National University of Computer & Emerging Sciences, Karachi Computer Science Department



**Course: Web Engineering Lab**

**Course Code: SE-3003**

**Spring 2024, Lab Manual – 05**

**Tailwind CSS**

**Introduction To Tailwind CSS**

Tailwind CSS is an open-source CSS framework that provides utility classes for styling your website directly in your HTML. Unlike traditional frameworks like Bootstrap, it doesn't offer pre-built components, but a large collection of low-level utility classes like ***text-blue-500, w-full, or flex-row.***

**How it works:**

You add these classes to your HTML elements, and Tailwind translates them into the corresponding CSS styles when you build your website. This allows for rapid development and a clean, consistent user interface.

**Benefits:**

* **Speed:** Build UIs quickly without writing much custom CSS.
* **Flexibility:** Highly customizable to achieve any design.
* **Maintainability:** Easier to read and maintain HTML with clear class names.
* **Responsiveness:** Responsive styles built-in for various screen sizes.

**Drawbacks:**

* **Learning curve**: Takes time to learn the class names and syntax.
* **Verbosity:** Can lead to HTML files with many class names.
* **Limited customization**: Not ideal for creating highly unique UI elements.

**How To Setup Tailwind CSS:**

There are two ways to use Tailwind CSS in Your html document.

1. Online Through CDN Link
2. Offline Use Through Setting up Tailwind offline for project.

* **Online Through CDN Link**

Include the Tailwind CSS CDN link in your HTML's <head> section:

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

**Demo Code Given Below**

<!doctype html>

<html>

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

</head>

<body>

  <h1 class="text-3xl font-bold underline bg-red-600 text-center mt-10">

    Welcome to Tailwind CSS

  </h1>

</body>

</html>

* **Setting up Tailwind offline for project**

1. For Setting up Tailwind CSS offline you must have Node.js Installed on your system
2. Link To download Node JS (Download and Install the latest version from the below link [Node.js (nodejs.org)](https://nodejs.org/en)
3. Create a Folder (E.g. TailwindCSS) and open your folder in vs code.
4. Open the Terminal in Your VS code and Initialize your project using ***npm init -y*** This will create the Package.json file in your folder***.***
5. Create Public Folder inside Your Main Folder.
6. Inside Public Folder Create your index.html file.
7. Now Install the Tailwind using ***npm install -D tailwindcss*** (run this on terminal) this will create package-lock. Json***.***
8. Now for initializing Tailwind CSS run ***npx tailwindcss init*** on terminal this will create tailwind.config.js.
9. Now we need to configure which files we need to use tailwindcss for that we have to write the location of our files and folders in the content inside tailwind.config.js.
10. Suppose I want to use it for html files inside the public folder so I will write

content: [["./Public/\*.{html}"]],

1. Now Create a New Folder src in your main folder
2. Create input.css file inside src folder and paste these lines.

@tailwind base;

@tailwind components;

@tailwind utilities;

1. Now we will create our actual output file for that we will run the command ***npx tailwindcss -i ./src/input.css -o ./Public/output.css –watch*** in terminal
2. Now link your output.css with html files (i.e index.html)

* **Tip for Tailwind CSS Class**

You can install the extension Tailwind CSS IntelliSense in your vs code to get the help of tailwind classes.

**Getting Started with Tailwindcss Classes**

1. **Core Utility Classes:**

Tailwind CSS takes a unique approach to styling your web projects, offering a vast collection of pre-defined classes instead of traditional CSS. Think of these classes as tiny building blocks, each performing a specific styling task like setting margin, color, or font size. Unlike custom CSS, you combine these utility classes directly in your HTML, reducing repetitive code and speeding up development**.**

