

React-4 - Tailwind and Weather App

Agenda

- tailwind setup
- intro
- creating weather app

Notes : Please watch the lecture to understand How weather app is created . You will learn how html css js based application's code flow differs from react application

steps to add Tailwind to react

1. In your react app go to terminal and type two command

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

2. A new file Tailwind.config.js will be created . Replace this content

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*..{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

3. go to `index.css` and on top add these three lines

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Resources

- Tailwind Installation Guide : <https://tailwindcss.com/docs/guides/vite>
- Tailwind Official Website : <https://tailwindcss.com/docs/utility-first>
- Tailwind cheat sheet : <https://nerdcave.com/tailwind-cheat-sheet>

VS code Extension

- Tailwind intellisense
- inline fold-> hide your tailwind classes
- ES7+ React/Redux/React-Native snippet

Tailwind CSS Revision Notes

1. Utility-First Principle

Tailwind CSS is a utility-first framework, which means it provides low-level utility classes to build custom designs directly in your HTML or JSX. Each class corresponds to a single CSS property or a combination of them.

2. Basic Concepts

Classes for Common CSS Properties:

- **Color:**

- **Text color:** `text-{color}-{shade}`
 - Sets the color of the text.
- **Background color:** `bg-{color}-{shade}`
 - Sets the background color of an element.
- **Border color:** `border-{color}-{shade}`
 - Sets the color of the border.

```
<div className="text-blue-500 bg-red-100 border-green-300">  
  Color Example  
</div>
```

- **Effect:**

- The text will be blue with a medium shade.
- The background will be a light shade of red.

- The border will be a medium shade of green.

• Spacing:

- **Padding:** `p-{size}`, `pt-{size}`, `pr-{size}`, `pb-{size}`, `pl-{size}`, `px-{size}`, `py-{size}`
 - Adds space inside the element.
 - `p-4`: Adds padding of 1rem (16px) on all sides.
 - `m-2`: Adds margin of 0.5rem (8px) on all sides.

```
<div className="p-4 m-2">
  Spacing Example
</div>
```

• Effect:

- The element will have 1rem (16px) padding on all sides.
- The element will have 0.5rem (8px) margin on all sides.

• Sizing:

- **Width:** `w-{size}`
 - Sets the width of an element.
- **Height:** `h-{size}`
 - Sets the height of an element.
- **Max-width:** `max-w-{size}`
 - Sets the maximum width of an element.
- **Max-height:** `max-h-{size}`
 - Sets the maximum height of an element.

```
<div className="w-1/2 h-32 max-w-xs">
  Sizing Example
</div>
```

• Effect:

- The element will have a width of 50% of its parent.
- The element will have a fixed height of 8rem (128px).
- The element's width will not exceed 20rem (320px).

• Typography:

- **Font size:** `text-{size}`

- Sets the font size of the text.
- **Font weight:** `font-{weight}`
 - Sets the thickness of the font.
- **Line height:** `leading-{size}`
 - Sets the space between lines of text.
- **Text alignment:** `text-{left|center|right|justify}`
 - Aligns the text horizontally.

```
<div className="text-lg font-bold leading-tight text-center">
  Typography Example
</div>
```

- **Effect:**
 - The text will have a font size of 1.125rem (18px).
 - The text will be bold.
 - The lines will have minimal space between them.
 - The text will be centered horizontally.

• Flexbox:

- **Display:** `flex`
 - Applies a flexbox layout to the element.
- **Flex direction:** `flex-{row|col}`
 - Sets the direction of the flex items.
- **Justify content:** `justify-{start|center|end|between|around|evenly}`
 - Aligns flex items along the main axis.
- **Align items:** `items-{start|center|end|stretch|baseline}`
 - Aligns flex items along the cross axis.
- **Gap:** `gap-{size}`
 - Sets the space between flex items.

```
<div className="flex flex-col justify-center items-center gap-4">
  Flexbox Example
</div>
```

- **Effect:**
 - The element will have a flex layout with a vertical direction.

- Flex items will be centered along both axes.
- There will be a gap of 1rem (16px) between flex items.

• Positioning:

- **Position:** `static`, `relative`, `absolute`, `fixed`, `sticky`
 - Sets the position type of an element.

```
<div className="relative top-2 left-4">
  Positioning Example
</div>
```

• Effect:

- The element will be positioned relative to its normal position.
- It will be moved 0.5rem (8px) from the top and 1rem (16px) from the left.

• Borders:

- **Border width:** `border`, `border-{size}`, `border-t-{size}`, `border-r-{size}`, `border-b-{size}`, `border-l-{size}`
 - Sets the width of the border on all or specific sides.
- **Border radius:** `rounded`, `rounded-{size}`, `rounded-t-{size}`, `rounded-r-{size}`, `rounded-b-{size}`, `rounded-l-{size}`
 - Sets the rounding of the corners of the element.
- **Border style:** `border-{style}`
 - Sets the style of the border (e.g., solid, dashed).

```
<div className="border-4 border-dashed rounded-lg">
  Borders Example
</div>
```

• Effect:

- The element will have a border with a width of 4px.
- The border will be dashed.
- The corners will be rounded with a large radius.

• Backgrounds:

- **Background size:** `bg-{auto|cover|contain}`
 - Sets the size of the background image.
- **Background position:** `bg-{center|top|right|bottom|left}`
 - Sets the position of the background image.

```
<div className="bg-cover bg-center" style={{ backgroundImage:
'url(path/to/image.jpg)' }}>
  Background Example
</div>
```

- **Effect:**

- The background image will cover the entire element.
- The background image will be centered.

- **Effects:**

- **Box shadow:** `shadow`, `shadow-{size}`

- Adds a shadow effect to the element.

- **Opacity:** `opacity-{value}`

- Sets the transparency level of the element.

```
<div className="shadow-lg opacity-75">
  Effects Example
</div>
```

- **Effect:**

- The element will have a large shadow.
- The element will be 75% opaque (25% transparent).

- **Transitions & Animations:**

- **Transition:** `transition`, `transition-{property}`, `duration-{time}`, `ease-{type}`

- Adds a transition effect when properties change.

```
<div className="transition duration-500 ease-in-out transform hover:scale-
105">
  Transitions Example
</div>
```

- **Effect:**

- The element will have a transition effect applied to all properties.
- The transition will last 500ms.
- The transition will ease in and out smoothly.
- On hover, the element will scale up to 105% of its original size.

- **Pseudo-classes:**

- **Hover:** `hover:{utility}`
 - Applies styles when the element is hovered.
- **Focus:** `focus:{utility}`
 - Applies styles when the element is focused.
- **Active:** `active:{utility}`
 - Applies styles when the element is active (clicked).

```
<button className="bg-blue-500 hover:bg-blue-700 focus:bg-blue-900  
active:bg-blue-600">  
  Pseudo-classes Example  
</button>
```

- **Effect:**
 - The button will have a blue background.
 - On hover, the background will change to a darker blue.
 - On focus, the background will change to an even darker blue.
 - When active, the background will be a medium-dark blue.