Tailwind CSS

**Tailwind CSS** is an [open source](https://en.wikipedia.org/wiki/Open_source" \o "Open source) [CSS](https://en.wikipedia.org/wiki/CSS" \o "CSS) [framework](https://en.wikipedia.org/wiki/CSS_framework" \o "CSS framework). The main feature of this library is that, unlike other CSS frameworks like [Bootstrap](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)" \o "Bootstrap (front-end framework)), it does not provide a series of predefined classes for [elements](https://en.wikipedia.org/wiki/HTML_element" \o "HTML element) such as buttons or tables. Instead, it creates a list of "utility" CSS classes that can be used to style each element by mixing and matching.

**Introduction**

Benefit of using tailwind CSS that we easily style or customize our HTML element without leaving html file with the help of the classes. Not like Bootstrap (Bootstrap add a lot of extra code and file to your website which can effect it’s performance and loading speed suppose we create a div container and use Bootstrap classes it automatically add display-flex property that is not required for container but tailwind CSS Dose not add extra things)

Shortcuts

press r + enter to restart the server

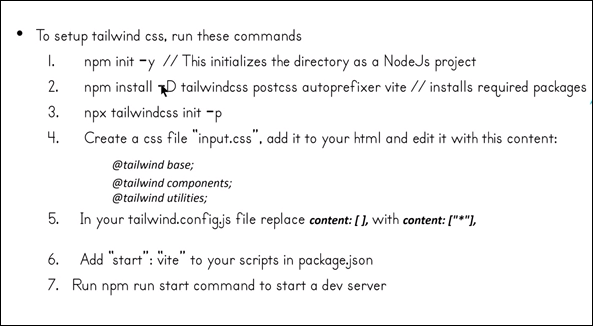
press u + enter to show server url

press o + enter to open in browser

press c + enter to clear console

press q + enter to quit

vite – it is a local server



1. **Font Family.**

CSS frameworks like Tailwind CSS, you can use utility classes to quickly apply specific font families to your elements, like font-sans, font-serif, and font-mono.

font-sans

The quick brown fox jumps over the lazy dog.

font-serif

The quick brown fox jumps over the lazy dog.

font-mono

The quick brown fox jumps over the lazy dog.

1. **Font sizing.**

text-base, text-xs, text-sm, text-lg, text-xl, text-2xl, text-3xl, and text-4xl set the text size to base, extra-small, small, large, extra-large, 2 times extra-large, 3 times extra-large, and 4 times extra-large, respectively,

1. **Font weight.**

* Font-light
* Font-normal
* Font-medium
* Font-semibold
* Font-bold

1. **Font style.**

* Italic
* Non-italic

1. **Latter-spacing.**

* Tracking-tight
* Tracking-normal
* Tracking-wide

1. **Margin.**  
   In Tailwind CSS, margin classes are used to control the spacing around an element.

* m-(size): Apply from All side.(means left the space from all side (L,R,B,T)around the element)
* mr-(size): Apply from Right side. (means left the space from the right side of the element)
* ml-(size): Apply from Left side. . (means left the space from the left side of the element)
* mt-(size): Apply from Top side. (means left the space from the top side of the element)
* mb-(size): Apply from Bottom side. (means left the space from the bottom side of the element)
* mx-(size): Apply from X-axis (means left the space from both the sides left and right on x-axis) .
* my-(size): Apply from Y-axis (means left the space from both the sides top and bottom on y-axis) .
* We can also use Negative margin
* Class=”-mx-8”

1. **Paddling.**  
   In Tailwind CSS, padding classes are used to control the spacing inside an element.

* p-(size): Apply from All side. (means left the space from all side (L,R,B,T) between the border and element)
* pr-(size): Apply from Right side. (means left the space from right side between the border and element)
* pl-(size): Apply from Left side. (means left the space from left side between the border and element)
* pt-(size): Apply from Top side. (means left the space from top side between the border and element)
* pb-(size): Apply from Bottom side. (means left the space from bottom side between the border and element)
* px-(size): Apply from X-axis (left and right) side. . (means left the space from right and left side between the border and element)
* py-(size): Apply from Y-axis (top and bottom) side. (means left the space from top and bottom side between the border and element)

1. **Space Between**

The space-x and space-y utilities are used to apply space between child elements in a flex container. They automatically apply margin-right (space-x) or margin-bottom (space-y) to the child elements, creating the desired spacing.

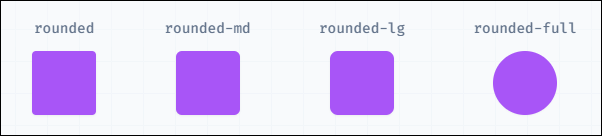
* If you want to create horizontal spacing between the child elements , you can use space-x: space-x-{amount}
* If you want to create vertical spacing between the child elements, you can use space-y: space-y-{amount}
* space-x-reverse, space-y-reverse it arrange your child element in reverse order.

1. **Border**

**Border-radius**

**Use utilities like rounded , rounded-sm , rounded-md, rounded-lg to apply different border radius sizes to an element.**

**For all side**

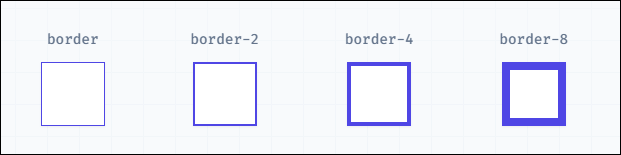
****

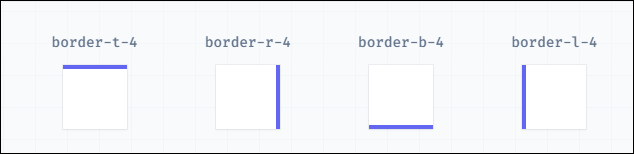
**For two side**

****

**For individual side**

**Border-width**

****

****

****

**Border-style**

****

1. **Square brackets Notation**

If you are referring to square brackets in the context of dynamically generating classes or using dynamic values in your classes,(Means suppose you are using random value for margin or padding in any element but that value of utility class are not exist in that case we use square brackets notation for random value in tailwind CSS ).

Example

Class=”mx-[34px]” 1rem = 16px

Class=”my-[4rem]”

1. **Responsive design for every screen size**

Every utility class in tailwind can be applied conditionally at different breakpoint, which makes it a piece of cake to build complex responsive interface without ever leaving your HTML page.

There are five breakpoints by default , inspired by common device resolutions.

****

1. **Hover, Focus and Activat**

**Hover (hover:):**

The **hover** variant is used to apply styles when the mouse is over an element.

For example, you might want to change the background color of a button when a user hovers over it:

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

**Focus (focus:):**

The **focus** variant is used to apply styles when an element gains focus. This is often used in the context of form elements, like input fields.

For example, you might want to change the border color of an input field when it is in focus:

<**input type="text" class="border focus:border-blue-500">**

**Active (active:):**

The **active** variant is used to apply styles when an element is actively being interacted with, typically when a mouse button is pressed. For example, you might want to change the background color of a button when it is being clicked:

**<button class="bg-blue-500 active:bg-blue-800 text-white font-bold py-2 px-4 rounded">Click me</button>**

1. **Apply directive**

In Tailwind CSS, the **@apply** directive is a powerful feature that allows you to use existing utility classes to compose and apply your own custom styles.

We can create a own class of many utility classes with the help of apply directive method.

**Random**

Flex:- flex class is used to make a flexible of your element and arrange in column wise.

Justify-end : its arrange your item on extreme end.