**Common Tailwind Utility Classes and Examples**

|  |  |  |
| --- | --- | --- |
| **Utility Class** | **Description** | **Example** |
| Spacing | | |
| m-x | Margin on all sides (horizontal) | <div class="m-4">Content with 1rem margin around</div> |
| my-y | Margin on top and bottom (vertical) | <p class="my-2">Paragraph with 0.5rem top and bottom margin</p> |
| p-x | Padding on all sides (horizontal) | <button class="p-4">Button with 1rem padding</button> |
| py-y | Padding on top and bottom (vertical) | <h2 class="py-3">Heading with 1.5rem top and bottom padding</h2> |
| Sizing | | |
| w-x | Width | <img class="w-full" src="image.jpg"> (Full-width image) |
| h-y | Height | <iframe class="h-64" src="video.mp4"></iframe> (64px tall iframe) |
| min-w-x | Minimum width | <input class="min-w-full" type="text"> (Input field with minimum full width) |
| max-h-y | Maximum height | <div class="max-h-48">Scrollable content with max height of 48px</div> |
| Text & Color | | |
| text-x | Text color | <h3 class="text-blue-500">Blue heading</h3> |
| font-weight | Font weight | <span class="font-bold">Important text</span> |
| text-size | Text size | <p class="text-2xl">Large paragraph text</p> |
| bg-x | Background color | <div class="bg-gray-200">Content with background color</div> |
| Layout & Display | | |
| flex | Makes element a flexbox container | <div class="flex">Flexbox with child elements</div> |
| grid | Makes element a grid container | <main class="grid grid-cols-3">Three-column grid layout</main> |
| block | Makes element display as block-level | <a class="block" href="#">Block-level button</a> |
| inline | Makes element display as inline | <span class="inline">Inline text</span> |
| Positioning | | |
| absolute | Positions element relative to its nearest positioned ancestor | <div class="absolute top-0 right-0">Absolutely positioned element</div> |
| relative | Establishes positioning context for its children | <div class="relative">Element with absolutely positioned children</div> |
| fixed | Positions element relative to the viewport | <nav class="fixed top-0 left-0">Fixed navigation bar</nav> |

1. **Customization:**

Tailwind's beauty lies in modifying base classes using various suffixes. For example, the text-gray-700 class sets the text color to a specific gray shade. Modify it with:

**Responsiveness:** Add suffixes like sm:, md:, lg:, etc., to apply styles only at specific screen sizes (e.g., text-gray-700 md:text-gray-800 changes text color on medium screens and above).

**Hover, Focus, Active States:** Use .hover, .focus, and .active to define styles for different interactions (e.g., button:hover:bg-blue-500 changes button background on hover).

**Variants:** Apply variations like opacity-50 for transparency, rounded-full for circular elements, and more, directly after the base class (e.g., text-gray-700 font-bold opacity-75).

**Customizing Values:**

Each utility class often accepts numerical values (e.g., px for spacing, rem for font size). While Tailwind provides pre-defined options, you can override them:

Use inline styles directly after the class (e.g., p-4 px-8 sets both horizontal and vertical padding, but with different values).

In the tailwind.config.js file, adjust the default values for specific utilities by changing the corresponding array in the config section.

**Creating Custom Utility Classes:**

When specific styles appear frequently, create custom classes for organization and efficiency: Define a .card class in your CSS with desired margin, padding, and background:

.card {

margin: 1rem;

padding: 2rem;

background-color: #f5f5f5;

}

Use it throughout your HTML as needed:

HTML

<div class="card">

</div>

**Using Arbitrary Values and Properties:**

Tailwind allows adding custom CSS properties and values not directly covered by utilities:

Use the CSS @apply directive within your template or component:

**HTML**

<span class="text-red-500" @apply="font-italic underline">Emphasized text</span>

Use code [with caution.](https://gemini.google.com/faq#coding)

This adds the inline font-italic and underline styles while keeping text-red-500 from the .text-red-500 class.

**Mastering Responsiveness**

Tailwind CSS shines in its approach to responsive design, offering built-in tools and modifiers to seamlessly adapt your website across different screen sizes. Here's a deep dive into how it empowers responsive development:

1. **Mobile-First Philosophy:**

Tailwind adopts a mobile-first approach, meaning styles are applied by default to mobile devices (small screens).

As screen sizes increase, you add responsive modifiers to target larger devices (e.g., md:, lg:, xl:) and adjust styles accordingly. This ensures your website looks its best on mobile first, then gracefully scales up for larger screens.

1. **Responsive Modifiers:**

Each utility class comes with responsive variants denoted by suffixes like md:, lg:, xl:, etc.

These modifiers let you apply different styles at different screen widths.

For example, you can use p-4 md:p-8 to set padding of 1rem on all sides for small screens, but increase it to 2rem on medium and larger screens.

1. **Breakpoints:**

You can customize the screen sizes at which responsive modifiers take effect by adjusting Tailwind's pre-defined breakpoints in the tailwind.config.js file. This allows you to tailor the responsiveness to your specific design needs and target devices you prioritize.

**To Customize do the following.**

Open tailwind.config.js in your preferred text editor or IDE.

