Complete TailwindCSS Cheatsheet - Beginner to Advanced

Table of Contents

Getting Started

Installation Methods

Via CDN (Quick Start)

```html
<script src="https://cdn.tailwindcss.com"></script>
```

Via npm/yarn

```
npm install -D tailwindcss
npx tailwindcss init
```

With PostCSS

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

Basic Configuration (tailwind.config.js)

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{html,js,jsx,ts,tsx}"],
  theme: {
    extend: {
      colors: {
        "custom-blue": "#1fb6ff",
      },
    },
  },
  plugins: [],
};
```

Core Concepts

Utility-First Philosophy

- Every CSS property as a utility class
- Compose complex designs from simple utilities
- No need to write custom CSS

Basic Syntax Pattern

```
{property}-{value}
{property}-{breakpoint}-{value}
{property}-{state}-{value}
```

Examples

```
<!-- Padding -->
<div class="p-4">
  <!-- padding: 1rem -->
  <div class="px-4 py-2">
    <!-- px: 1rem, py: 0.5rem -->

    <!-- Colors -->
    <div class="bg-blue-500 text-white">
      <!-- Responsive -->
      <div class="w-full md:w-1/2 lg:w-1/3"></div>
    </div>
  </div>
</div>
```

Layout & Positioning

Display

```html
<!-- Display Types -->
<div class="block">
  <!-- display: block -->
  <div class="inline-block">
    <!-- display: inline-block -->
    <div class="inline">
      <!-- display: inline -->
      <div class="flex">
        <!-- display: flex -->
        <div class="grid">
          <!-- display: grid -->
          <div class="hidden">
            <!-- display: none -->

            <!-- Table Display -->
            <div class="table">
              <div class="table-row">
                <div class="table-cell"></div>
              </div></div></div></div
      ></div>
    </div>
  </div>
</div>
```

Position

```html
<div class="static">
  <!-- position: static -->
  <div class="relative">
    <!-- position: relative -->
    <div class="absolute">
      <!-- position: absolute -->
      <div class="fixed">
        <!-- position: fixed -->
        <div class="sticky">
          <!-- position: sticky -->

          <!-- Positioning with offsets -->
          <div class="absolute top-0 right-0">
            <div class="fixed bottom-4 left-4">
              <div class="relative -top-2 -left-1"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Z-Index

```html
<div class="z-0">
  <!-- z-index: 0 -->
  <div class="z-10">
    <!-- z-index: 10 -->
    <div class="z-20">
      <!-- z-index: 20 -->
      <div class="z-50">
        <!-- z-index: 50 -->
        <div class="z-auto"><!-- z-index: auto --></div>
      </div>
    </div>
  </div>
</div>
```

Float & Clear

```html
<div class="float-left">
  <div class="float-right">
    <div class="float-none">
      <div class="clear-both"></div>
    </div>
  </div>
</div>
```

Typography

Font Family

```html
<p class="font-sans"><!-- font-family: ui-sans-serif, system-ui, ... --></p>
<p class="font-serif"><!-- font-family: ui-serif, Georgia, ... --></p>
<p class="font-mono"><!-- font-family: ui-monospace, SFMono-Regular, ... --></p>
```

Font Size

```html
<p class="text-xs"><!-- font-size: 0.75rem --></p>
<p class="text-sm"><!-- font-size: 0.875rem --></p>
<p class="text-base"><!-- font-size: 1rem --></p>
<p class="text-lg"><!-- font-size: 1.125rem --></p>
<p class="text-xl"><!-- font-size: 1.25rem --></p>
<p class="text-2xl"><!-- font-size: 1.5rem --></p>
```

```html
<p class="text-3xl"><!-- font-size: 1.875rem --></p>
<p class="text-4xl"><!-- font-size: 2.25rem --></p>
<p class="text-5xl"><!-- font-size: 3rem --></p>
<p class="text-6xl"><!-- font-size: 3.75rem --></p>
<p class="text-7xl"><!-- font-size: 4.5rem --></p>
<p class="text-8xl"><!-- font-size: 6rem --></p>
<p class="text-9xl"><!-- font-size: 8rem --></p>
```

Font Weight

```html
<p class="font-thin"><!-- font-weight: 100 --></p>
<p class="font-extralight"><!-- font-weight: 200 --></p>
<p class="font-light"><!-- font-weight: 300 --></p>
<p class="font-normal"><!-- font-weight: 400 --></p>
<p class="font-medium"><!-- font-weight: 500 --></p>
<p class="font-semibold"><!-- font-weight: 600 --></p>
<p class="font-bold"><!-- font-weight: 700 --></p>
<p class="font-extrabold"><!-- font-weight: 800 --></p>
<p class="font-black"><!-- font-weight: 900 --></p>
```

Text Alignment & Decoration

```html
<!-- Alignment -->
<p class="text-left"></p>
<p class="text-center"></p>
<p class="text-right"></p>
<p class="text-justify">
  <!-- Decoration -->
</p>

<p class="underline"></p>
<p class="line-through"></p>
<p class="no-underline">
  <!-- Transform -->
</p>

<p class="uppercase"></p>
<p class="lowercase"></p>
<p class="capitalize"></p>
<p class="normal-case"></p>
```

Line Height & Letter Spacing

```html
<!-- Line Height -->
<p class="leading-none"><!-- line-height: 1 --></p>
<p class="leading-tight"><!-- line-height: 1.25 --></p>
```

```
<p class="leading-normal"><!-- line-height: 1.5 --></p>
<p class="leading-relaxed"><!-- line-height: 1.625 --></p>
<p class="leading-loose">
  <!-- line-height: 2 -->

  <!-- Letter Spacing -->
</p>

<p class="tracking-tighter"><!-- letter-spacing: -0.05em --></p>
<p class="tracking-tight"><!-- letter-spacing: -0.025em --></p>
<p class="tracking-normal"><!-- letter-spacing: 0em --></p>
<p class="tracking-wide"><!-- letter-spacing: 0.025em --></p>
<p class="tracking-wider"><!-- letter-spacing: 0.05em --></p>
<p class="tracking-widest"><!-- letter-spacing: 0.1em --></p>
```

