# day-8-L1A

# Tailwind CSS with Astro: A Beginner's Guide

## Overview of Today

- □ Understanding Tailwind CSS and its utility-first approach
- □ Core concepts and basic utilities
- □ Setting up Tailwind with Astro
- □ Customizing Tailwind configuration
- □ Building responsive designs
- □ Best practices and common patterns

---

## Resources that'll help you

- Tailwind in 100 Seconds - Quick overview
- Tailwind CSS Documentation - Official documentation (one of the best docs)
- Tailwind CSS Cheat Sheet - Quick reference
- Tailwind with Astro Installation Guide - Official guide from Tailwind to Integrate with Tailwind V4
- Tailwind Colors - The Tailwind default Colors, visualized for copy paste

---

## Understanding Tailwind CSS

Tailwind CSS is a utility-first CSS framework that allows you to build custom designs directly in your HTML/JSX using predefined classes.

(*Highly recommended to skip things below and focus mainly on the videos and tailwind docs above*)

### Core Concepts

1. **Utility-First Approach**:
    - Small, single-purpose classes

    - Composable designs
    - Less need to write custom CSS
    - Rapid prototyping
2. **Responsive Design**:
    - Mobile-first approach
    - Breakpoint prefixes
    - Flexible layouts
3. **Component Patterns**:

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

## Common Utility Categories

- Layout:
    - display/position
- Typography:
    - Font/text
- Colors:
    - background/text colors
- Spacing
- Flexbox/Grid
- borders

---

# Setting Up Tailwind with Astro

## Installation

Installation Steps are here

> **Important**
>
> Use the Tailwind CSS IntelliSense VS Code extension for better development experience

- In VScode, go to the Extensions, search for tailwind, and install the official Tailwind Plugin

## Basic Usage Examples

- Copy the Code below and run it, notice afterwards the styling in the `class` attribute.
- hover on the any of the String in a random `class` of the ones below

```
---
// src/pages/index.astro
---

<div class="container mx-auto px-4">
  <h1 class="text-3xl font-bold text-center my-8">
    Welcome to My Site
  </h1>

  <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    <div class="bg-white p-6 rounded-lg shadow-md">
      <h2 class="text-xl font-semibold mb-2">Card 1</h2>
      <p class="text-gray-600">Some content here</p>
    </div>
    <!-- More cards... -->
  </div>
</div>
```

### Common Utility Patterns

```
+ Layouts: flex, grid
+ Spacing: p-{size}, m-{size}, gap-{size}
+ Typography: text-{size}, font-{weight}
+ Colors: bg-{color}-{shade}, text-{color}-{shade}
+ Responsive: sm:, md:, lg:, xl:

- Avoid long chains of utilities
- Don't mix Tailwind with regular CSS unless necessary
```

# Exploring Tailwind basics

## Basic Utility Classes

Tailwind uses utility classes that each serve a single purpose. Here are the most common ones:

### Layout & Sizing

```html
<!-- Display -->
<div class="block">Block</div>
<div class="inline">Inline</div>
<div class="flex">Flex container</div>
<div class="hidden">Hidden element</div>

<!-- Width & Height -->
<div class="w-full">100% width</div>
<div class="w-1/2">50% width</div>
<div class="h-screen">Full viewport height</div>
```

### Typography

```html
<!-- Font Size -->
<p class="text-sm">Small text</p>
<p class="text-base">Base text</p>
<p class="text-lg">Large text</p>
<p class="text-2xl">Even larger</p>

<!-- Font Weight -->
<p class="font-normal">Normal weight</p>
<p class="font-bold">Bold text</p>

<!-- Text Alignment -->
<p class="text-left">Left aligned</p>
<p class="text-center">Centered</p>
```

### Spacing

```html
<!-- Margin -->
<div class="m-4">Margin all sides</div>
<div class="mt-4">Margin top</div>
<div class="mb-4">Margin bottom</div>
```

```
<!-- Padding -->
<div class="p-4">Padding all sides</div>
<div class="px-4">Padding left & right</div>
<div class="py-4">Padding top & bottom</div>
```

---

## Responsive Design

Tailwind uses a mobile-first approach with breakpoint prefixes:

```
<!-- This element will: -->
<div class="
  w-full          <!-- Be full width on mobile -->
  md:w-1/2       <!-- Be half width on medium screens -->
  lg:w-1/3       <!-- Be one-third width on large screens -->
">
```

