TAILWIND

SHEERAZ MEHBOOB

Contents

TailWind CSS

- Uses an approach to build Web Apps called Utility First approach.
- You can create a unique Web app without using CSS
- Unlike Bootstrap Tailwind has no classes, it only has a responsive grid and flex box classes

→ You can either download GitHub repository or setup the environment by creating a new folder tailwind that contains "index.html" file.

Requirements:

- NPM
- NPX (use to rum modules without actually install them but install temporary copy of any package)
- Git
- Install Tailwind
    - ❖ Tailwind JIT CLI
        - ▪ Method -1:
            1. Command: npx tailwindcss -o ./css/styles.css --jit --purge "./**/*.html" – watch
            2. --purge (automatically find all classes in html files)
            3. JIT (Just in Time Compiler)
            4. --watch (lookup for changes in files)
        - ▪ Method -2:
            1. npm init -y
            2. npm install -D @tailwindcss/jit tailwindcss postcss
            3. npx tailwindcss init –p
            4. customize tailwind.config.css and package.json accordingly
            5. run npm start
    - ❖ CDN (not preferred when working on project as it is not fully customizable)
        - ▪ No Customization
        - ▪ No Plugins
        - ▪ No Tree Shaking (No automatic detection whether a part of code is removed or added)
        - ▪ No JIT

Styles:

Custom Builds

By default, styles are defined like break points but we can customize them or add new ones. We can do this by extend the portion of "config" file by adding additional elements to existing object.

- New:

```
theme: {
  extend: {
    screens: {
      medium: '1200px'
    }
  },
}
```

- Customize:

```
theme: {
  extend: {
  },
screens: {
      sm: '640px',
      md: '768px',
      lg: '1024px',
      xl: '1280px',
      '2xl': '1536px'
```

```
        }
}
```

Adding Styles

- Tailwind CSS cause crazy amount of classes but we can create our own custom classes.
- Add a new file 'styles.css' into 'src' folder. The styles.css is input file that is preprocessing for tailwind i.e. Tailwind will look into this input file to override the existing tailwind classes and output the styles based on that.

  ➔ First of all we will incorporate following

```
@tailwind base: (Series of styles Tailwind need to reset how things looks on
browser).
 @tailwind components: This directive in styles.css tells Tailwind CSS to
include all of the styles for its built-in components in your compiled CSS.
 @tailwind utilities: Has the styles that Tailwind CSS needs to output certain
utilities.
```

In order to do all this we need two commands @**layer** and @**apply**. @**layer** allows us to choose which layer to target like headline etc and to add components just simply put those in components. @apply is used when you want to apply a specific Tailwind CSS style.

  ➔ Modify your packge.json file as:

```
  "scripts": {
    "start": "tailwindcss -i ./src/styles.css -o ./build/css/styles.css --
watch"
},
```

  ➔ Stop the server using "Ctrl + C" command and restart it by "npm start".
  ➔ Now just simply modify your input file and see the changes.

- We can even write simple css to create our own custom classes or customize the elements.

"Now on we will discuss classes so we might not have much of the documentation ahead."

Fonts

  ➔ Your font may not have all sort of styling like all weights etc.

Font Family:

- We can modify or add any font.
- Add fontFamily in config.js file just after theme tag as:

```
theme: {
  fontFamily: {
    'sans': ['Helvetica', 'Arial', 'sans-serif'],
    'display': ['Custom Font', 'Other Font'],
  }
}
```

You can replace the array with any of your desired fonts.

Font Size:

text-size – size will be replaced by any of the tag like 'sm', 'lg', 'xs' etc.

Font Weight:

font-weight – weight will be replaced by any of the weights like font-100, font-200, font-300 or font-normal etc.

Font Style:

This contains only **italic** or **not-italic**.

Text

Color:

"text-color-shade" however shade is optional like color-red-500 or color-transparent etc.

Text Opacity:

"text-opacity-amount" to control the transparency of the text like text-opacity-40.
The opacity amount ranges from 0 to 100.

Text Alignment:

"align-position" replace the position with left, right, center etc.

Text Decoration:

"underline", "line-through", "no-underline" are the few classes to decorate the text.

Text Transform:

"uppercase", "lowercase", "capitalize", "normal-case" are the few classes to transform the text.

Text Overflow:

"truncate", "overflow-ellipsis", "overflow-clip" are few classes. Overflow is used to specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

List Styling

In tailwind list are shown without bullet points or numbers and any indentation. So, to style list we will be discussing few classes.

List Style Type:

"list-style" replace style by none, disc, decimal are the classes for styling bullets in list.

List Style Position:

"list-position" Adds indentation to the list like inside, outside.

Spacing Typography:

The style to control spacing between lines, characters etc.

Line Height:

In tailwind it is called leading. "leading-size" size will be replaced by 3,4,5 to 10.
We can also use relative line height to make it relative to the root or current container like "leading-none", "leading-snug", "leading-relaxed", "leading-normal", "leading-loose" etc.

Letter Spacing:

Space between the letters also called tracking. "tracking-size" and size is replaced by tighter, tight, normal, wide, wider, widest.

Borders

Border Width:

"Border-side-amount" like border-r-2
Sides: l, r, b, t
Amounts: 0, 2, 4, 8
"border-amount" will add border on all size of specified weight. Default weight is 1px.

Border Opacity:

"border-opacity-amount" where amount ranging from 0 to 100 by the difference of 5 like border-opacity-50.

Border Color:

"border-color-strength" color are the default like white, blue, red etc. and strength of the color ranges from 100,200 to 900.

Border Radius:

"rounded-side-amount" where sides are t, l, b, r, tl, tr, bl, br etc. and amount can be none, full, sm, md, lg, xl, 2xl etc.

Border Style:

"border-style" where styles are none, solid, dashed, dotted, double etc.

Rings

Rings are similar to borders but they have shadows and some other options.

Ring Width:

"ring-amount" amount ranges from 0 to 8 and an "inset". And by default if amount not specified it will be 3.

Ring Opacity:

"ring-opacity-amount" amount ranges from 0 to 100.

Ring Color:

"ring-color-strength" color is already defined colors like red, blue etc. and strength is color strength ranges from 100 to 900.

Ring Offset Width:

This add a small border around the rings. "ring-offset-amount" amount ranges from 0 to 8.

Ring Offset Color:

In addition to offset you can control the color of that offset as well. "ring-offset-color-thickness" thickness is strength of color ranges from 100 to 900.

Divide

Allows to setup borders between elements. "divide-direction-amount" direction can be x or y axis and amount ranges from 0 to 8 and reverse. "reverse" is used with flex reverse-column or row property.

Divide Opacity:

"divide-opacity-amount" amount ranges from 0 to 100.

Divide Color:

"divide-color-strength" color can be red, blue, green etc. while strength ranges from 100 to 900.

Divide Styles:

"divide-style" styles can be none, solid, dashed, doted or double.