



Setup & Configuration

Configuration Options

theme : Customize colors, spacing, fonts, breakpoints, etc.

Example:

```
theme: {
  extend: {
    colors: {
      'custom-blue': '#123456',
    }
  }
}
```

plugins : Add custom styles and functionality.

Example:

```
plugins: [
  require('@tailwindcss/forms'),
]
```

variants : Control which variants are enabled for each utility.

(Deprecated in Tailwind CSS v3.0+; use the **content** configuration instead)

prefix : Add a prefix to all Tailwind CSS class names to avoid conflicts.

Example:

```
prefix: 'tw-'
```

corePlugins : Enable or disable specific Tailwind CSS core plugins.

Example:

```
corePlugins: {
  container: false,
}
```

darkMode : Enable dark mode based on prefers-color-scheme or class name.

Example:

```
darkMode: 'class'
```

content : Specify the files to scan for usage of Tailwind CSS classes.

Example:

```
content: [
  './public/**/*.html',
  './src/**/*.js', './src/**/*.ts', './src/**/*.tsx',
]
```

safelist : Specify classes that should not be removed during production builds.

Example:

```
safelist: [
  'bg-red-500',
  'text-green-200',
  'lg:bg-blue-400'
]
```

blocklist : Specify classes that should always be removed during production builds.

Example:

```
blocklist: [
  'bg-red-500',
  'text-green-200',
  'lg:bg-blue-400'
]
```

Typography Styles

font-sans	Apply a sans-serif font family.
font-serif	Apply a serif font family.
font-mono	Apply a monospace font family.
text-xs	Extra small text size (0.75rem / 12px).
text-sm	Small text size (0.875rem / 14px).
text-base	Base text size (1rem / 16px).
text-lg	Large text size (1.125rem / 18px).
text-xl	Extra large text size (1.25rem / 20px).
font-thin	Font weight 100.
font-extralight	Font weight 200.
font-light	Font weight 300.
font-normal	Font weight 400 (normal).
font-medium	Font weight 500 (medium).
font-semibold	Font weight 600 (semi-bold).
font-bold	Font weight 700 (bold).
font-extrabold	Font weight 800 (extra-bold).
font-black	Font weight 900 (black).
italic	Set font style to italic.
not-italic	Set font style to normal (non-italic).
uppercase	Transform text to uppercase.
lowercase	Transform text to lowercase.
capitalize	Transform text to capitalize.

Core Concepts & Utilities

Utility-First Fundamentals

Tailwind CSS uses a utility-first approach, meaning you compose styles by applying pre-defined utility classes directly in your HTML.

Example:

```
<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">
  Click me
</button>
```

Responsive Design: Use breakpoint prefixes like `sm:`, `md:`, `lg:`, `xl:`, and `2xl:` to apply styles at different screen sizes.

Example:

```
<div class="md:flex">
  <!-- ... -->
</div>
```

Hover, Focus, and Other States: Use prefixes like `hover:`, `focus:`, `active:`, `disabled:` to style different element states.

Example:

```
<button class="hover:text-green-500 focus:outline-none focus:ring-2 focus:ring-green-500">
  Hover me
</button>
```

Dark Mode: Use the `dark:` prefix to style elements in dark mode (requires dark mode to be enabled in your `tailwind.config.js`).

Example:

```
<div class="bg-white dark:bg-gray-800">
  <!-- ... -->
</div>
```

Customization & Directives

Adding Custom Styles

While Tailwind encourages utility classes, you can add custom CSS:

- Using `@apply`: Apply existing Tailwind utility classes within your CSS.

```
.btn-primary {
  @apply bg-blue-500 hover:bg-blue-700 text-white font-bold
  py-2 px-4 rounded;
}
```
- Directly in CSS:** Write standard CSS rules in your CSS files. Tailwind will handle the rest.

```
.my-custom-class {
  color: purple;
  font-size: 1.2rem;
}
```

Essential Utilities

Typography: <code>font-{family}</code> <code>text-{size}</code> <code>font-{weight}</code> <code>italic</code>, <code>not-italic</code> <code>underline</code>, <code>line-through</code> <code>text-{color}</code>	Background: <code>bg-{color}</code> <code>bg-{image}</code> <code>bg-{position}</code> <code>bg-{size}</code> <code>bg-repeat</code>, <code>bg-no-repeat</code>
Spacing (Margin & Padding): <code>m{t b l r x y}-{size}</code> <code>p{t b l r x y}-{size}</code>	Flexbox & Grid: <code>flex</code>, <code>inline-flex</code> <code>grid</code>, <code>inline-grid</code> <code>flex-{row col}</code> <code>justify-{content}</code> <code>items-{alignment}</code> <code>grid-cols-{number}</code>
Border: <code>border</code> <code>border-{width}</code> <code>border-{color}</code> <code>rounded</code>, <code>rounded-{size}</code>	Positioning: <code>static</code>, <code>relative</code>, <code>absolute</code>, <code>fixed</code>, <code>sticky</code> <code>top-{value}</code>, <code>right-{value}</code>, <code>bottom-{value}</code>, <code>left-{value}</code> <code>z-{index}</code>
Display: <code>block</code>, <code>inline-block</code> <code>inline</code> <code>flex</code>, <code>inline-flex</code> <code>grid</code>, <code>inline-grid</code> <code>table</code>, <code>inline-table</code> <code>hidden</code>	Effects: <code>shadow</code>, <code>shadow-{size}</code> <code>opacity-{value}</code> <code>blur</code>, <code>blur-{size}</code> <code>transition</code>, <code>duration-{time}</code> <code>transform</code>, <code>scale-{value}</code>

Useful Directives

@tailwind : Injects Tailwind's base, components, and utilities styles. <pre>@tailwind base; @tailwind components; @tailwind utilities;</pre>	@layer : Organize your custom CSS into layers (base, components, utilities) for better control over specificity. <pre>@layer components { .btn-primary { @apply ...; } }</pre>
@apply : Inline existing utility classes into custom CSS rules. <pre>.my-element { @apply flex items-center; }</pre>	@config : Specify a custom Tailwind config file (rarely needed). <pre>@config 'path/to/my/tailwind.config.js';</pre>
@variants : (Deprecated) - In Tailwind v2, used to generate variants for custom CSS. Superseded by the content configuration.	

AI Tools & Resources

AI-Powered Tailwind Tools

Several AI tools can help you generate Tailwind CSS code, including:

- **TeleportHQ**: AI-powered design-to-code platform that supports Tailwind CSS.
- **Locofy.ai**: Convert designs to responsive code, including Tailwind CSS.
- **Galytix**: Use AI to generate Tailwind code from Figma designs.
- **Windframe**: Quickly prototype website sections with AI-generated Tailwind CSS code.

Official Resources

- **Tailwind CSS Documentation:** tailwindcss.com
- **Tailwind UI:** tailwindui.com - Component library built with Tailwind CSS.

Community Resources

- **Tailwind CSS Discord:** Active community for support and discussions.
- **Tailwind Weekly:** Newsletter with tips, tricks, and resources.
- **tailwindcss-plugins:** A repository of useful Tailwind CSS plugins.
- **Various online tutorials, blog posts, and videos.**