

Date: 19/04/2024 Lecture 9.8 Day: Friday

Tailwind CSS and Bootstrap

What ~~are~~ is Tailwind CSS.

Tailwind CSS is indeed a utility-first CSS framework ~~not~~.

Tailwind CSS offers a similar approach to styling as described earlier providing a set of utility classes to apply styling directly in HTML markup.

Bootstrap as previously mentioned is CSS framework used for responsive website.

Tailwind CSS is a utility-first CSS framework that provides a set of ~~per~~ pre-designed utility classes to style your HTML element directly in the markup. Instead of writing custom CSS, you use these utility classes to apply styles as margin, padding, text alignment, color and more.

Here a brief overview of tailwind CSS and where you might use it

1 Utility - First approach

Tailwind CSS promotes a utility first approach to styling where you apply styles directly to HTML elements using classes like "text-center", "bg-blue-500", "p-4", etc. This approach eliminates the need to write custom CSS and allows for rapid development and prototyping.

② Responsive design: Tailwind CSS includes responsive design utility that make it easy to create layout that adapt to different screen size, you can use "sm", "md", "lg" and "xl" to apply "rule" to apply style specific to different breakpoint

③ Customization: while Tailwind CSS comes with a default set of utility classes this is highly customizable.

you can configure, color, spacing, typography, breakpoint and more to match your project design system.

Developer experience:

- ④ Tailwind CSS can enhance developer experience by providing a consistent and predictable way to style elements across project. It also offers feature like JIT (Just-in-time) mode which speeds up mode which speeds up development by compiling style on-demand.

- ⑤ Integration: Tailwind CSS can be integrated into various front-end frameworks and build tool such as React, Vue.js, Next.js and Webpack making it versatile and adaptable to different project setups.

You might use Tailwind CSS in project where:

Date: _____

Day: _____

- ① Rapid prototyping and iteration are essential.
- ② You want to maintain a consistent design system across your project.
- ③ You prefer a utility-first approach to styling.
- ④ You need to create responsive layout efficiently.
- ⑤ You're working on a project where custom CSS may lead to code duplication or maintenance challenges.

Tailwind CSS can be a powerful tool in the frontend developer arsenal, but it's important to evaluate whether its approach aligns with your project requirements and your team preferences.

Day: _____

Date: _____

Setup Tailwind CSS

- First you can install extension in vs code \Rightarrow Tailwind CSS Intellisense

- Open the Tailwind CSS website
There is multiple way to run Tailwind in HTML
 - Play CDN : use the play to try Tailwind right in the browser without any build step. The play CDN is designed for development purpose only and is not the best choice for production.

Note \Rightarrow Its not recommend.

How we use CDN:

Add the play CDN script to your HTML in the `<head>` of your HTML.

```
<Html>
  <head>
    <Script src = "http://cdn.tailwind
      .css.com"></script>
  </head>
```

Date: 27/11/2021

Day:

How we can use different components of tailwind.

Tailwind can provide different template. That way we can use their website (www.tailblocks.com)

What is tailblock.

Tailblock is a website that offer a collection of pre designed Tailwind CSS components and template.

Here are some reasons.

- ① Time Saving
- ② UI Component
- ③ Customization.

There is also a website that provide plugin of tailwind.css

[Http://play.tailwindcss.com](http://play.tailwindcss.com).

An advanced online playground for tailwind CSS that lets you use all of Tailwind built-time feature directly in browser.

Mr:

Day:

What is npx?
Npx stand for Node Package execute. It is simply an NPM package runner. It allows developers to execute any Javascript package available on the NPM registry without even installing it. Npx is installed automatically with NPM version 5.2.

What is the different between Npx and NVM, and also NPM.
In summary, NPM is the default package manager for Node.js. Npx allows you to run package without installing them globally. NVM allows you to manage different versions of Node.js and YARN is an alternative package manager with additional features.

Start Tailwind

Date: 22/4/2024

Day: ^{Tuesday} AM

Tailwind.css install in
Node.js.

The simplest and fastest way to get up and running with Tailwind CSS from scratch is with the Tailwind CLI tool. The CLI is also available as a standalone executable if you want to use it without installing Node.js.

1 => Install Tailwind.css

Step 1: npm install -D tailwindcss

Step 2 => npx tailwindcss init

② Configure your template path
Add the path to all of your template file in your folder.
create a file which name is "tailwind.config.js"

You can past this

Day:

module.exports = {
 content: ["./**/*.html"],
 theme: "./src/theme.js",
 extend: "./src/extend.js",
 plugins: []
}

Add the Tailwind directive
create a folder which
name is src than you
create a file input.css
copy this and past there.

