Tailwind CSS Utility-first CSS

START HERE

npm i tailwindcss @tailwindcss/cli
Install

hover:bg-sky-700
State variant

@import "tailwindcss";
Load it in your CSS

w-[17px]
One-off value

npx @tailwindcss/cli -i *<in>* **-o** *<out>* **--watch**
Build CSS

flex items-center gap-4
Layout in markup

p-4 text-lg font-bold
Space and type

md:flex-row
Responsive variant

Your source files class names found → **scan** → **input.css** @import "tailwindcss" → **build** → **Tailwind CLI / Vite** only used classes → **emit** → **output.css** linked in HTML

CONCEPTS

Utilities are single-purpose classes you combine in markup. Variants are prefixes: md:, hover:, dark:. Unprefixed means every size (mobile first).

Write class names in full. Tailwind scans for literal strings, so "bg-" + color is never found; use a lookup of complete names instead.

SETUP AND BUILD

npm install tailwindcss @tailwindcss/cli
Install with the standalone CLI

npm install tailwindcss @tailwindcss/vite
Install the Vite plugin

@import "tailwindcss";
Pull in Tailwind (preflight, theme, utilities)

npx @tailwindcss/cli -i *<in>* **-o** *<out>*
Build the CSS once

npx @tailwindcss/cli -i *<in>* **-o** *<out>* **--watch**
Rebuild on change

npx @tailwindcss/cli -i *<in>* **-o** *<out>* **--minify**
Minified production build

@source "*<path>***";**
Scan extra files for classes

LAYOUT

flex
Flex container

flex-col
Stack children vertically

items-center
Align items on the cross axis

justify-between
Spread items along the main axis

gap-4
Gap between children (1 unit = 0.25rem)

grid grid-cols-3
Three-column grid

col-span-2
Span two grid columns

block / inline-block / hidden
Display values

relative / absolute / fixed / sticky
Positioning

inset-0
Pin to all four edges

z-10
Stacking order

overflow-hidden
Clip overflowing content

container mx-auto
Centred, width-capped container

LAYOUT (cont.)

flex-wrap
Let items wrap

grow / shrink-0
Flex grow and shrink

SPACING AND SIZING

p-4
Padding on all sides

px-4 py-2
Horizontal and vertical padding

pt-2 / pr-2 / pb-2 / pl-2
Padding per side

m-4
Margin on all sides

mx-auto
Centre a block horizontally

w-full
Width 100%

h-screen
Height 100vh

w-64
Fixed width (16rem)

size-10
Width and height together

max-w-md
Cap the width

min-h-screen
At least full viewport height

w-1/2
Fractional width

TYPOGRAPHY

text-lg
Font size (sm, base, lg, xl, 2xl...)

font-bold
Font weight

text-center
Align text

text-gray-700
Text colour

leading-tight
Line height

font-mono
Monospace font family

truncate
One line with ellipsis

line-clamp-3
Clamp to three lines

uppercase
Transform to upper case

underline
Underline text

COLOUR AND EFFECTS

bg-white
Background colour

bg-blue-500
Palette colour (50 to 950)

bg-black/50
Colour with 50% opacity

COLOUR AND EFFECTS (cont.)

opacity-50
Element opacity

border
1px border

border-2 border-gray-300
Border width and colour

rounded-lg
Rounded corners

rounded-full
Pill or circle

shadow-md
Box shadow

ring-2 ring-blue-500
Outline ring (focus styles)

transition
Animate property changes

VARIANTS

hover:bg-sky-700
On hover

focus:ring-2
On focus

active: / disabled:
Pressed / disabled state

focus-visible:outline-2
Keyboard focus only

sm: md: lg: xl: 2xl:
Min-width breakpoints (40/48/64/80/96rem)

VARIANTS (cont.)

max-md:hidden
Below the md breakpoint

dark:bg-gray-800
Dark mode

group / group-hover:underline
Style child on parent hover

first: last: odd: even:
Position variants

CUSTOM VALUES AND THEME

w-[17px]
Arbitrary value

bg-[#316ff6]
Arbitrary colour

bg-(--my-var)
Use a CSS variable

bg-red-500!
Important modifier (trailing !)

@theme { --color-brand: #f00; }
Add a design token (bg-brand)

@theme { --breakpoint-sm: 30rem; }
Override a default value

@apply *<utilities>***;**
Inline utilities into your own CSS

@custom-variant dark (&:where(.dark, .dark *));
Class-based dark mode