Colors & Backgrounds

Text Colors

```
<!-- Standard Colors -->
<p class="text-black"></p>
<p class="text-white"></p>
<p class="text-gray-500"></p>
<p class="text-red-500"></p>
<p class="text-blue-500"></p>
<p class="text-green-500"></p>
<p class="text-yellow-500"></p>
<p class="text-purple-500"></p>
<p class="text-pink-500"></p>
<p class="text-indigo-500">
  <!-- Color Shades (50-950) -->
</p>

<p class="text-blue-50"><!-- Lightest --></p>
<p class="text-blue-100"></p>
<p class="text-blue-200"></p>
<p class="text-blue-300"></p>
<p class="text-blue-400"></p>
<p class="text-blue-500"><!-- Default --></p>
<p class="text-blue-600"></p>
<p class="text-blue-700"></p>
<p class="text-blue-800"></p>
<p class="text-blue-900"></p>
<p class="text-blue-950"><!-- Darkest --></p>
```

Background Colors

```
<div class="bg-white">
  <div class="bg-gray-100">
    <div class="bg-blue-500">
      <div class="bg-gradient-to-r from-blue-500 to-purple-600"></div>
    </div>
  </div>
</div>
```

Background Images & Gradients

```
<!-- Background Size -->
<div class="bg-auto">
  <div class="bg-cover">
    <div class="bg-contain">
      <!-- Background Position -->
      <div class="bg-center">
        <div class="bg-top">
          <div class="bg-bottom">
            <div class="bg-left">
              <div class="bg-right">
                <!-- Gradients -->
                <div
                  class="bg-gradient-to-r from-purple-400 via-pink-500 to-red-500"
                >
                  <div
                    class="bg-gradient-to-br from-green-400 to-blue-600"
                  ></div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Opacity

```
<div class="opacity-0">
  <!-- opacity: 0 -->
  <div class="opacity-25">
    <!-- opacity: 0.25 -->
    <div class="opacity-50">
      <!-- opacity: 0.5 -->
      <div class="opacity-75">
        <!-- opacity: 0.75 -->
        <div class="opacity-100"><!-- opacity: 1 --></div>
        </div>
      </div>
    </div>
  </div>
```

Spacing & Sizing

Padding & Margin

```
<!-- All sides -->
<div class="p-4">
  <!-- padding: 1rem -->
  <div class="m-4">
    <!-- margin: 1rem -->

    <!-- Specific sides -->
    <div class="pt-4">
      <!-- padding-top: 1rem -->
      <div class="pr-4">
        <!-- padding-right: 1rem -->
        <div class="pb-4">
          <!-- padding-bottom: 1rem -->
          <div class="pl-4">
            <!-- padding-left: 1rem -->

            <!-- Horizontal/Vertical -->
            <div class="px-4">
              <!-- padding-left + padding-right -->
              <div class="py-4">
                <!-- padding-top + padding-bottom -->
                <div class="mx-4">
                  <!-- margin-left + margin-right -->
                  <div class="my-4">
                    <!-- margin-top + margin-bottom -->

                    <!-- Negative margins -->
                    <div class="-m-4">
                      <div class="-mt-2">
                        <!-- Auto margins -->
                        <div class="mx-auto"><!-- margin: 0 auto --></div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  </div>
</div>
```

Spacing Scale

```
0    → 0px
px   → 1px
0.5  → 0.125rem (2px)
1    → 0.25rem (4px)
1.5  → 0.375rem (6px)
2    → 0.5rem (8px)
2.5  → 0.625rem (10px)
3    → 0.75rem (12px)
3.5  → 0.875rem (14px)
4    → 1rem (16px)
5    → 1.25rem (20px)
6    → 1.5rem (24px)
7    → 1.75rem (28px)
8    → 2rem (32px)
9    → 2.25rem (36px)
10   → 2.5rem (40px)
11   → 2.75rem (44px)
12   → 3rem (48px)
14   → 3.5rem (56px)
16   → 4rem (64px)
20   → 5rem (80px)
24   → 6rem (96px)
28   → 7rem (112px)
32   → 8rem (128px)
36   → 9rem (144px)
40   → 10rem (160px)
44   → 11rem (176px)
48   → 12rem (192px)
52   → 13rem (208px)
56   → 14rem (224px)
60   → 15rem (240px)
64   → 16rem (256px)
72   → 18rem (288px)
80   → 20rem (320px)
96   → 24rem (384px)
```

Width & Height

```
<!-- Fixed sizes -->
<div class="w-0">
  <!-- width: 0px -->
  <div class="w-px">
    <!-- width: 1px -->
    <div class="w-1">
```

```
        <!-- width: 0.25rem -->
        <div class="w-96">
          <!-- width: 24rem -->

          <!-- Fractional widths -->
          <div class="w-1/2">
            <!-- width: 50% -->
            <div class="w-1/3">
              <!-- width: 33.333333% -->
              <div class="w-2/3">
                <!-- width: 66.666667% -->
                <div class="w-1/4">
                  <!-- width: 25% -->
                  <div class="w-3/4">
                    <!-- width: 75% -->
                    <div class="w-1/5">
                      <!-- width: 20% -->
                      <div class="w-1/6">
                        <!-- width: 16.666667% -->
                        <div class="w-1/12">
                          <!-- width: 8.333333% -->

                          <!-- Full sizes -->
                          <div class="w-full">
                            <!-- width: 100% -->
                            <div class="w-screen">
                              <!-- width: 100vw -->

                              <!-- Min/Max widths -->
                              <div class="min-w-0">
                                <div class="min-w-full">
                                  <div class="max-w-xs">
                                    <div class="max-w-sm">
                                      <div class="max-w-md">
                                        <div class="max-w-lg">
                                          <div class="max-w-xl">
                                            <div class="max-w-2xl">
                                              <div class="max-w-4xl">
                                                <div class="max-w-6xl">
                                                  <div class="max-w-full"></div>
                                                </div>
                                              </div>
                                            </div>
                                          </div>
                                        </div>
                                      </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Flexbox & Grid

Flexbox Container

```
<div class="flex">
  <!-- display: flex -->
  <div class="inline-flex">
    <!-- display: inline-flex -->

    <!-- Direction -->
    <div class="flex-row">
      <!-- flex-direction: row -->
      <div class="flex-row-reverse">
        <!-- flex-direction: row-reverse -->
        <div class="flex-col">
          <!-- flex-direction: column -->
          <div class="flex-col-reverse">
            <!-- flex-direction: column-reverse -->

            <!-- Wrap -->
            <div class="flex-wrap">
              <!-- flex-wrap: wrap -->
              <div class="flex-wrap-reverse">
                <!-- flex-wrap: wrap-reverse -->
                <div class="flex-nowrap"><!-- flex-wrap: nowrap --></div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div></div
  >
</div>
```

