CSS Transitions

CSS transitions allows you to change property values smoothly, over a given duration.

The CSS transition Property

To create a transition effect, you must specify the CSS property you want to add a transition to, and the duration of the transition.

The CSS transition property is a shorthand property for:

- transition-property (Required)
- transition-duration (Required)
- transition-timing-function
- transition-delay

CSS Transition Example

The following example shows a 100px * 100px element. The element has specified a transition effect for the width property, with a duration of 2 seconds:

Example

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  transition: width 2s;
}
```

How to Trigger the Transition

The transition is triggered when there is a change in the element's properties. This often happens within pseudo-classes (:hover, :active, :focus, or :checked).

So, from the code above, the transition effect will start when the width property changes value.

Now, we add a div:hover class that specifies a new value for the width property when a user mouses over the element:

Example

```
div:hover {
  width: 300px;
}
```

Change Multiple Property Values

You can change multiple properties by separating them by commas.

The following example adds a transition effect for the width, height, and background-color properties, with a duration of 2 seconds for the width, 4 seconds for the height, and 3 seconds for the background-color:

Example

Add a transition effect for the width, height, and background-color properties:

```
div {
  transition: width 2s, height 4s, background-color 3s;
}
```

CSS Transition Speed Curve

The transition-timing-function property specifies the speed curve of the transition effect.

This property can have one of the following values:

- ease - transition will start slow, then go fast, and end slow (this is default)
- linear - transition will keep the same speed from start to end

- ease-in - transition will start slow
- ease-out - transition will end slow
- ease-in-out - transition will have a slow start and end

The following example shows some of the different speed curves that can be used:

Example

Some different speed curves for transition:

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

CSS Transition Delay

The transition-delay property specifies a delay before the transition starts.

The transition-delay value is defined in seconds (s) or milliseconds (ms).

The following example has a 1 second delay before starting:

Example

Add a 1 second delay before starting:

```
div {
  transition-delay: 1s;
}
```

Transition + Transform

The following example combines transition and transform for a :

Example

```
div {
  transition: width 2s, height 2s, background-color 2s, transform 2s;
}
```

The following example combines transition and transform for a button:

Example

```
button {
  transition: background-color 1s ease-out, transform 1s ease-out;
}
```

More Transition Examples

The CSS transition properties can be specified one by one, like this:

Example