### Breakpoint Reference

```
+ sm:   640px  (Small screens)
+ md:   768px  (Medium screens)
+ lg:   1024px (Large screens)
+ xl:   1280px (Extra large screens)
+ 2xl:  1536px (2X large screens)
```

### Common Responsive Patterns

```
<!-- Responsive Navigation -->
<nav class="
  flex flex-col    <!-- Stack vertically on mobile -->
  md:flex-row     <!-- Row layout on medium+ screens -->
">

<!-- Responsive Grid -->
<div class="
  grid
  grid-cols-1     <!-- Single column on mobile -->
  md:grid-cols-2  <!-- Two columns on medium -->
  lg:grid-cols-3  <!-- Three columns on large -->
">
```

## State Variants

Tailwind provides variants for different states:

```html
<!-- Hover States -->
<button class="
  bg-blue-500
  hover:bg-blue-700
  text-white
  hover:text-gray-100
">

<!-- Focus States -->
<input class="
  border-gray-300
  focus:border-blue-500
  focus:ring-2
">

<!-- Active States -->
<button class="
  bg-blue-500
  active:bg-blue-800
">

<!-- Group Hover -->
<div class="group">
  <p class="group-hover:text-blue-500">
    Changes on parent hover
  </p>
</div>
```

## Color System

Tailwind provides a robust color system with shades:

```html
<!-- Text Colors -->
<p class="text-gray-500">Gray text</p>
<p class="text-blue-600">Blue text</p>
```

```
<!-- Background Colors -->
<div class="bg-red-500">Red background</div>
<div class="bg-green-200">Light green background</div>

<!-- Border Colors -->
<div class="border-2 border-purple-500">
  Purple border
</div>
```

### Color Scale

Each color comes in shades from 50 to 900:

```
+ 50:  Lightest
+ 100: Very light
+ 200: Light
...
+ 900: very very Dark
+ 950: Darkest
```

---

## Arbitrary Values

When you need specific values not in the default scale:

```
<!-- Arbitrary values use square brackets -->
<div class="
  w-[762px]
  text-[22px]
  bg-[#ff6b6b]
">
```

---

## Additional Concepts

### Flex & Grid

```
<!-- Flexbox -->
<div class="
```

```
  flex
  justify-between
  items-center
">

<!-- Grid -->
<div class="
  grid
  grid-cols-3
  gap-4
">
```

## Common Patterns

```
+ Combine classes for complex styles
+ Use group for parent-child interactions
+ Stack responsive classes from mobile up
+ Use arbitrary values sparingly

- Don't mix Tailwind with regular CSS
- Avoid extremely long class strings
- Don't repeat complex patterns
```

## Example Component

```
<button class="
  px-4
  py-2
  bg-blue-500
  text-white
  rounded-lg
  hover:bg-blue-600
  focus:ring-2
  focus:ring-blue-300
  disabled:opacity-50
">
  Click me
</button>
```

## Customizing Tailwind

There are multiple Strategies to customize Tailwind, which include:

- Using arbitrary Values, more info here
- Using Custom CSS, more info here
    - You can either regular Custom CSS:

```
@import "tailwindcss";
.my-custom-style {
/* ... */
}
```

-
  ```
  - or by using `@layer` directives, all details on the doc link above, some important info:
      - the `base` layer means all defined CSS will be added by default to all included Astro Elements
      - the `component` layer is responsible for complicated classes you want to add to your project.
      -
  ```

> **Tip**
> In those Layers, you can write Tailwind classes, by using the `@apply` , Exmaple:

```
@layer components{
  .btn{
-   @apply bg-slate-800 p-4 text-slate-200;
- }
}
```

---

## Tasks

- □ Set up a new Astro project with Tailwind CSS
- □ Create a responsive deck of Cards
    - □ Build the card grid layout to be: 2 Columns on Laptops and 1 on Mobile
- □ Create any reusable component using Astro and Tailwind

---

# Development Workflow

Tip

Group related utilities with @apply in your components when patterns become repetitive

## Example Component Pattern

`src/components/Card.astro`

```astro
---
interface Props {
  title: string;
  description: string;
}

const { title, description } = Astro.props;
---

<div class="
  bg-white
  rounded-lg
  shadow-md
  p-6
  hover:shadow-lg
  transition-shadow
  duration-300
">
  <h2 class="text-xl font-semibold mb-2">{title}</h2>
  <p class="text-gray-600">{description}</p>
</div>
```