Flexbox Alignment

```
<!-- Justify Content (main axis) -->
<div class="justify-start">
  <!-- justify-content: flex-start -->
  <div class="justify-end">
    <!-- justify-content: flex-end -->
    <div class="justify-center">
      <!-- justify-content: center -->
      <div class="justify-between">
        <!-- justify-content: space-between -->
        <div class="justify-around">
          <!-- justify-content: space-around -->
          <div class="justify-evenly">
            <!-- justify-content: space-evenly -->

            <!-- Align Items (cross axis) -->
            <div class="items-start">
              <!-- align-items: flex-start -->
              <div class="items-end">
                <!-- align-items: flex-end -->
                <div class="items-center">
                  <!-- align-items: center -->
                  <div class="items-baseline">
                    <!-- align-items: baseline -->
                    <div class="items-stretch">
                      <!-- align-items: stretch -->

                      <!-- Align Content -->
                      <div class="content-center">
                        <!-- align-content: center -->
                        <div class="content-start">
                          <!-- align-content: flex-start -->
                          <div class="content-end">
                            <!-- align-content: flex-end -->
                            <div class="content-between">
                              <!-- align-content: space-between -->
                              <div class="content-around">
                                <!-- align-content: space-around -->
                                <div class="content-evenly">
                                  <!-- align-content: space-evenly -->
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Flexbox Items

```
<!-- Flex Grow/Shrink -->
<div class="flex-1">
  <!-- flex: 1 1 0% -->
  <div class="flex-auto">
    <!-- flex: 1 1 auto -->
    <div class="flex-initial">
      <!-- flex: 0 1 auto -->
      <div class="flex-none">
        <!-- flex: none -->

        <!-- Align Self -->
        <div class="self-auto">
          <!-- align-self: auto -->
          <div class="self-start">
            <!-- align-self: flex-start -->
            <div class="self-end">
              <!-- align-self: flex-end -->
              <div class="self-center">
                <!-- align-self: center -->
                <div class="self-stretch">
                  <!-- align-self: stretch -->
                  <div class="self-baseline">
                    <!-- align-self: baseline -->

                    <!-- Order -->
                    <div class="order-1">
                      <!-- order: 1 -->
                      <div class="order-2">
                        <!-- order: 2 -->
                        <div class="order-last">
                          <!-- order: 9999 -->
                          <div class="order-none"><!-- order: 0 --></div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

CSS Grid

```html
<!-- Grid Container -->
<div class="grid">
  <!-- display: grid -->
  <div class="inline-grid">
    <!-- display: inline-grid -->

    <!-- Grid Template Columns -->
    <div class="grid-cols-1">
      <!-- grid-template-columns: repeat(1, minmax(0, 1fr)) -->
      <div class="grid-cols-2">
        <!-- grid-template-columns: repeat(2, minmax(0, 1fr)) -->
        <div class="grid-cols-3">
          <!-- grid-template-columns: repeat(3, minmax(0, 1fr)) -->
          <div class="grid-cols-4">
            <!-- grid-template-columns: repeat(4, minmax(0, 1fr)) -->
            <div class="grid-cols-6">
              <!-- grid-template-columns: repeat(6, minmax(0, 1fr)) -->
              <div class="grid-cols-12">
                <!-- grid-template-columns: repeat(12, minmax(0, 1fr)) -->

                <!-- Grid Template Rows -->
                <div class="grid-rows-1">
                  <!-- grid-template-rows: repeat(1, minmax(0, 1fr)) -->
                  <div class="grid-rows-2">
                    <!-- grid-template-rows: repeat(2, minmax(0, 1fr)) -->
                    <div class="grid-rows-3">
                      <!-- grid-template-rows: repeat(3, minmax(0, 1fr)) -->

                      <!-- Grid Column Span -->
                      <div class="col-span-1">
                        <!-- grid-column: span 1 / span 1 -->
                        <div class="col-span-2">
                          <!-- grid-column: span 2 / span 2 -->
                          <div class="col-span-full">
                            <!-- grid-column: 1 / -1 -->

                            <!-- Grid Row Span -->
                            <div class="row-span-1">
                              <!-- grid-row: span 1 / span 1 -->
                              <div class="row-span-2">
                                <!-- grid-row: span 2 / span 2 -->
                                <div class="row-span-full">
                                  <!-- grid-row: 1 / -1 -->

                                  <!-- Grid Column Start/End -->
                                  <div class="col-start-1">
                                    <!-- grid-column-start: 1 -->
                                    <div class="col-start-2">
                                      <!-- grid-column-start: 2 -->
                                      <div class="col-end-3">
                                        <!-- grid-column-end: 3 -->
                                        <div class="col-end-4">
                                          <!-- grid-column-end: 4 -->
                                        </div>
                                      </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div></div
  >
</div>
```

Grid Gaps

```
<div class="gap-0">
  <!-- gap: 0px -->
  <div class="gap-1">
    <!-- gap: 0.25rem -->
    <div class="gap-2">
      <!-- gap: 0.5rem -->
      <div class="gap-4">
        <!-- gap: 1rem -->
        <div class="gap-8">
          <!-- gap: 2rem -->

          <!-- Separate row/column gaps -->
          <div class="gap-x-4">
            <!-- column-gap: 1rem -->
            <div class="gap-y-2"><!-- row-gap: 0.5rem --></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Responsive Design

Breakpoints

```
sm   → @media (min-width: 640px)
md   → @media (min-width: 768px)
lg   → @media (min-width: 1024px)
xl   → @media (min-width: 1280px)
2xl  → @media (min-width: 1536px)
```

Responsive Utilities

```html
<!-- Mobile First Approach -->
<div class="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-1/5">
  <!-- Hide/Show at breakpoints -->
  <div class="block md:hidden">
    <!-- Show on mobile, hide on desktop -->
    <div class="hidden md:block">
      <!-- Hide on mobile, show on desktop -->

      <!-- Responsive text sizes -->
      <h1 class="text-2xl md:text-4xl lg:text-6xl">
        <!-- Responsive spacing -->
        <div class="p-4 md:p-8 lg:p-12">
          <!-- Responsive flexbox -->
          <div class="flex-col md:flex-row"></div>
        </div>
      </h1>
    </div>
  </div>
</div>
```

Container

```html
<div class="container">
  <!-- Responsive container with max-widths -->
  <div class="container mx-auto"><!-- Centered container --></div>
</div>
```

Container breakpoints:

- `sm` (640px): max-width: 640px
- `md` (768px): max-width: 768px
- `lg` (1024px): max-width: 1024px
- `xl` (1280px): max-width: 1280px
- `2xl` (1536px): max-width: 1536px

Pseudo-classes & States

Hover & Focus

```
<button class="bg-blue-500 hover:bg-blue-700">
  <input
    class="border focus:border-blue-500 focus:ring-2 focus:ring-blue-200"
  />
  <a class="text-blue-500 hover:text-blue-700 hover:underline"></a>
</button>
```

Active & Visited

```
<button class="bg-blue-500 active:bg-blue-800">
  <a class="text-blue-500 visited:text-purple-500"></a>
</button>
```

First/Last Child

```
<div class="first:mt-0">
  <!-- :first-child -->
  <div class="last:mb-0">
    <!-- :last-child -->
    <div class="odd:bg-gray-100">
      <!-- :nth-child(odd) -->
      <div class="even:bg-gray-200"><!-- :nth-child(even) --></div>
    </div>
  </div>
</div>
```

Disabled & Required

```
<input class="disabled:opacity-50 disabled:cursor-not-allowed" />
<input class="required:border-red-500" />
```

Group Hover

```
<div class="group">
  <img class="group-hover:opacity-75" />
  <p class="group-hover:text-blue-500"></p>
</div>
```

Focus Within

```html
<div class="focus-within:ring-2 focus-within:ring-blue-500">
  <input class="..." />
</div>
```

Animations & Transitions

Transitions

```html
<div class="transition">
  <!-- transition-property: all -->
  <div class="transition-colors">
    <!-- transition-property: color, background-color, border-color -->
    <div class="transition-opacity">
      <!-- transition-property: opacity -->
      <div class="transition-shadow">
        <!-- transition-property: box-shadow -->
        <div class="transition-transform">
          <!-- transition-property: transform -->

          <!-- Duration -->
          <div class="duration-75">
            <!-- transition-duration: 75ms -->
            <div class="duration-100">
              <!-- transition-duration: 100ms -->
              <div class="duration-150">
                <!-- transition-duration: 150ms -->
                <div class="duration-200">
                  <!-- transition-duration: 200ms -->
                  <div class="duration-300">
                    <!-- transition-duration: 300ms -->
                    <div class="duration-500">
                      <!-- transition-duration: 500ms -->
                      <div class="duration-700">
                        <!-- transition-duration: 700ms -->
                        <div class="duration-1000">
                          <!-- transition-duration: 1000ms -->

                          <!-- Timing Function -->
                          <div class="ease-linear">
                            <!-- transition-timing-function: linear -->
                            <div class="ease-in">
                              <!-- transition-timing-function: cubic-bezier(0.4, 0, 1, 1) -->
                              <div class="ease-out">
                                <!-- transition-timing-function: cubic-bezier(0, 0, 0.2, 1) -->
                                <div class="ease-in-out">
                                  <!-- transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1) -->

                                  <!-- Delay -->
                                  <div class="delay-75">
                                    <!-- transition-delay: 75ms -->
                                    <div class="delay-100">
                                      <!-- transition-delay: 100ms -->
                                      <div class="delay-150">
                                        <!-- transition-delay: 150ms -->
                                      </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Transforms

```
<!-- Scale -->
<div class="scale-0">
  <!-- transform: scale(0) -->
  <div class="scale-50">
    <!-- transform: scale(.5) -->
    <div class="scale-75">
      <!-- transform: scale(.75) -->
      <div class="scale-90">
        <!-- transform: scale(.9) -->
        <div class="scale-95">
          <!-- transform: scale(.95) -->
          <div class="scale-100">
            <!-- transform: scale(1) -->
            <div class="scale-105">
              <!-- transform: scale(1.05) -->
              <div class="scale-110">
                <!-- transform: scale(1.1) -->
                <div class="scale-125">
                  <!-- transform: scale(1.25) -->
                  <div class="scale-150">
                    <!-- transform: scale(1.5) -->

                    <!-- Rotate -->
                    <div class="rotate-0">
                      <!-- transform: rotate(0deg) -->
                      <div class="rotate-1">
                        <!-- transform: rotate(1deg) -->
                        <div class="rotate-2">
                          <!-- transform: rotate(2deg) -->
                          <div class="rotate-3">
                            <!-- transform: rotate(3deg) -->
                            <div class="rotate-6">
                              <!-- transform: rotate(6deg) -->
                              <div class="rotate-12">
                                <!-- transform: rotate(12deg) -->
                                <div class="rotate-45">
                                  <!-- transform: rotate(45deg) -->
                                  <div class="rotate-90">
                                    <!-- transform: rotate(90deg) -->
                                    <div class="rotate-180">
                                      <!-- transform: rotate(180deg) -->

                                      <!-- Translate -->
                                      <div class="translate-x-0">
                                        <!-- transform: translateX(0px) -->
                                        <div class="translate-x-1">
                                          <!-- transform: translateX(0.25rem) -->
                                          <div class="translate-x-2">
                                            <!-- transform: translateX(0.5rem) -->
                                            <div class="translate-y-0">
                                              <!-- transform: translateY(0px) -->
                                              <div class="translate-y-1">
                                                <!-- transform:
translateY(0.25rem) -->

                                                <!-- Skew -->
                                                <div class="skew-x-0">
                                                  <!-- transform: skewX(0deg) -->
                                                  <div class="skew-x-1">
                                                    <!-- transform: skewX(1deg) --
>
                                                    <div class="skew-x-2">
                                                      <!-- transform: skewX(2deg)
-->
                                                      <div class="skew-y-0">
                                                        <!-- transform:
skewY(0deg) -->
                                                        <div class="skew-y-1">
                                                          <!-- transform:
skewY(1deg) -->
                                                        </div>
                                                      </div>
                                                    </div>
                                                  </div>
                                                </div>
                                              </div>
                                            </div>
                                          </div>
                                        </div>
                                      </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Pre-built Animations

```
<div class="animate-none">
  <!-- animation: none -->
  <div class="animate-spin">
    <!-- animation: spin 1s linear infinite -->
    <div class="animate-ping">
      <!-- animation: ping 1s cubic-bezier(0, 0, 0.2, 1) infinite -->
      <div class="animate-pulse">
        <!-- animation: pulse 2s cubic-bezier(0.4, 0, 0.6, 1) infinite -->
        <div class="animate-bounce"><!-- animation: bounce 1s infinite --></div>
      </div>
    </div>
  </div>
</div>
```

Framework Integration

React Setup

Create React App

```
npm install -D tailwindcss
npx tailwindcss init
```

Vite React

```
npm create vite@latest my-project -- --template react
cd my-project
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

tailwind.config.js for React

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{js,jsx,ts,tsx}"],
  theme: {
    extend: {},
  },
  plugins: [],
};
```

React Component Example

```
import React, { useState } from "react";

const Button = ({ children, variant = "primary", size = "md", ...props }) => {
  const baseClasses =
    "font-medium rounded-lg transition-colors duration-200 focus:outline-none
focus:ring-2 focus:ring-offset-2";

  const variants = {
    primary: "bg-blue-600 hover:bg-blue-700 text-white focus:ring-blue-500",
    secondary:
      "bg-gray-200 hover:bg-gray-300 text-gray-900 focus:ring-gray-500",
    danger: "bg-red-600 hover:bg-red-700 text-white focus:ring-red-500",
  };

  const sizes = {
    sm: "px-3 py-1.5 text-sm",
    md: "px-4 py-2 text-base",
    lg: "px-6 py-3 text-lg",
  };
  return (
    <button
      className={`${baseClasses} ${variants[variant]} ${sizes[size]}`}
      {...props}
    >
      {children}
    </button>
  );
};

const Card = ({ children, className = "" }) => {
  return (
    <div className={`bg-white rounded-lg shadow-md p-6 ${className}`}>
      {children}
    </div>
  );
};

const App = () => {
  const [count, setCount] = useState(0);

  return (
    <div className="min-h-screen bg-gray-100 py-8">
      <div className="container mx-auto px-4">
        <Card className="max-w-md mx-auto">
          <h1 className="text-2xl font-bold text-gray-900 mb-4">
            Counter: {count}
          </h1>
          <div className="space-x-2">
            <Button onClick={() => setCount(count + 1)}>Increment</Button>
            <Button variant="secondary" onClick={() => setCount(count - 1)}>
              Decrement
            </Button>
            <Button variant="danger" onClick={() => setCount(0)}>
              Reset
            </Button>
          </div>
        </Card>
      </div>
    </div>
  );
};

export default App;
```

Next.js Setup

Installation

```
npx create-next-app@latest my-project --typescript --tailwind --eslint
# or
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

tailwind.config.js for Next.js

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    "./pages/**/*.{js,ts,jsx,tsx,mdx}",
    "./components/**/*.{js,ts,jsx,tsx,mdx}",
    "./app/**/*.{js,ts,jsx,tsx,mdx}",
  ],
  theme: {
    extend: {
      backgroundImage: {
        "gradient-radial": "radial-gradient(var(--tw-gradient-stops))",
        "gradient-conic":
          "conic-gradient(from 180deg at 50% 50%, var(--tw-gradient-stops))",
      },
    },
  },
  plugins: [],
};
```

Next.js Component Example

```
// components/Layout.jsx
import Head from "next/head";
import Link from "next/link";

export default function Layout({ children, title = "My App" }) {
  return (
    <>
      <Head>
        <title>{title}</title>
        <meta name="description" content="Generated by create next app" />
        <link rel="icon" href="/favicon.ico" />
      </Head>

      <div className="min-h-screen bg-gray-50">
        <nav className="bg-white shadow">
          <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
            <div className="flex justify-between h-16">
              <div className="flex items-center">
                <Link href="/" className="text-xl font-bold text-gray-900">
                  MyApp
                </Link>
              </div>
              <div className="flex items-center space-x-4">
                <Link href="/" className="text-gray-700 hover:text-gray-900">
                  Home
                </Link>
                <Link
                  href="/about"
                  className="text-gray-700 hover:text-gray-900"
                >
                  About
                </Link>
              </div>
            </div>
          </div>
        </nav>

        <main className="max-w-7xl mx-auto py-6 px-4 sm:px-6 lg:px-8">
          {children}
        </main>
      </div>
    </>
  );
}

// pages/index.js
import Layout from "../components/Layout";

export default function Home() {
  return (
    <Layout title="Home - MyApp">
      <div className="text-center">
        <h1 className="text-4xl font-bold text-gray-900 mb-8">
          Welcome to Next.js with Tailwind CSS
        </h1>
        <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
          {[1, 2, 3].map((item) => (
            <div key={item} className="bg-white p-6 rounded-lg shadow-md">
              <h2 className="text-xl font-semibold mb-4">Card {item}</h2>
              <p className="text-gray-600">
                This is a sample card component built with Tailwind CSS.
              </p>
            </div>
          ))}
        </div>
      </div>
    </Layout>
  );
}
```

Vue Setup

Vue 3 with Vite

```
npm create vue@latest my-project
cd my-project
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

tailwind.config.js for Vue

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./index.html", "./src/**/*.{vue,js,ts,jsx,tsx}"],
  theme: {
    extend: {},
  },
  plugins: [],
};
```

Vue Component Example

```
<!-- components/TheButton.vue -->
<template>
  <button :class="buttonClasses" @click="$emit('click')">
    <slot />
  </button>
</template>

<script setup>
import { computed } from "vue";

const props = defineProps({
  variant: {
    type: String,
    default: "primary",
    validator: (value) => ["primary", "secondary", "danger"].includes(value),
  },
  size: {
    type: String,
    default: "md",
    validator: (value) => ["sm", "md", "lg"].includes(value),
  },
});

const emit = defineEmits(["click"]);

const buttonClasses = computed(() => {
  const base =
    "font-medium rounded-lg transition-colors duration-200 focus:outline-none
focus:ring-2 focus:ring-offset-2";

  const variants = {
    primary: "bg-blue-600 hover:bg-blue-700 text-white focus:ring-blue-500",
    secondary:
      "bg-gray-200 hover:bg-gray-300 text-gray-900 focus:ring-gray-500",
    danger: "bg-red-600 hover:bg-red-700 text-white focus:ring-red-500",
  };

  const sizes = {
    sm: "px-3 py-1.5 text-sm",
    md: "px-4 py-2 text-base",
    lg: "px-6 py-3 text-lg",
  };

  return `${base} ${variants[props.variant]} ${sizes[props.size]}`;
});
</script>
```

Angular Setup

Installation

```
ng new my-app
cd my-app
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init
```

angular.json configuration

```
{
  "projects": {
    "my-app": {
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "styles": ["src/styles.css"]
          }
        }
      }
    }
  }
}
```

Advanced Techniques

Custom Utilities with @apply

```css
/* styles.css */
@tailwind base;
@tailwind components;
@tailwind utilities;

.btn {
  @apply px-4 py-2 rounded-lg font-medium transition-colors duration-200;
}

.btn-primary {
  @apply bg-blue-600 text-white hover:bg-blue-700 focus:ring-2 focus:ring-blue-
500;
}

.card {
  @apply bg-white rounded-lg shadow-md p-6;
}
```

Component Layer

```css
@layer components {
  .btn {
    @apply px-4 py-2 rounded-lg font-medium transition-colors duration-200
focus:outline-none focus:ring-2 focus:ring-offset-2;
  }

  .btn-primary {
    @apply bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-500;
  }

  .btn-secondary {
    @apply bg-gray-200 text-gray-900 hover:bg-gray-300 focus:ring-gray-500;
  }
}
```

Custom Base Styles

```css
@layer base {
  h1 {
    @apply text-2xl font-bold;
  }

  h2 {
    @apply text-xl font-semibold;
  }

  h3 {
    @apply text-lg font-medium;
  }

  a {
    @apply text-blue-600 hover:text-blue-800;
  }
}
```

Extending Theme

```
// tailwind.config.js
module.exports = {
  theme: {
    extend: {
      colors: {
        brand: {
          50: "#eff6ff",
          500: "#3b82f6",
          900: "#1e3a8a",
        },
      },
      fontFamily: {
        custom: ["Inter", "sans-serif"],
      },
      spacing: {
        "18": "4.5rem",
        "88": "22rem",
      },
      borderRadius: {
        "4xl": "2rem",
      },
      animation: {
        "fade-in": "fadeIn 0.5s ease-in-out",
        "slide-up": "slideUp 0.3s ease-out",
      },
      keyframes: {
        fadeIn: {
          "0%": { opacity: "0" },
          "100%": { opacity: "1" },
        },
        slideUp: {
          "0%": { transform: "translateY(20px)", opacity: "0" },
          "100%": { transform: "translateY(0)", opacity: "1" },
        },
      },
    },
  },
};
```

Dark Mode

```
// tailwind.config.js
module.exports = {
  darkMode: "class", // or 'media'
  theme: {
    extend: {
      colors: {
        gray: {
          900: "#0f0f0f",
        },
      },
    },
  },
};
```

```
<!-- Dark mode usage -->
<div class="bg-white dark:bg-gray-900">
  <h1 class="text-gray-900 dark:text-white">Title</h1>
  <p class="text-gray-600 dark:text-gray-300">Content</p>
</div>
```

```
// Dark mode toggle (React)
const [darkMode, setDarkMode] = useState(false);

useEffect(() => {
  if (darkMode) {
    document.documentElement.classList.add("dark");
  } else {
    document.documentElement.classList.remove("dark");
  }
}, [darkMode]);
```

Arbitrary Values

```
<!-- Custom values -->
<div class="top-[117px]">
  <!-- top: 117px -->
  <div class="bg-[#1da1f2]">
    <!-- background-color: #1da1f2 -->
    <div class="grid-cols-[1fr_500px_2fr]">
```

```html
      <!-- grid-template-columns: 1fr 500px 2fr -->
      <div class="text-[22px]">
        <!-- font-size: 22px -->
        <div class="before:content-['hello_world']">
          <!-- content: 'hello world' -->
        </div>
      </div>
    </div>
  </div>
</div>
```

CSS Variables

```css
:root {
  --color-primary: #3b82f6;
  --color-secondary: #6b7280;
}

.dark {
  --color-primary: #60a5fa;
  --color-secondary: #9ca3af;
}
```

```html
<div class="bg-[var(--color-primary)]">
  <div class="text-[color:var(--color-secondary)]"></div>
</div>
```

Popular Libraries & Packages

ShadCN/UI

Modern React components built with Radix UI and Tailwind CSS.

Installation

```
npx shadcn-ui@latest init
npx shadcn-ui@latest add button
npx shadcn-ui@latest add card
npx shadcn-ui@latest add input
```

Usage Examples

```
import { Button } from "@/components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
  CardTitle,
} from "@/components/ui/card";
import { Input } from "@/components/ui/input";
import { Label } from "@/components/ui/label";

export function LoginCard() {
  return (
    <Card className="w-[350px]">
      <CardHeader>
        <CardTitle>Login</CardTitle>
        <CardDescription>
          Enter your credentials to access your account.
        </CardDescription>
      </CardHeader>
      <CardContent>
        <form>
          <div className="grid w-full items-center gap-4">
            <div className="flex flex-col space-y-1.5">
              <Label htmlFor="email">Email</Label>
              <Input id="email" placeholder="Enter your email" />
            </div>
            <div className="flex flex-col space-y-1.5">
              <Label htmlFor="password">Password</Label>
              <Input
                id="password"
                type="password"
                placeholder="Enter your password"
              />
            </div>
          </div>
        </form>
      </CardContent>
      <CardFooter className="flex justify-between">
        <Button variant="outline">Cancel</Button>
        <Button>Login</Button>
      </CardFooter>
    </Card>
  );
}
```

Aceternity UI

Beautiful components with animations and effects.

Installation

```
npm install aceternity-ui
```

Popular Components

```
import { FloatingNav } from "@/components/ui/floating-navbar";
import { HeroParallax } from "@/components/ui/hero-parallax";
import { CardHoverEffect } from "@/components/ui/card-hover-effect";
import { BackgroundGradient } from "@/components/ui/background-gradient";

// Floating Navigation
const navItems = [
  { name: "Home", link: "/" },
  { name: "About", link: "/about" },
  { name: "Contact", link: "/contact" },
];

<FloatingNav navItems={navItems} />

// Background Gradient Card
<BackgroundGradient className="rounded-[22px] max-w-sm p-4 sm:p-10 bg-white dark:bg-zinc-900">
  <img
    src="https://images.unsplash.com/photo-1441974231531-c6227db76b6e?q=80&w=2560"
    alt="jordans"
    height="400"
    width="400"
    className="object-contain"
  />
  <p className="text-base sm:text-xl text-black mt-4 mb-2 dark:text-neutral-200">
    Air Jordan 4 Retro Reimagined
  </p>
  <p className="text-sm text-neutral-600 dark:text-neutral-400">
    The Air Jordan 4 Retro Reimagined Bred will release on Saturday, May 4, 2024.
  </p>
  <button className="rounded-full pl-4 pr-1 py-1 text-white flex items-center space-x-1 bg-black mt-4 text-xs font-bold dark:bg-zinc-800">
    <span>Buy now </span>
  </button>
</BackgroundGradient>
```

Framer Motion Integration

Animation library that works perfectly with Tailwind.

Installation

```
npm install framer-motion
```

Usage Examples

```
import { motion } from 'framer-motion';