```
div {
  transition-property: width;
  transition-duration: 2s;
  transition-timing-function: linear;
  transition-delay: 1s;
}
```

CSS Animations

CSS allows animation of HTML elements without using JavaScript!

What are CSS Animations?

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animation, you must specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

CSS animation-name and animation-duration

The animation-name property specifies a name for the animation.

The animation-duration property defines how long an animation should take to complete. If this property is not specified, no animation will occur, because the default value is 0s (0 seconds).

CSS @keyframes Rule

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

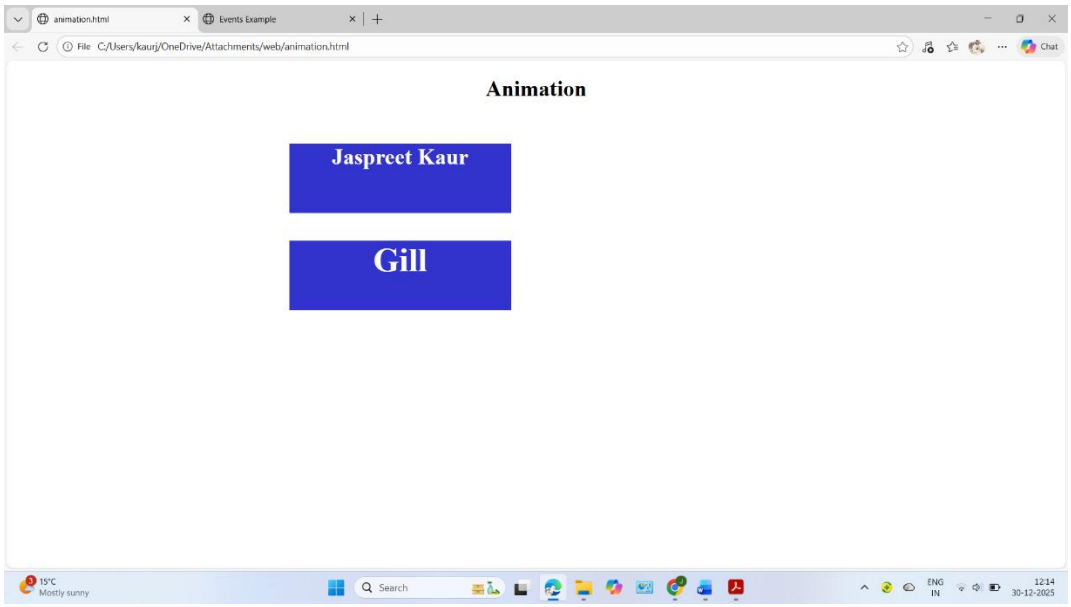
