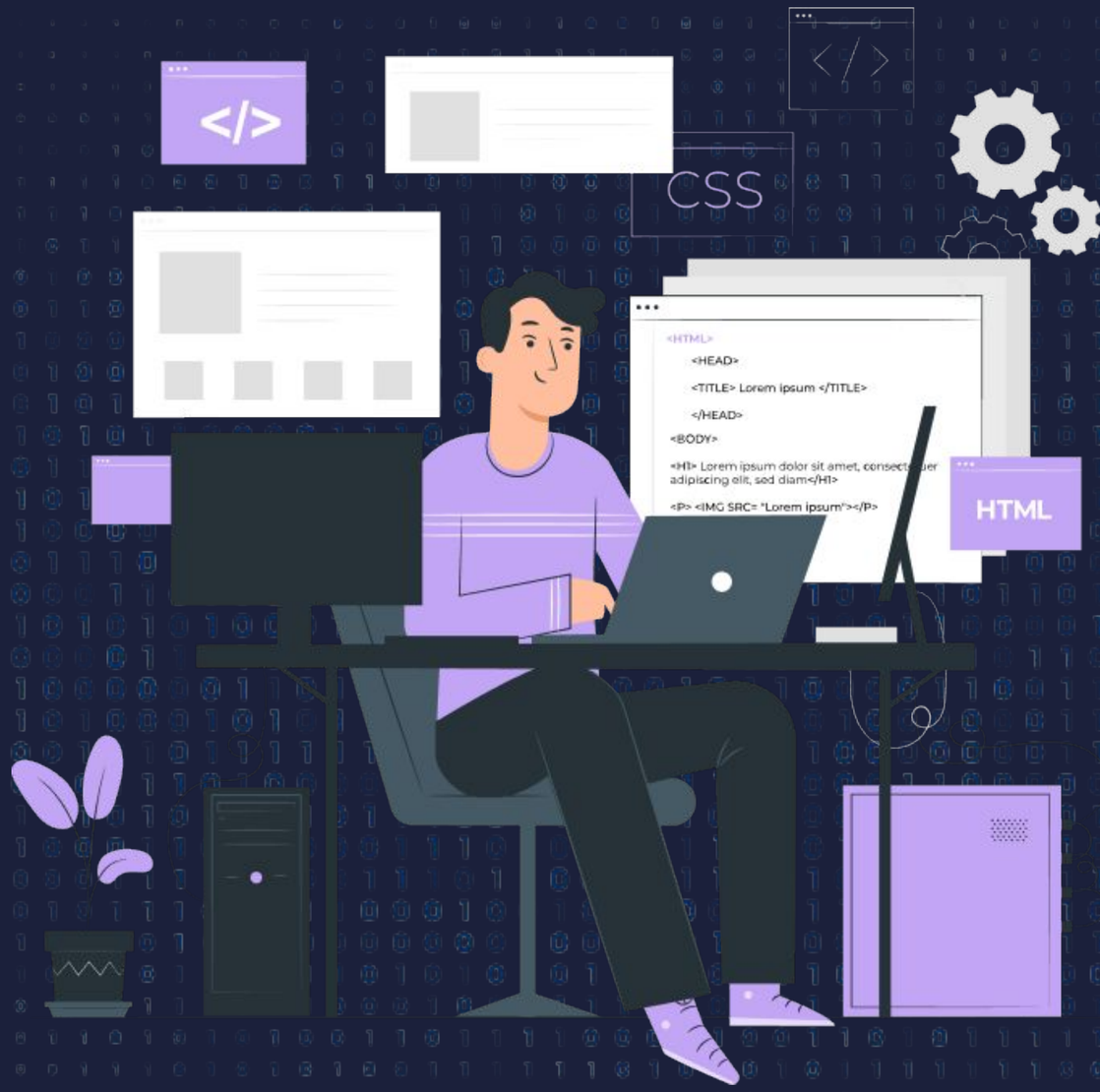
PW SKILLS

Understanding the layout and spacing



Topics

- Width & height
- Padding and Margins
- Display Modes

Width & Height

To apply width and height using tailwind classes, we can use w for width and h for height

Syntax

```
// for width
.w-{size}

// for height
.h-{size}
```

```
<div class="bg-red-600 w-80 h-80"></div>
```

Output

percentage width

```
<div class="bg-red-600 w-3/12">hello</div>
<div class="bg-green-600 w-4/12"> hello</div>
<div class="bg-orange-600 w-4/12">hello</div>
<div class="bg-blue-600 w-6/12">hello</div>
```

Output

hello
hello
hello
hello

w-full v/s w-screen

w-full is nothing but width:100%; in vanilla css while w-screen means width:100vw; in vanilla css

