作用。

Mobile First

Use unprefixd utilities to target mobile, and override them at larger breakpoints

Adding New Utilities

- 使用 @layer utilitis 来在 css 中扩展 utility class
- 使用 @variants responsive 生成支持 breakpoint 前缀的 class
- 使用 @vairants dark 生成支持 dark mode 的 class
- 使用 @variants + state name 生成支持 state 的 utility class
- 使用 addUtilities 在 config 中进行配置。方便跨 project 复用以及作为 lib 来发布这些 custom utilities

Extracting Components

- 除了抽取 template / 组件 / 在 css 中使用 @apply 进行组合复用以外
- 还可以在 config 中使用 addComponents 直接定义一组组合好的 class
- good for lib / share class across multi projects

Hover, Focus, and other States.

- Built-in State Variants
 - 出于文件体积的考虑，tailwindcss 并没有支持所有的 state virants
 - 如果你需要使用不支持的 state，可以编写 variant plugin
 - hover:
 - focus:
 - active:
 - group-hover
 - group 的目的都是一样的，在对 parent 施加 state 的时候对 child 也感觉是相反的
 - group-focus
 - child focus 的时候，对上级元素进行样式改变
 - focus-within
 - focus-visible
 - 使用 keyboard 进行 focus 的时候生效，需要 PostCSS 和 JS polyfills
 - Motion-safe
 - https://developer.mozilla.org/en-US/docs/Web/CSS/@media/prefers-reduced-motion
 - Motion-reduce
 - Disabled
 - Visited
 - Checked
 - first: First-child
 - last: Last-child
 - odd: Odd-child
 - even: Even-child
- add it to the child element, not the parent element

Adding Base Styles

- Preflight 的默认行为
 - 基于 modern-normalize 只是提供了一层薄薄的样式 用于浏览器的样式规范化
 - remove default margins
 - h1~h6 的字体，字重被重置为 inherit
 - List 样式被移除
 - Images / video / svg / iframe 等 display 为 block
 - 全局元素的 border style 被重置
 - buttons add default outline
 - config: corePlugins --> preflight: false
 - disable preflight
- Preflight: 一组开箱即用的样式，用于对基本的 html 元素进行样式化。
- 应用 base style 的几种方式
 - 在 css 中使用 @layer base
 - 直接在 html 上添加相应的 classes
 - 在 config 文件中使用 addBase function

Dark Mode Support

- dark: prefix
- not enabled by default
- dark mode strategy
 - use prefers-color-scheme media
 - 通过在 html 上添加 dark class 来手动切换 class
- color-related classes scope: