

## CSS3 Transitions

- The CSS3 transition feature allows the changes in CSS property values to occur smoothly over a specified duration.
- Transitions in CSS allows us to control the way in which transition takes place between the two states of the element. Different states may be defined using pseudo-classes like :hover or :active or dynamically set using JavaScript.
- We can use the transitions to animate the changes, and make the changes visually appealing to the user and hence, giving better user experience and interactivity.
- The "transition" property is specified as one or more single-property transitions, separated by commas.



## CSS Transition Properties

There are four CSS transition properties that we can use for transitions in an element :-

### a) "transition-delay" Property

- This property allows us to determine the amount of time to wait before the transition actually starts to take place.
- The delay may be zero, positive, or negative.
- A value of 0s (or 0ms) will begin the transition effect immediately.
- A positive value will delay the start of the transition effect for the given length of time. Negative values are also allowed for this property.

#### Syntax



where,

a) element = p, div, span etc.

b) value = time

Note :- Here, time can be in seconds(s) or milliseconds(ms), and you should use 's' or 'ms' after the number (without quotes).

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    a
    {
      width: 200px;
      height: 70px;
      background-color: blueviolet;
      color: white;
      padding: 15px;
      text-decoration: none;
      transition: 1s;
    }
    a:hover
    {
      transition: 1s;
      transition-delay: 2s;
      background-color: orangered;
    }
  </style>
</head>
<body>
  <h1 align="center">CSS3 Transition Property -- Delay</h1>
  <center> <a href="#abc">Click Me</a> </center>

</body>
</html>
```

## b) "transition-duration" Property

- This property allows you to determine how long it will take to complete the transition from one CSS property to the other.
- By default, the value is 0s, meaning that no animation will occur.

### Syntax

```
element
{
  transition-duration : value
}
```

where,

a) element = p, div, span etc.

b) value = time.

Note :- Here, time can be in seconds(s) or milliseconds(ms), and you should use 's' or 'ms' after the number (without quotes).

LEARN 2 EARN  
LABS

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    a
    {
      width: 200px;
      height: 70px;
      background-color: blueviolet;
      color: white;
      padding: 15px;
      text-decoration: none;
      transition: 1s;
    }

    a:hover
    {
      transition-duration: 2s;
      background-color: orangered;
    }
  </style>
</head>
<body>
  <h1 align="center">CSS3 Transition Property -- Delay</h1>
  <center> <a href="#abc">Click Me</a> </center>

</body>
</html>
```

### c) "transition-property" Property

- The transition-property CSS property specifies the names of the CSS properties to which a transition effect should be applied.
- This property allows us to select the CSS properties which you want to animate during transition(change).

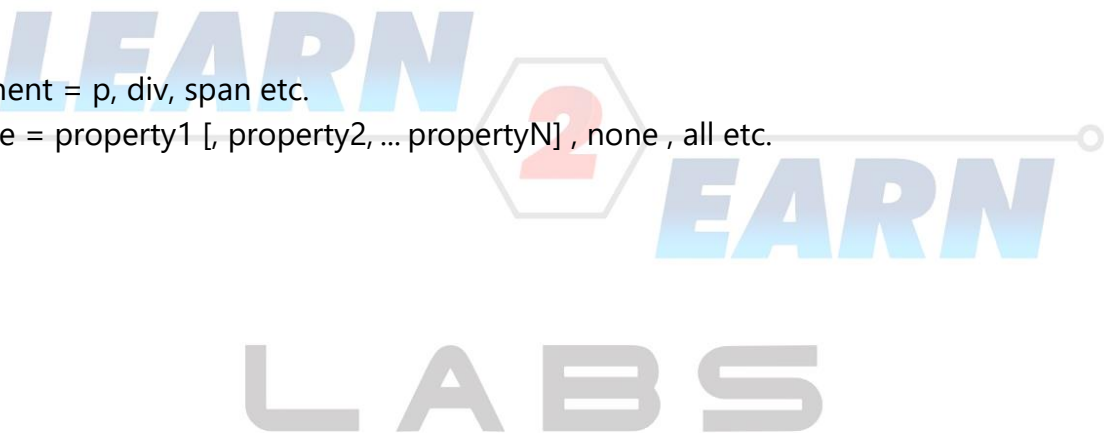
#### Syntax

```
element
{
  transition-property : value
}
```

where,

a) element = p, div, span etc.

b) value = property1 [, property2, ... propertyN] , none , all etc.



## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
  <style>
    .box{
      background-color: red;
      width: 300px;
      height: 200px;
      margin: auto;
      transition-property: background-color, width, height;
      transition-duration: 2s;
    }
    .box:hover{
      background-color: pink;
      width: 200px;
      height: 150px;
    }
    h1, h2{
      color: green;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>CSS3 Transition Property</h1>
  <div class="box"> </div>
</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS transition-property property</title>
  <style>
    .box{
      background-color: red;
      width: 300px;
      height: 200px;
      margin: auto;
      transition-property: all;
      transition-duration: 2s;
    }
    .box:hover{
      background-color: pink;
      width: 200px;
      height: 150px;
    }
    h1, h2{
      color: green;
      text-align: center;
    }
  </style>
</head>

<body>
  <h1>CSS3 Transition Property</h1>

  <div class="box"> </div>
</body>
</html>
```



## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS transition-property property</title>
  <style>
    .box{
      background-color: red;
      width: 300px;
      height: 200px;
      margin: auto;
      transition-property: none;
      transition-duration: 2s;
    }
    .box:hover{
      background-color: pink;
      width: 200px;
      height: 150px;
    }
    h1, h2{
      color: green;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>CSS3 Transition Property</h1>

  <div class="box"> </div>
</body>
</html>
```

#### d) "transition-timing-function" Property

- This property specifies the speed of transition.
- The transition-timing-function CSS property specifies how the intermediate values of the CSS properties being affected by a transition effect are calculated.

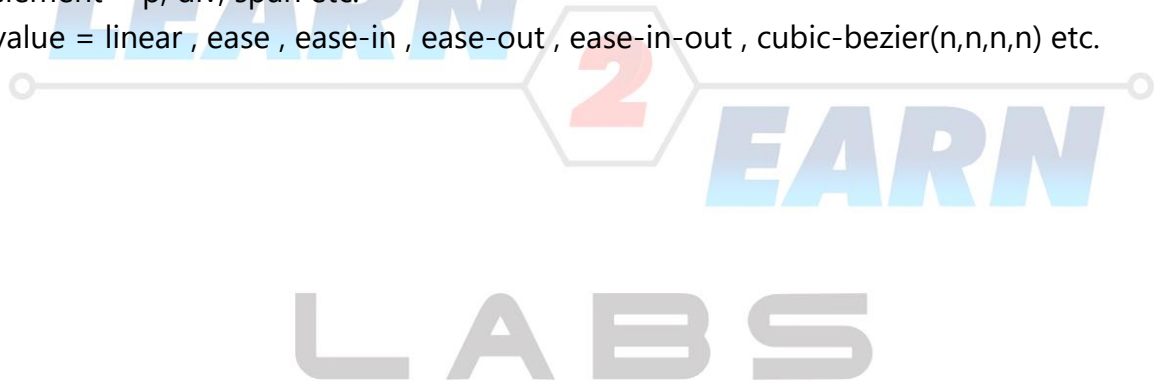
#### Syntax

```
element
{
  transition-timing-function : value
}
```

where,

a) element = p, div, span etc.

b) value = linear , ease , ease-in , ease-out , ease-in-out , cubic-bezier(n,n,n,n) etc.



## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS transition-property property</title>
  <style>
    .box{
      background-color: red;
      width: 500px;
      height: 400px;
      margin: auto;
      transition-property: all;
      transition-duration: 1s;
      transition-timing-function: linear;
    }
    .box:hover{
      background-color: pink;
      width: 200px;
      height: 150px;
    }
    h1, h2{
      color: green;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>CSS3 Transition Property</h1>
  <div class="box"> </div>
</body>
</html>
```

### e) "transition" Property

- The transition CSS property allows you to define the transition between two states of an element. It is a shorthand property for transition-property, transition-duration, transition-timing-function and transition-delay.
- This property is a shorthand property for setting all the four individual transition properties in a single declaration.

#### Syntax

```
element
{
  transition : value
}
```

where,

a) element = p, div, span etc.

b) value = [transition-property transition-duration transition-timing-function transition-delay]

## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS transition-property property</title>
  <style>
    .box{
      background-color: red;
      width: 500px;
      height: 400px;
      margin: auto;
      transition: all 2s ease-out 2s;
    }
    .box:hover{
      background-color: pink;
      transition: 2s;
      width: 200px;
      height: 150px;
    }
    h1, h2{
      color: green;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>CSS3 Transition Property</h1>
  <div class="box"></div>
</body>
</html>
```

## CSS3 Animations

- CSS allows animation of HTML elements without using JavaScript or Flash!
- CSS Animation property is used to create animation on the webpage. It can be used as a replacement of animation created by Flash and JavaScript.
- The animation is created in the @keyframe rule. It is used to control the intermediate steps in a CSS animation sequence.
- In CSS all properties are not animatable. In general, any CSS property that accepts values that are numbers, lengths, percentages, or colors is animatable.
- When the element is set to "position : static" this property will not work.

### CSS Animation Properties

Below are the list of all the animation-related properties that is provided by CSS :-

#### a) "animation-name" Property

The animation-name CSS property specifies the name of @keyframes defined animations that should be applied to the selected element.

#### Syntax

```
element
{
  animation-name : value;
}
```

where,

a) element = p, div, span etc.

b) value = keyframe-name, none.

Note :- The "keyframe-name" is user defined.

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
    }
    @keyframes hello
    {
      from
      {
        left : 0%
      }
      to
      {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div> </div>
</body>
</html>
```

## b) "animation-duration" Property

The animation-duration CSS property specifies the number of seconds (s) or milliseconds (ms) an animation should take to complete one cycle.

### Syntax

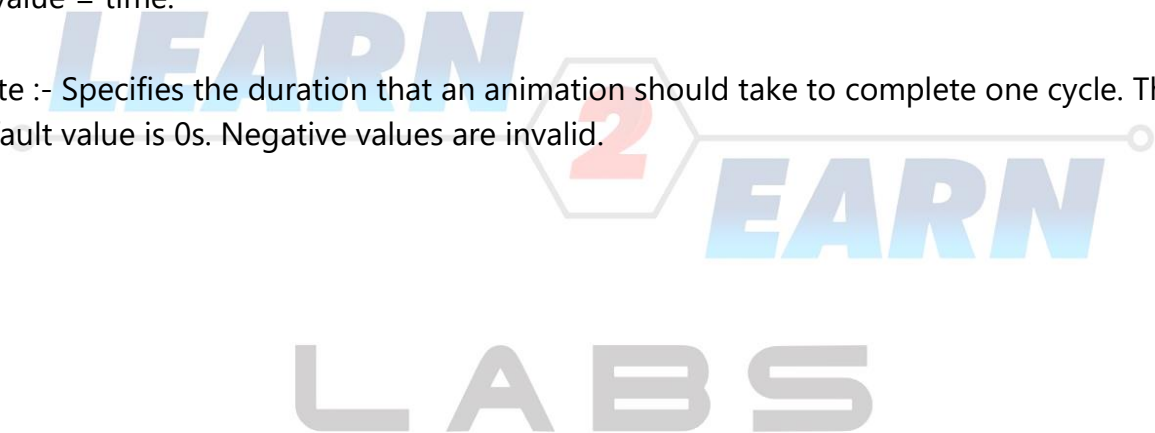
```
element
{
  animation-duration : value;
}
```

where,

a) element = p, div, span etc.

b) value = time.

Note :- Specifies the duration that an animation should take to complete one cycle. The default value is 0s. Negative values are invalid.





## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body>
</html>
```

### c) "animation-timing-function" Property

- The animation-timing-function CSS property specifies how a CSS animation should progress over the duration of each cycle.
- This property specifies the speed curve of the animation.

#### Syntax

```
element
{
  animation-timing-function : value;
}
```

where,

a) element = p, div, span etc.

b) value = linear , ease , ease-in , ease-out , ease-in-out , cubic-bezier(n,n,n,n) etc.

Where,

- 1) ease: The animation starts slowly, then fast, and then finally ends slowly (this is default).
- 2) linear: The animation plays with the same speed from start to end.
- 3) ease-in: The animation plays with a slow start.
- 4) ease-out: The animation plays with a slow end.
- 5) ease-in-out: The animation starts and ends slowly.

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
      animation-timing-function: ease;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body>
</html>
```

#### d) "animation-delay" Property

- The animation-delay CSS property defines when the animation will start.
- The value of this property can be specified in seconds (s) or milliseconds (ms).
- The value of this property is positive as well as negative. A negative value causes the animation to begin immediately, but partway through its cycle.

#### Syntax

```
element
{
  animation-delay : value;
}
```

where,

a) element = p, div, span etc.

b) value = time etc.



## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
      animation-timing-function: linear;
      animation-delay: 2s;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body>
</html>
```

### e) "animation-iteration-count" Property

The animation-iteration-count CSS property sets the number of times an animation cycle should be played before stopping.

#### Syntax

```
element
{
  animation-iteration-count : value;
}
```

where,

a) element = p, div, span etc.

b) value = number , infinite etc.

number - Specifies the number of times an animation should repeat. Default value is 1. Negative values are not allowed.

LEARN 2 EARN  
LABS

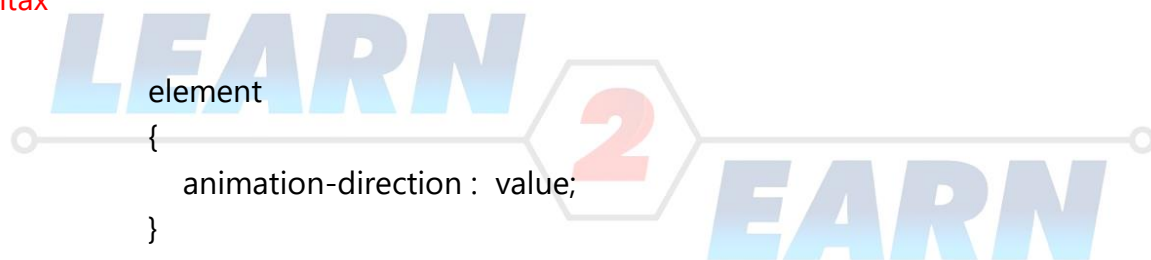
## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
      animation-timing-function: linear;
      animation-delay: 2s;
      animation-iteration-count: 2;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body>
</html>
```

## f) "animation-direction" Property

- This property specifies the direction of the animation.
- The animation-direction CSS property specifies whether the animation should play in reverse on alternate cycles or not.
- The animation-direction property can have the following values:
  - a) normal — (default) - animation is played forward (keyframes 0% - 100%)
  - b) reverse — animation is played backwards (keyframes (100% - 0%))
  - c) alternate — the animation is played forward, then it is reversed and repeated.
  - d) alternate-reverse — the animation is played backwards then forward.

### Syntax



where,

a) element = p, div, span etc.

b) value = normal , reverse , alternate , alternate-reverse etc.



## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
      animation-timing-function: linear;
      animation-delay: 2s;
      animation-iteration-count: 2;
      animation-direction: alternate-reverse;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body>
</html>
```

### g) "animation-fill-mode" Property

- This property specifies a style for the element applied before or after the animation is executed.
- This property specifies what values are applied by the animation before and after it is executing.
- The animation-fill-mode CSS property specifies how a CSS animation should apply styles to its target before and after it is executing.

The animation-fill-mode property can have the following values :-

- **a) forwards** - specifies that the element should keep the style values set by the last keyframe (depends on animation-iteration-count and animation-direction properties).
- **b) backwards** - specifies that the element should get the style values set by the first keyframe (depends on animation-direction) and keep it within animation-delay period.
- **c) both** - specifies that the animation should follow the rules for both forwards and backwards.
- **d) none** - (default) specifies that no style will be applied to the element before or after the animation is executed.

### Syntax

```
element
{
  animation-fill-mode : value;
}
```

where,

a) element = p, div, span etc.

b) value = none, forwards, backwards(default), both etc.

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    div {
      width: 300px;
      height: 300px;
      background-color: blueviolet;
      position: relative;
      animation-name: hello;
      animation-duration: 1s;
      animation-timing-function: linear;
      animation-delay: 2s;
      animation-iteration-count: 2;
      animation-direction: alternate-reverse;
      animation-fill-mode: forwards;
    }
    @keyframes hello {
      from {
        left: 0%
      }
      to {
        left: 50%;
      }
    }
  </style>
</head>
<body>
  <h1>CSS Animation</h1>
  <div></div>
</body></html>
```