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

Step 4 =>

npx tailwindcss -i ./src/input.css
-o ./src/output.css --watch

start using Tailwind in your
HTML.

Add your compiled CSS file
to the <head> past the link
link href = "output.css" rel = "stylesheet"
Now

Date: _____

Lecture 99 Day: _____

Lecture 100

In this lecture we are
study and the solution of
Exercise 16 in Lecture 97
Generate Dummy Data.

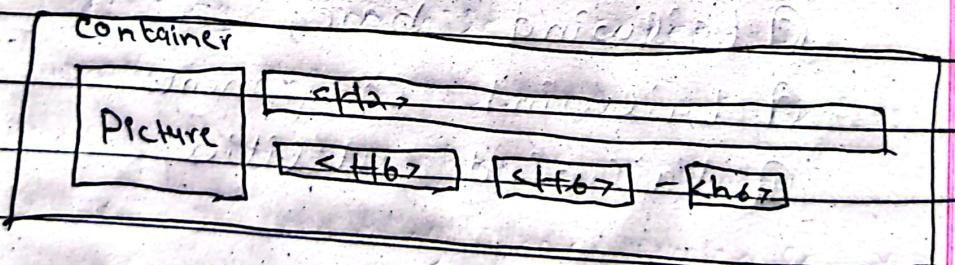
x x x x v

Lecture 100

Exercise 17

Design this lay out

using tailwind CSS design
this cart



Done

Date: 2024/4/12024 Lecture 101 Day: wed

"In this Lecture we understand different utility classes that we use in Tailwind CSS, and also we make a clone of Twitter. First we learn Responsive Design and classes like sm, md, lg, xl, xxl..."

Q What is Responsive Design?
using responsive utility variant to build adaptive user interface.

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

There are five breakpoint by default inspired by common device resolution.

Breakpoint	min width	CSS
'sm'	640px	@media(min-width: 640px)
'md'	768px	
'lg'	1024px	
'xl'	1280px	

Date: 25/4/2024 Day: Thursday

working mobile first

By default, Tailwind uses a mobile-first breakpoint system similar to what you might be used to in other frameworks like Bootstrap.

What this means is that unprefixed utility (like `uppercase` take effect on all screen sizes while prefixed utilities (like `(MD: uppercase)` only take effect at the specified breakpoint.

Targeting mobile screens,

where this approach surprises people most often is that to style something for mobile you need to use the unprefixed version of a utility not the `sm` prefixed version. Don't think of `sm` as meaning "on small screens" think of it as at "the small breakpoint"

Note → Don't use `sm` to target mobile device.

Date: Targeting a breakpoint range
Day:
By default, style applied by rules like "md:flex" will apply at that breakpoint and stay applied at larger breakpoint

If you'd like to apply a utility only when a specific breakpoint range is active, stack a responsive modifier like `md` with a `max-*` modifier to limit that style to a specific range.

For example

```
<div class="md:max-xl:flex">
```

Tailwind generates a corresponding `max-*` modifier for each breakpoint so out of the box the following modifiers are available.

Modifier

`max-sm`

`max-lg`

is so on.

Date: _____

Day: _____

Customizing your theme

You can completely customize your breakpoint is your "tailwind.config.js" file. we can also overriding the default.

| tailwind.config.js

module.exports = {

theme: {

screen: {

'sm': '400px',

'md': '600px',

'lg': '1000px'

}

{

{

Overriding a single screen

To override a single screen size (like 'lg') add your custom 'screen' value under the theme extend key,

extend: {

screen: {

'lg': { 'min': '300px', 'max':

'550px' }

Arbitrary values

If you need to use a one-off breakpoint that doesn't make sense to include in your them we the 'mix' and 'min' modifiers to generate a custom breakpoint on the fly using any arbitrary value.

```
div { class = "min-[320px]:text-center  
max-[600px]:bg-sky-300" } 
```

Padding and margin

How to apply padding?

To apply padding to all sides (left, right, top, and bottom) using Tailwind CSS, you can use the P [size] utility class where '[size]' is a number representing the desired size. You can also use different sides by combining classes.

Date: _____

Day: _____

Here How you can apply padding to all sides equally.

```
<div class = "p-4"></div>
```

This will apply padding of 1 rem (16 pixel by default) to all sides of the element.