Padding and Margins

To apply padding or margin using tailwind classes, it's very easy

Syntax

```
// for padding
.p-{size}
// for margin
.m-{size}
```

```
//padding top
.pt-{size}

//padding bottom
.pb-{size}

//padding right
.pr-{size}

//padding left
.pl-{size}

//padding on x axis
.px-{size}

// padding on y axis
.py-{size}
```

```
//margin top
.mt-{size}

//margin bottom
.mb-{size}

//margin right
.mr-{size}

//margin left
.ml-{size}

//margin on x axis
.mx-{size}

// margin on y axis
.my-{size}
```

Output


```
<div class="bg-orange-600 w-32 h-32 m-4">I have got some margin</div>
<div class="bg-green-600 w-32 h-32 p-4"> i have got some padding</div>
```

Display Modes

Lets now dive into the display modes in tailwind

Syntax

```
.{display}
```

```
<div class='m-4 w-20 h-20 bg-red-800'></div>
<div class='m-4 w-20 h-20 bg-red-800 hidden'></div>
```

Output

Flexbox

Flex is one of the important topics in Tailwind CSS, flexbox is tailwind is really easy to use and work with.

said in the display modes topic, to apply make a div flex, we just need to add class name flex.

Syntax

```
.flex
```

```
<div class='m-2'>
<div class='w-20 h-20 bg-blue-300 border'>1 without flex</div>
<div class='w-20 h-20 bg-blue-300 border'>2 without flex</div>
<div class='w-20 h-20 bg-blue-300 border'>3 without flex</div>
</div>

<div class='m-2 flex'>
<div class='w-20 h-20 bg-blue-300 border'>4 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>5 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>6 with flex</div>
</div>
```

Output

Flexbox - Justify

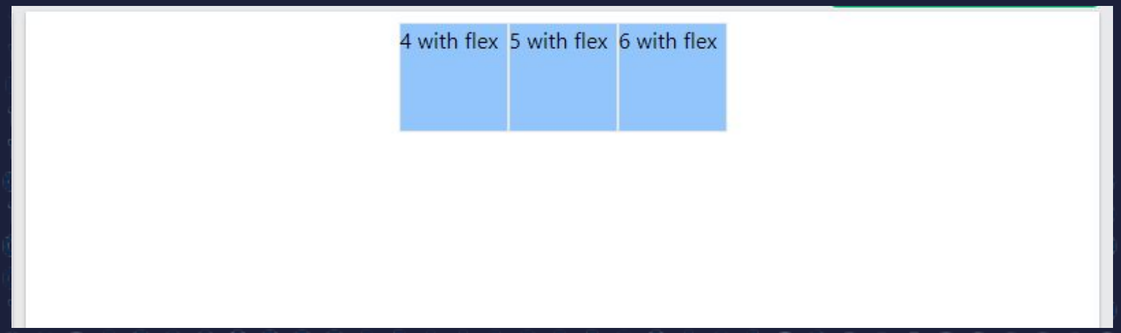
Now let's look how to add justify-content property of a flexbox using tailwind css

Syntax

```
.justify-{alignment}
```

```
<div class='m-2 flex justify-center'>
<div class='w-20 h-20 bg-blue-300 border'>4 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>5 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>6 with flex</div>
</div>
```

Output



Flexbox - Align-items

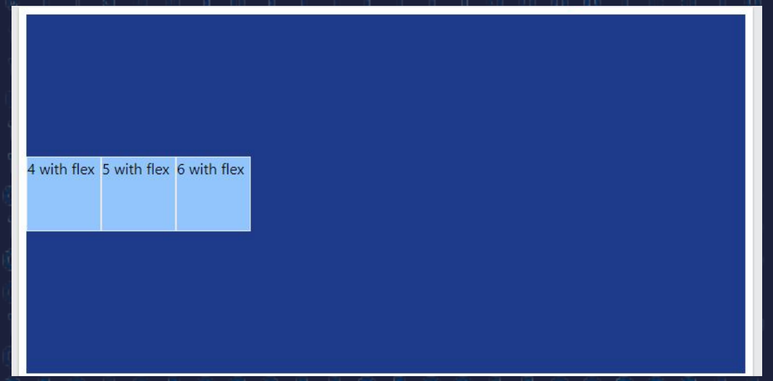
Lets now see how align-items property is added to the flex box using tailwind css

Syntax

```
.items-{alignment}
```

```
<div class='m-2 flex items-center h-96 bg-blue-900'>
<div class='w-20 h-20 bg-blue-300 border'>4 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>5 with flex</div>
<div class='w-20 h-20 bg-blue-300 border'>6 with flex</div>
</div>
```