The following example binds the "myAnimation" animation to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow":

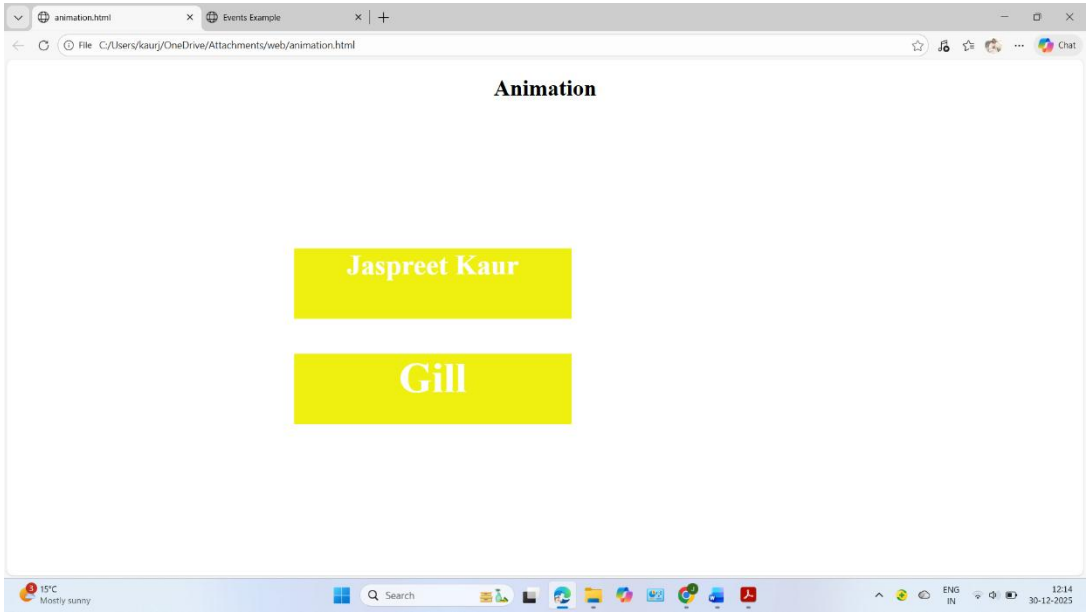
Example

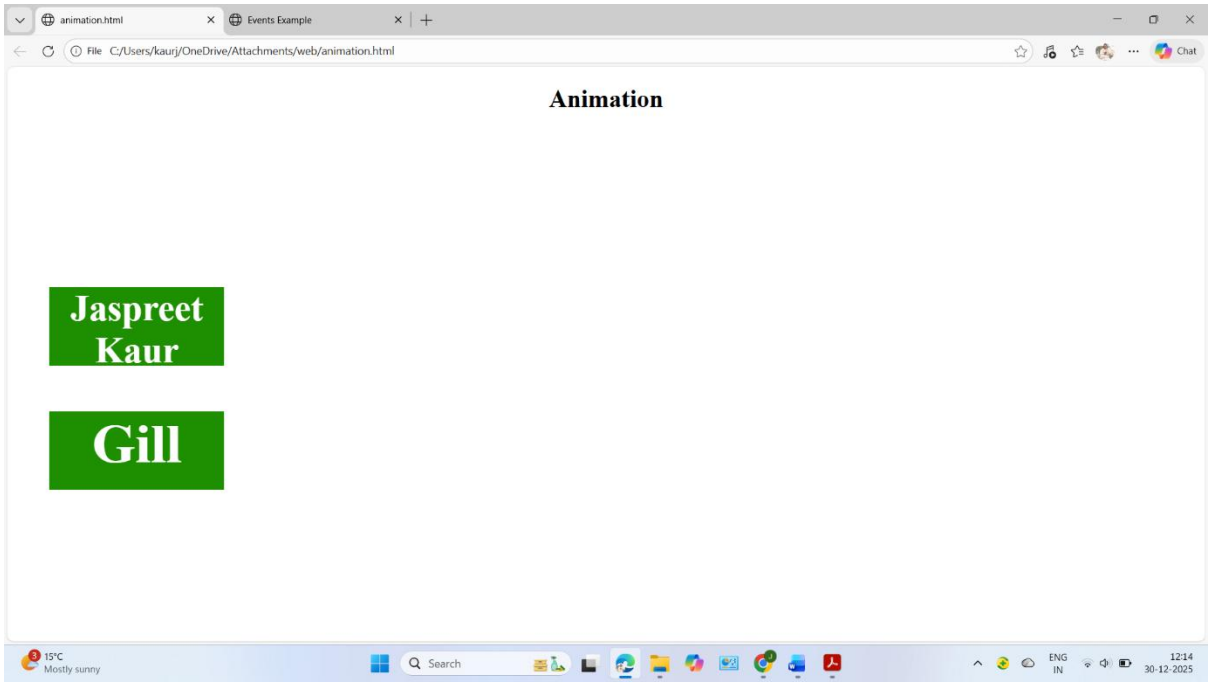
```
/* The animation code */
@keyframes myAnimation {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: myAnimation;
```

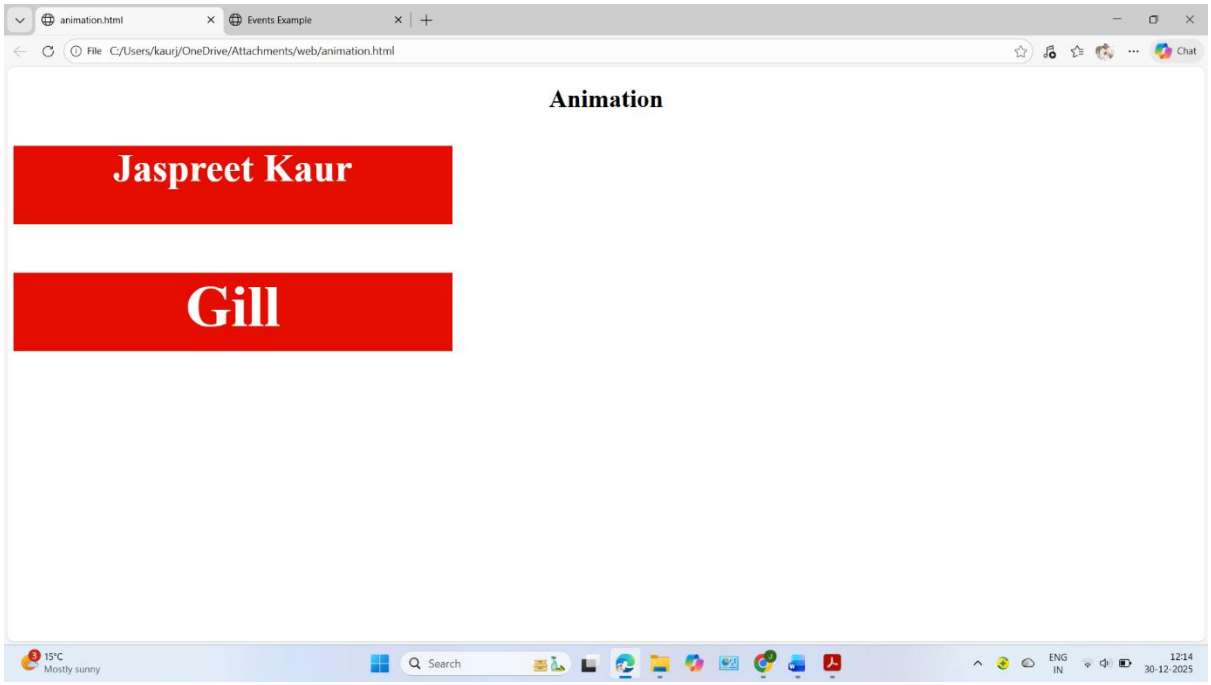
```
  animation-duration: 4s;
}
```

Task: Animation






animation.html
Events Example
File C:/Users/kaurj/OneDrive/Attachments/web/animation.html
Chat
Animation
Jaspreet Kaur
Gill
15°C
Mostly sunny
Search
ENG
IN
12:14
30-12-2025


animation.html
Events Example
File C:/Users/kaurj/OneDrive/Attachments/web/animation.html
Chat
Animation
Jaspreet Kaur
Gill
15°C
Mostly sunny
Search
ENG
IN
12:14
30-12-2025