// Basic animations
<motion.div
  className="bg-blue-500 p-4 rounded-lg"
  initial={{ opacity: 0, y: 20 }}
  animate={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.5 }}
>
  Animated content
</motion.div>

// Hover animations
<motion.button
  className="bg-blue-500 text-white px-4 py-2 rounded-lg"
  whileHover={{ scale: 1.05 }}
  whileTap={{ scale: 0.95 }}
>
  Interactive Button
</motion.button>

// Layout animations
<motion.div
  className="bg-white p-6 rounded-lg shadow-lg"
  layout
  layoutId="card"
>
  Content that animates layout changes
</motion.div>

// Stagger children
const container = {
  hidden: { opacity: 0 },
  show: {
    opacity: 1,
    transition: {
      staggerChildren: 0.1
    }
  }
}

const item = {
  hidden: { opacity: 0, y: 20 },
  show: { opacity: 1, y: 0 }
}
```

```
<motion.div
  className="grid grid-cols-3 gap-4"
  variants={container}
  initial="hidden"
  animate="show"
>
  {items.map((item, index) => (
    <motion.div
      key={index}
      className="bg-white p-4 rounded-lg shadow"
      variants={item}
    >
      {item.content}
    </motion.div>
  ))}
</motion.div>
```

Headless UI

Completely unstyled, accessible UI components.

Installation

```
npm install @headlessui/react
# or for Vue
npm install @headlessui/vue
```

React Examples

```
import { Fragment } from 'react'
import { Disclosure, Menu, Transition } from '@headlessui/react'
import { ChevronDownIcon } from '@heroicons/react/20/solid'

// Dropdown Menu
<Menu as="div" className="relative inline-block text-left">
  <Menu.Button className="inline-flex w-full justify-center gap-x-1.5 rounded-md bg-white px-3 py-2 text-sm font-semibold text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 hover:bg-gray-50">
    Options
    <ChevronDownIcon className="-mr-1 h-5 w-5 text-gray-400" />
  </Menu.Button>

  <Transition
    as={Fragment}
    enter="transition ease-out duration-100"
    enterFrom="transform opacity-0 scale-95"
    enterTo="transform opacity-100 scale-100"
    leave="transition ease-in duration-75"
    leaveFrom="transform opacity-100 scale-100"
    leaveTo="transform opacity-0 scale-95"
  >
    <Menu.Items className="absolute right-0 z-10 mt-2 w-56 origin-top-right rounded-md bg-white shadow-lg ring-1 ring-black ring-opacity-5 focus:outline-none">
      <div className="py-1">
        <Menu.Item>
          {({ active }) => (
            <a
              href="#"
              className={`${
                active ? 'bg-gray-100 text-gray-900' : 'text-gray-700'
              } block px-4 py-2 text-sm`}
            >
              Account settings
            </a>
          )}
        </Menu.Item>
      </div>
    </Menu.Items>
  </Transition>
</Menu>

// Modal Dialog
import { Dialog } from '@headlessui/react'

const [isOpen, setIsOpen] = useState(false)

<Dialog
  open={isOpen}
  onClose={() => setIsOpen(false)}
  className="relative z-50"
>
  <div className="fixed inset-0 bg-black/30" aria-hidden="true" />
  <div className="fixed inset-0 flex items-center justify-center p-4">
    <Dialog.Panel className="mx-auto max-w-sm rounded bg-white p-6">
      <Dialog.Title className="text-lg font-medium leading-6 text-gray-900 mb-4">
        Confirm Action
      </Dialog.Title>
      <Dialog.Description className="text-sm text-gray-500 mb-4">
        Are you sure you want to proceed with this action?
      </Dialog.Description>

      <div className="flex gap-4">
        <button
          className="flex-1 rounded bg-blue-600 px-4 py-2 text-white hover:bg-blue-700"
          onClick={() => setIsOpen(false)}
        >
          Confirm
        </button>
        <button
          className="flex-1 rounded bg-gray-300 px-4 py-2 text-gray-700 hover:bg-
```

```
gray-400"
          onClick={() => setIsOpen(false)}
        >
          Cancel
        </button>
      </div>
    </Dialog.Panel>
  </div>
</Dialog>
```

React Hook Form with Tailwind

```
import { useForm } from "react-hook-form";

function ContactForm() {
  const {
    register,
    handleSubmit,
    formState: { errors },
  } = useForm();

  const onSubmit = (data) => console.log(data);

  return (
    <form
      onSubmit={handleSubmit(onSubmit)}
      className="max-w-md mx-auto space-y-4"
    >
      <div>
        <label className="block text-sm font-medium text-gray-700 mb-1">
          Name
        </label>
        <input
          {...register("name", { required: "Name is required" })}
          className={`w-full px-3 py-2 border rounded-md shadow-sm focus:outline-
none focus:ring-2 focus:ring-blue-500 ${
            errors.name ? "border-red-500" : "border-gray-300"
          }`}
        />
        {errors.name && (
          <p className="mt-1 text-sm text-red-600">{errors.name.message}</p>
        )}
      </div>

      <div>
        <label className="block text-sm font-medium text-gray-700 mb-1">
          Email
        </label>
        <input
          type="email"
          {...register("email", {
            required: "Email is required",
            pattern: {
              value: /^\S+@\S+$/i,
              message: "Invalid email address",
            },
          })}
          className={`w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-2 focus:ring-blue-500 ${
            errors.email ? "border-red-500" : "border-gray-300"
          }`}
        />
        {errors.email && (
          <p className="mt-1 text-sm text-red-600">{errors.email.message}</p>
        )}
      </div>

      <button
        type="submit"
        className="w-full bg-blue-600 text-white py-2 px-4 rounded-md hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-offset-2 transition-colors"
      >
        Submit
      </button>
    </form>
  );
}
```

Additional Useful Libraries

Lucide React (Icons)

```
npm install lucide-react
```

```
import { Mail, Phone, MapPin } from "lucide-react";

<div className="flex items-center space-x-2">
  <Mail className="w-5 h-5 text-gray-500" />
  <span>contact@example.com</span>
</div>;
```

React Hot Toast (Notifications)

```
npm install react-hot-toast
```

```
import toast, { Toaster } from 'react-hot-toast'

// In your component
const notify = () => toast.success('Successfully saved!')