Output



Flexbox Direction

There are predefined class names for the flexbox direction property as well

Syntax

```
.flex-{direction}
```

Output


```
<div class="border m-2">
      <div>flex-row</div>
      <div class="m-2 flex flex-row">
        <div class="w-20 h-20 bg-blue-300 border">1 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">2 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">3 with flex</div>
      </div>
</div>
<div class="border m-2">
      <div>flex-col</div>
      <div class="m-2 flex flex-col">
        <div class="w-20 h-20 bg-blue-300 border">1 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">2 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">3 with flex</div>
      </div>
</div>
<div class="border m-2">
      <div>flex-row-reverse</div>
      <div class="m-2 flex flex-row-reverse">
        <div class="w-20 h-20 bg-blue-300 border">1 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">2 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">3 with flex</div>
      </div>
</div>
<div class="border m-2">
      <div>flex-col-reverse</div>
      <div class="m-2 flex flex-col-reverse">
        <div class="w-20 h-20 bg-blue-300 border">1 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">2 with flex</div>
        <div class="w-20 h-20 bg-blue-300 border">3 with flex</div>
      </div>
</div>
```

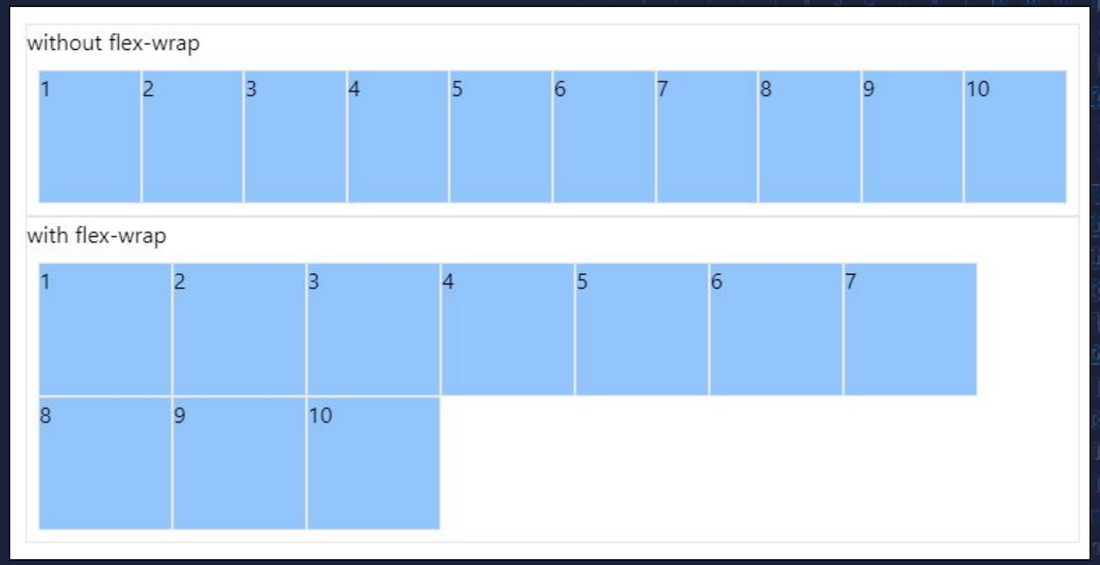
Flexbox Wrap

We can also use the flex wrap property by just adding this tailwind class

Syntax

```
.flex-{wrap}
```

Output




```
<div class="border">
      <div>without flex-wrap</div>
      <div class="m-2 flex">
        <div class="w-24 h-24 bg-blue-300 border">1</div>
        <div class="w-24 h-24 bg-blue-300 border">2</div>
        <div class="w-24 h-24 bg-blue-300 border">3</div>
        <div class="w-24 h-24 bg-blue-300 border">4</div>
        <div class="w-24 h-24 bg-blue-300 border">5</div>
        <div class="w-24 h-24 bg-blue-300 border">6</div>
        <div class="w-24 h-24 bg-blue-300 border">7</div>
        <div class="w-24 h-24 bg-blue-300 border">8</div>
        <div class="w-24 h-24 bg-blue-300 border">9</div>
        <div class="w-24 h-24 bg-blue-300 border">10</div>
      </div>
</div>
<div class="border">
      <div>with flex-wrap</div>
      <div class="m-2 flex flex-wrap">
        <div class="w-24 h-24 bg-blue-300 border">1</div>
        <div class="w-24 h-24 bg-blue-300 border">2</div>
        <div class="w-24 h-24 bg-blue-300 border">3</div>
        <div class="w-24 h-24 bg-blue-300 border">4</div>
        <div class="w-24 h-24 bg-blue-300 border">5</div>
        <div class="w-24 h-24 bg-blue-300 border">6</div>
        <div class="w-24 h-24 bg-blue-300 border">7</div>
        <div class="w-24 h-24 bg-blue-300 border">8</div>
        <div class="w-24 h-24 bg-blue-300 border">9</div>
        <div class="w-24 h-24 bg-blue-300 border">10</div>
      </div>
</div>
```


SKILLS
THANK YOU