Search for the screens property within the theme object. It's often found under the extend section, like this:

JavaScript

module.exports = {

theme: {

extend: {

screens: {

'sm': '640px',

'md': '768px',

'lg': '800px', // New breakpoint for two cards

'xl': '1280px',

'2xl': '1536px',

}

}

}

};

1. **Utility-First Styling:**

Unlike frameworks with pre-built components, Tailwind uses low-level utility classes, giving you granular control over responsiveness. You can combine and modify these classes to achieve intricate, device-specific layouts and styles.

1. **Responsiveness in Action:**

<!doctype html>

<html>

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="./output.css">

</head>

<body class="bg-slate-900">

  <p class="sm:text-sky-500 md:text-green-600 lg:text-red-700 text-white">

    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatum doloribus ipsum, nesciunt, odit, maiores consequatur sed est temporibus rerum vel nihil perferendis veniam aliquam! Distinctio quas aliquid beatae officia corrupti.

  </p>

</body>

</html>

The provided HTML uses Tailwind's responsive modifiers to change the text color based on screen size:

**sm:text-sky-500**: Sets light blue on small screens.

**md:text-green-600:** Overrides to dark green on medium screens.

**lg:text-red-700:** Overrides again to dark red on large screens.

**text-white:** Acts as the fallback for all other screen sizes, using white text.

This code achieves responsive text color adjustments using Tailwind's built-in screen adaptation without requiring additional media queries or custom CSS.

**Some Example Code Snippets**

**Navbar with Tailwind**

<!doctype html>

<html>

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="./output.css">

</head>

<body class="">

  <nav class="bg-blue-500"> <div class="max-w-7xl mx-auto px-2 sm:px-4">

    <div class="flex items-center justify-between h-16">

      <div class="flex items-center">

        <a href="#" class="flex-shrink-0">

          <img class="h-8 w-8" src="logo.svg" alt="Logo">

        </a>

      </div>

      <div class="hidden sm:block">

        <a href="#" class="ml-auto px-3 py-2 text-white font-medium">Home</a> <a href="#" class="ml-4 px-3 py-2 text-white font-medium">About</a>

        <a href="#" class="ml-4 px-3 py-2 text-white font-medium">Services</a>

        <a href="#" class="ml-4 px-3 py-2 text-white font-medium">Contact</a>

      </div>

      <div class="-mr-2 flex items-center sm:hidden">

        <button type="button" class="inline-flex items-center justify-center p-2 rounded-md text-white hover:bg-blue-400 focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white" aria-controls="mobile-menu" aria-expanded="false">

          <span class="sr-only">Open menu</span>

          <svg class="h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke="currentColor" aria-hidden="true">

            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16" />

          </svg>

        </button>

      </div>

    </div>

  </div>

</nav>

<div class="sm:hidden" id="mobile-menu">

  <div class="px-2 pt-2 pb-3 space-y-1">

    <a href="#" class="block px-3 py-2 text-base font-medium text-white">Home</a>

    <a href="#" class="block px-3 py-2 text-base font-medium text-white">About</a>

    <a href="#" class="block px-3 py-2 text-base font-medium text-white">Services</a>

    <a href="#" class="block px-3 py-2 text-base font-medium text-white">Contact</a>

  </div>

</div>

</body>

</html>

**Grid Example Code With cards.**

<div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4">

<div class="card bg-white rounded-lg shadow-md p-4">

<img src="image1.jpg" alt="Image 1" class="w-full h-48 object-cover rounded-t-lg">

<div class="p-4">

<h3 class="text-xl font-bold text-gray-700">Card Title 1</h3>

<p class="text-gray-600">Card description goes here.</p>

</div>

</div>

<div class="card bg-white rounded-lg shadow-md p-4">

<img src="image2.jpg" alt="Image 2" class="w-full h-48 object-cover rounded-t-lg">

<div class="p-4">

<h3 class="text-xl font-bold text-gray-700">Card Title 2</h3>

<p class="text-gray-600">Card description goes here.</p>

</div>

</div>

<div class="card bg-white rounded-lg shadow-md p-4">

<img src="image3.jpg" alt="Image 3" class="w-full h-48 object-cover rounded-t-lg">

<div class="p-4">

<h3 class="text-xl font-bold text-gray-700">Card Title 3</h3>

<p class="text-gray-600">Card description goes here.</p>

</div>

</div>

<div class="card bg-white rounded-lg shadow-md p-4">

<img src="image4.jpg" alt="Image 4" class="w-full h-48 object-cover rounded-t-lg">

<div class="p-4">

<h3 class="text-xl font-bold text-gray-700">Card Title 4</h3>

<p class="text-gray-600">Card description goes here.</p>

</div>

</div>

</div>