// In your app root
<Toaster
  position="top-right"
  toastOptions={{
    className: 'bg-white text-gray-900 shadow-lg',
    duration: 4000,
  }}
/>
```

Recharts (Charts)

```
npm install recharts
```

```
import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip,
ResponsiveContainer } from 'recharts'

const data = [
  { name: 'Jan', value: 400 },
  { name: 'Feb', value: 300 },
  { name: 'Mar', value: 600 },
]

<div className="w-full h-64">
  <ResponsiveContainer>
    <LineChart data={data}>
      <CartesianGrid strokeDasharray="3 3" />
      <XAxis dataKey="name" />
      <YAxis />
      <Tooltip />
      <Line type="monotone" dataKey="value" stroke="#3b82f6" strokeWidth={2} />
    </LineChart>
  </ResponsiveContainer>
</div>
```

Best Practices

Performance Optimization

Purge Unused Styles

```
// tailwind.config.js
module.exports = {
  content: ["./src/**/*.{js,jsx,ts,tsx}", "./public/index.html"],
  // Tailwind will only include styles for classes found in these files
};
```

Use JIT Mode (Just-In-Time)

```
// tailwind.config.js
module.exports = {
  mode: "jit", // Enable JIT mode
  content: ["./src/**/*.{js,jsx,ts,tsx}"],
  // ...
};
```

Code Organization

Component-Based Approach

```
// Create reusable component classes
const buttonBase =
  "px-4 py-2 rounded-lg font-medium transition-colors duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2";
const buttonVariants = {
  primary: "bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-500",
  secondary: "bg-gray-200 text-gray-900 hover:bg-gray-300 focus:ring-gray-500",
};

const Button = ({ variant = "primary", children, ...props }) => (
  <button className={`${buttonBase} ${buttonVariants[variant]}`} {...props}>
    {children}
  </button>
);
```

Use CSS-in-JS Libraries

```
// With clsx for conditional classes
import clsx from "clsx";

const Button = ({ variant, size, disabled, children }) => (
  <button
    className={clsx("font-medium rounded-lg transition-colors", {
      "bg-blue-600 text-white hover:bg-blue-700": variant === "primary",
      "bg-gray-200 text-gray-900 hover:bg-gray-300": variant === "secondary",
      "px-3 py-1.5 text-sm": size === "sm",
      "px-4 py-2": size === "md",
      "px-6 py-3 text-lg": size === "lg",
      "opacity-50 cursor-not-allowed": disabled,
    })}
    disabled={disabled}
  >
    {children}
  </button>
);
```

Accessibility Best Practices

Focus States

```
<!-- Always include focus states -->
<button
  class="bg-blue-500 hover:bg-blue-600 focus:ring-2 focus:ring-blue-500 focus:outline-none"
>
  Button
</button>

<!-- Form inputs -->
<input
  class="border border-gray-300 focus:border-blue-500 focus:ring-2 focus:ring-blue-200"
/>
```

Screen Reader Support

```
<!-- Use semantic HTML -->
<nav class="...">
  <ul class="flex space-x-4">
    <li><a href="/" class="text-blue-600 hover:text-blue-800">Home</a></li>
    <li>
      <a href="/about" class="text-blue-600 hover:text-blue-800">About</a>
    </li>
  </ul>
</nav>

<!-- Add ARIA labels when needed -->
<button class="p-2 rounded-full hover:bg-gray-100" aria-label="Close dialog">
  <XIcon class="w-5 h-5" />
</button>
```

Color Contrast

```html
<!-- Ensure sufficient contrast -->
<div class="bg-blue-600 text-white">
  <!-- Good contrast -->
  <div class="bg-yellow-200 text-yellow-800">
    <!-- Good contrast -->

    <!-- Avoid poor contrast -->
    <!-- <div class="bg-gray-100 text-gray-200"> Bad contrast -->
  </div>
</div>
```

Responsive Design Best Practices

Mobile-First Approach

```html
<!-- Start with mobile styles, then add larger breakpoints -->
<div
  class="
  p-4
  sm:p-6
  md:p-8
  lg:p-12

  text-sm
  sm:text-base
  lg:text-lg

  grid-cols-1
  sm:grid-cols-2
  lg:grid-cols-3
  xl:grid-cols-4
"
></div>
```

Container Queries (Future)

```html
<!-- Use responsive utilities effectively -->
<div class="container mx-auto px-4 sm:px-6 lg:px-8">
  <div class="max-w-4xl mx-auto">
    <!-- Content -->
  </div>
</div>
```

Development Workflow

VS Code Extensions

- **Tailwind CSS IntelliSense**: Autocomplete and syntax highlighting
- **Headwind**: Class sorting
- **Tailwind Docs**: Quick documentation lookup

PostCSS Plugins

```
// postcss.config.js
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
    ...(process.env.NODE_ENV === "production" ? { cssnano: {} } : {}),
  },
};
```

Build Scripts

```
{
  "scripts": {
    "build-css": "tailwindcss -i ./src/input.css -o ./dist/output.css --watch",
    "build-css-prod": "tailwindcss -i ./src/input.css -o ./dist/output.css --minify"
  }
}
```

Common Patterns

Card Components

```
<div
  class="bg-white rounded-lg shadow-md hover:shadow-lg transition-shadow duration-200"
>
  <img class="w-full h-48 object-cover rounded-t-lg" src="..." alt="..." />
  <div class="p-6">
    <h3 class="text-xl font-semibold text-gray-900 mb-2">Title</h3>
    <p class="text-gray-600 mb-4">Description text here...</p>
    <button
      class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 transition-colors"
    >
      Learn More
    </button>
  </div>
</div>
```

Form Layouts

```html
<form class="max-w-lg mx-auto space-y-6">
  <div class="grid grid-cols-1 sm:grid-cols-2 gap-4">
    <div>
      <label class="block text-sm font-medium text-gray-700 mb-1">First Name</label>
      <input class="w-full px-3 py-2 border border-gray-300 rounded-md focus:ring-2 focus:ring-blue-500 focus:border-transparent" />
    </div>
    <div>
      <label class="block text-sm font-medium text-gray-700 mb-1">Last Name</label>
      <input class="w-full px-3 py-2 border border-gray-300 rounded-md focus:ring-2 focus:ring-blue-500 focus:border-transparent" />
    </div>
  </div>
  <div>
    <label class="block text-sm font-medium text-gray-700 mb-1">Email</label>
    <input type="email" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:ring-2 focus:ring-blue-500 focus:border-transparent" />
  </div>
  <div>
    <label class="block text-sm font-medium text-gray-700 mb-1">Message</label>
    <textarea rows="4" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:ring-2 focus:ring-blue-500 focus:border-transparent"></textarea>
  </div>
  <button type="submit" class="bg-blue-600 text-white px-4 py-2 rounded-md hover:bg-blue-700 transition-colors">Submit</button>
</form>
```

more coming soon ...