For example to apply different padding size to top, right, bottom and left sides respectively.

```
<div class = "pt-2 pr-4 pb-6 pl-8"></div>
```

PT-2 = add padding of 0.5rem (8px)
to top

Pr-4 = add padding 1rem (16px) right

Pb-6 = add padding 1.5rem (24px) bottom

PL-8 = add padding 2rem (32px) left

You can combine these classes as needed to achieve the desired padding for your element.

Day:

Date: How to give margin?
To apply margin to all sides
(left, right, top and bottom)
using Tailwing CSS you can
use the $m = \{size\}$ utility
class where $\{size\}$ is a
number representing the desired
margin size. You can also use
different size for different sides.
by combining classes.

Here how you can apply margin
to all margin to all side equally

Here `<div class="m-4" > </div>`

$mt = \text{margin-top}$

$mr = \text{margin-right}$

$mb = \text{margin-bottom}$

$ml = \text{margin-left}$

$mx = \text{margin-left and Right}$

$my = \text{margin-top and bottom}$

Date: _____

Day: _____

Space Between

To create space between elements using Tailwind CSS, you can use the space-[direction]-[size] utility classes. These classes provide a convenient way to add equal spacing between child elements within a Flex container in the specified direction.

① Space-x-[size]: Adds horizontal space between Flex items.

② Space-y-[size] → Add vertical space between Flex items.

for example =>

```
<div class = "flex space-x-4">  
  </div>
```

The [size] value 1 is equal to 4px.

4px is equal to 1rem.

Day:

width and height

In Tailwind CSS, you can use utility classes to set the width, min-height, max-height, min-width and max-width of element.

D width : you can set the width of an element using the 'w-[size]' utility classes, where [size] is a predefined value or a custom using the 'w-[value] class.

for example,

<div class = "w-8" > width 8 ~~of~~ ~~in~~ </div>

- ⇒ The width size 1 is equal to 4 and 0.25 rem
- ⇒ w-full give full width.
- ⇒ we can also give width-auto
- ⇒ you can also give width percent.

- ⇒ like 50%
⇒ w-1/2 and w-1/2
- ⇒ width-screen used for give 100 ...

Date:

Day:

② Height : Similarly, you can set the height of an element using the `height` utility classes.

For example :

`<div class="h-32"></div>`

The size value as seen is its height value.

③ max-width and max-height

To set the ^xminimum height of an element and also max height.

`max-height = max-h-§7`

`max-width = max-w-§7`

④ min-width and min-height

`min-width = min-w-§7`

`min-height = min-wh-§7`

Day:

Flex box and their property.

In Tailwind CSS, the 'Flex' utility class is used to create flexible layout using CSS Flexbox. Flexbox is a layout model that allows you to design complex layout with ease by distributing space among items in a container along a single axis (either horizontally or vertically) and controlling their alignment, order, and sizing.

Here how you can use the Flex utility class in Tailwind CSS.

① Display flex: The Flex class sets the display property of an element to "Flex" making it a Flex container. This allows its children (Flex item) to be laid out according to the Flexbox.

```
<div class = "flex" ></div>
```

② Flex Direction: By default the 'flex' class sets direction to 'row' meaning that flex

Items are laid out in a horizontal line. You can change the flex direction using additional utility class such as 'flex-col'.

<div class = "flex flex-col">
// For rows you can use flex-row;

- ③ Flex Wrap: The flex-wrap class controls whether flex items are forced onto a single line or can wrap onto multiple line if needed. You can use 'flex-wrap', 'flex-wrap-reverse' or 'flex-wrap-around' to control wrapping behavior.

<div class = "flex flex-no-wrap">
</div>

- ④ Justify Content: The justify-content class aligns flex item along the main axis (horizontal axis by default). You can use value like 'justify-start', 'justify-end', 'justify-center', 'justify-between', 'justify-around' or 'justify-evenly'.

```
div class = "flex justify-center" style="display: flex; justify-content: center;">  
  </div>
```

Align Items: The `item-value` class aligns Flex item alone the cross axis (vertical axis by default). You can use value like `'items-start'`, `'items-end'`, `'item-center'`, `'items-baseline'`, or `'items-stretch'`.

```
<div class = "flex items-end"></div>
```

Align Content: The `content-value` class aligns Flex lines within the Flex container when there extra spaces in the cross axis. You can use value like `'content-start'`, `'content-end'`, `'content-center'`, `'content-between'` or `'content-stretch'`.

```
<div class = "flex flex-wrap content-center" style="display: flex; flex-wrap: wrap; justify-content: center;">  
  </div>
```