Lesson:

Working with transition, animation and transform

PW SKILLS

Topics to be covered

1. Transition
2. Animation
3. Transform

Transition

Transition Properties

Transitions in CSS are used to create smooth animations between different states of an element. In Tailwind CSS, you can apply transitions using the transition utility class along with other classes to control the duration, timing function, and delay.

Syntax:

```
transition-{properties}
```

Properties	Corresponding CSS
none	transition-property: none;
all	transition-property: all; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;
[EMPTY]	transition-property: color, background-color, border-color, text-decoration-color, fill, stroke; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;
colors	transition-property: color, background-color, border-color, text-decoration-color, fill, stroke; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;
opacity	transition-property: opacity; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;
shadow	transition-property: box-shadow; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;
transform	transition-property: transform; transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1); transition-duration: 150ms;

By looking at the above table, you can understand what all css properties will be applied by using the tailwind classes for transition

Example:

```
<a
      href="#"
      class="mx-2 bg-blue-500 hover:bg-blue-700 px-2 py-1 text-white
rounded transition"
    >
      with transition
    </a>
    <a
      href=""
      class="bg-blue-500 hover:bg-blue-700 px-2 py-1 text-white rounded"
      >without transition</a
    >
```

OUTPUT:

In the above example, you can clearly see that there is a minor transforimg effect on the first button. Those little transformations have a lot impact on user experiences.
So adding such transition will help you create a better user experience and tailwind makes our work very easy by providing such utility classes.

Transition Duration
The duration property class in Tailwind CSS is used to control the duration of a transition, which is the amount of time it takes for the transition to complete. By setting the duration, you can control how fast or slow the transition occurs when an element's state changes.

Syntax:

```
duration-{value}
```

value	Amount (Duration in milli seconds)
0	0s
75	75ms
100	100ms
150	150ms
200	200ms
300	300ms
500	500ms
700	700ms
1000	1000ms

Example:

```
<button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
transition duration-150"
    >
      duration-150
    </button>
    <button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300"
    >
      duration-300
    </div>
    <button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-700"
    >
      duration-700
</button>
```

OUTPUT:

In the above example, you can clearly see that the duration of the transition is changing according to the value which we added in each button

Transition Timing Function

The transition timing function is a property that determines how the intermediate states of a transition are calculated over time. It controls the acceleration and deceleration of the transition animation, giving it a specific easing effect. Tailwind CSS provides utility classes to define various timing functions. By default, if you don't specify a timing function, the transition will have a linear timing function. This means that the transition progresses evenly from the starting state to the ending state.

Syntax:

```
ease-{type}
```

type	Properties
linear	transition-timing-function: linear;
in	transition-timing-function: cubic-bezier(0.4, 0, 1, 1);
out	transition-timing-function: cubic-bezier(0, 0, 0.2, 1);
in-out	transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1);

Example:

```
<button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-linear"
    >
      ease-linear
    </button>
    <button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in"
    >
      ease-in
    </div>
    <button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-out"
    >
      ease-out
    </button>
    <button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in-out"
    >
      ease-in-out
</button>
```

OUTPUT:

Transition Delay

Transition delay is a property that allows you to specify a delay before a transition starts. It controls the time between the change in the CSS property and the beginning of the transition animation. In Tailwind CSS, you can use utility classes to define transition delays.

Syntax:

```
delay-{value}
```

value	Amount (Duration in milli seconds)
0	0s
75	75ms
100	100ms
150	150ms
200	200ms
300	300ms
500	500ms
700	700ms
1000	1000ms

Example:

```
<button
      href="#"
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in-out delay-150"
    >
      delay-150
    </button>
    <button
      href=""
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in-out delay-300"
    >
      delay-300
    </div>
    <button
      href=""
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in-out delay-700"
    >
      delay-700
    </button>
    <button
      href=""
      class="bg-blue-500 hover:bg-red-700 px-2 py-1 text-white rounded
duration-300 ease-in-out delay-1000"
    >
      delay-1000
</button>
```

OUTPUT:

From the above example you can clearly see that how the delay classes help use to delay the transition.

Animation

Tailwind CSS comes with a set of pre-built animation utility classes that you can apply to your HTML elements. These classes allow you to achieve various animation effects without writing CSS or JavaScript code.

Syntax:

```
animate-{type}
```

Properties	Corresponding CSS
none	animation: none;

spin	animation: spin 1s linear infinite;
	
	@keyframes spin {
	from {
	transform: rotate(0deg);
	}
	to {
	transform: rotate(360deg);
	}
	}

ping	animation: ping 1s cubic-bezier(0, 0, 0.2, 1) infinite;
	
	@keyframes ping {
	75%, 100% {
	transform: scale(2);
	opacity: 0;
	}
	}

Properties	Corresponding CSS

pulse	animation: pulse 2s cubic-bezier(0.4, 0, 0.6, 1) infinite;
	
	@keyframes pulse {
	0%, 100% {
	opacity: 1;
	}
	50% {
	opacity: .5;
	}
	}

bounce	animation: bounce 1s infinite;
	
	@keyframes bounce {
	0%, 100% {
	transform: translateY(-25%);
	animation-timing-function: cubic-bezier(0.8, 0, 1, 1);
	}
	50% {
	transform: translateY(0);
	animation-timing-function: cubic-bezier(0, 0, 0.2, 1);
	}
	}

Spin

you can add the animate-spin class to make an element spin continuously

```
<div class="flex items-center px-3 py-1 bg-blue-600 text-white rounded">
      Loading
      <img
        class="animate-spin"
        width="30"
        height="30"
        src="https://img.icons8.com/fluency-systems-filled/48/spinner-frame-5.png"
        alt="spinner-frame-5"
      />
    </div>
```

OUTPUT:

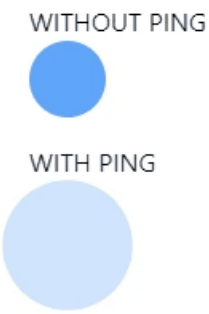
In the above code, we have added a loading image and to that image, we have added the animate-spin class to make that image spin

Ping
Add the animate-ping utility to make an element scale and fade like a radar ping or ripple of water

```
<h2>WITHOUT PING</h2>
   <div class="h-12 w-12 rounded-full bg-blue-400 mb-4"></div>

   <h2 class="mb-4">WITH PING</h2>
   <div
     class="animate-ping duration-2000 h-12 w-12 rounded-full bg-blue-400"
   ></div>
```

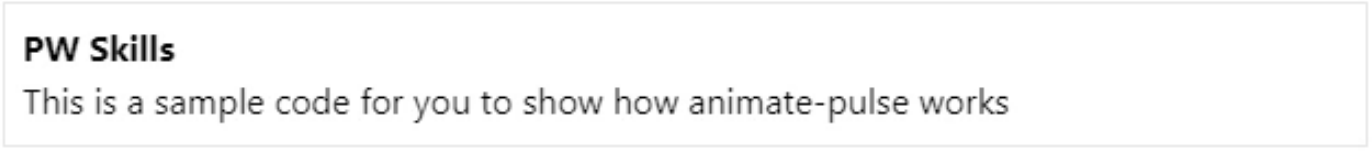
OUTPUT:



Pulse
The "animate-pulse" animation in Tailwind CSS is a simple way to add a pulsating effect to elements on your web page.

```
<div class="border px-2 py-2 animate-pulse">
      <h2 class="font-bold">PW Skills</h2>
      <p>This is a sample code for you to show how animate-pulse works</p>
    </div>
```

OUTPUT:



In the above example, you can see the div and the content inside it giving a pulse animation

Bounce
The "animate-bounce" animation in Tailwind CSS is a simple way to add a bouncing effect to elements on your web page

```
<div class="bg-blue-500 rounded-full w-24 h-24 animate-bounce"></div>
```

OUTPUT:

In the above example we have a div with some width and height, also we have made it round with rounded-full class name and we have added the animate-bounce class name to make this div gives us a bouncing animation

By using all these animations, we can improve the user experience and design of our website.