**Unit II**

**Animation**

**Animation**: What is an Animation? The Start and End States, Interpolation, Animations in HTML.

**CSS Animations** : All About CSS Animations, Creating a Simple Animation, Detailed Look at the CSS Animation Property, Keyframes, Declaring Multiple Animations, Wrap-up.

**CSS Transitions** :All About CSS Transitions, Adding a Transition, Looking at Transitions in Detail, The Longhand Properties, Longhand Properties vs. Shorthand Properties, Working with Multiple Transitions.

**2.1 Animation**

**2.1.1 Introduction**

* + **An animation is a visualization of change**  that occurs over a period of time.Animations are something primarily in the domain of games, intros, cartoons, banner advertisement.
  + Animations are so deeply ingrained, they make up a large part of our application’s overall user experience. They can make our applications easier to navigate. They help our content be more presentable. They can help our creations feel more alive and fun.
  + An animation lets an element gradually change from one style to another.

While we think of animation as a recent creation brought about by film and computers, people have been fiddling with ways to communicate motion for a long time back.



[ a sequence of pictures from 3000 BC ([wikipedia](http://en.wikipedia.org/wiki/File:Vase_animation.svg)) ]

Some of those ways ranged from cave paintings and elaborate mechanical devices to more contemporary solutions we can relate to such as what we see on television, computers, and smart phones. Today, almost everything we do on a device with a screen is just one click, tap, or keystroke away from springing to life.

**Computer Animations**

**Computer Animations** consists of Three important parts, such as

i) **starting state**

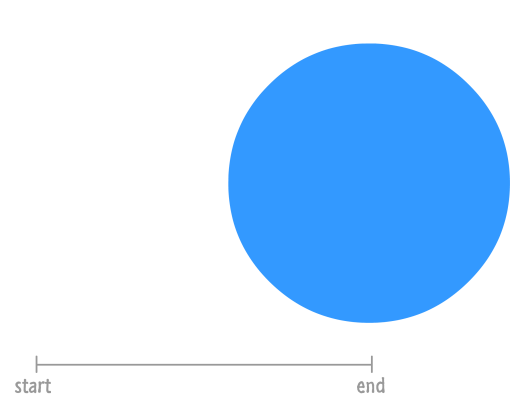
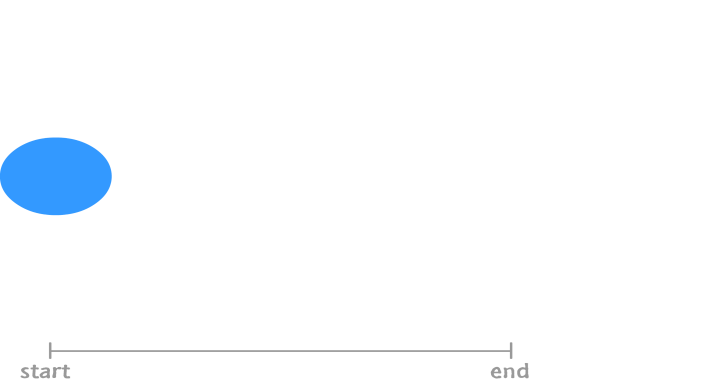
ii) **ending state**

iii) **duration**

 and over which the **transition**. It is occur in between the two states .

**2.1.2 The Start and End States**

**States are** the basic building blocks of  **Animation .**If visualizing change is an important part of an animation, we need to create some reference points so that we can compare what has changed. These reference points the **start** state and the **end** state.



We start off with a blue circle that is small and located to the left of the screen. At the end state, our blue circle now looks bigger and in right side of the screen. There are two notable changes. One change is the position. Our blue circle starts off on the left side of the screen. It ends up on the right hand side. Another change is the size. Our circle goes from being small to being much larger.

To make an animation out of this, we need to just play the start and end states repeatedly, It will give an effectlike something that just bounces from left to right very awkwardly. We need to smooth things out between the start and end states which is called **interpolation**.

**2.1.3 Interpolation**

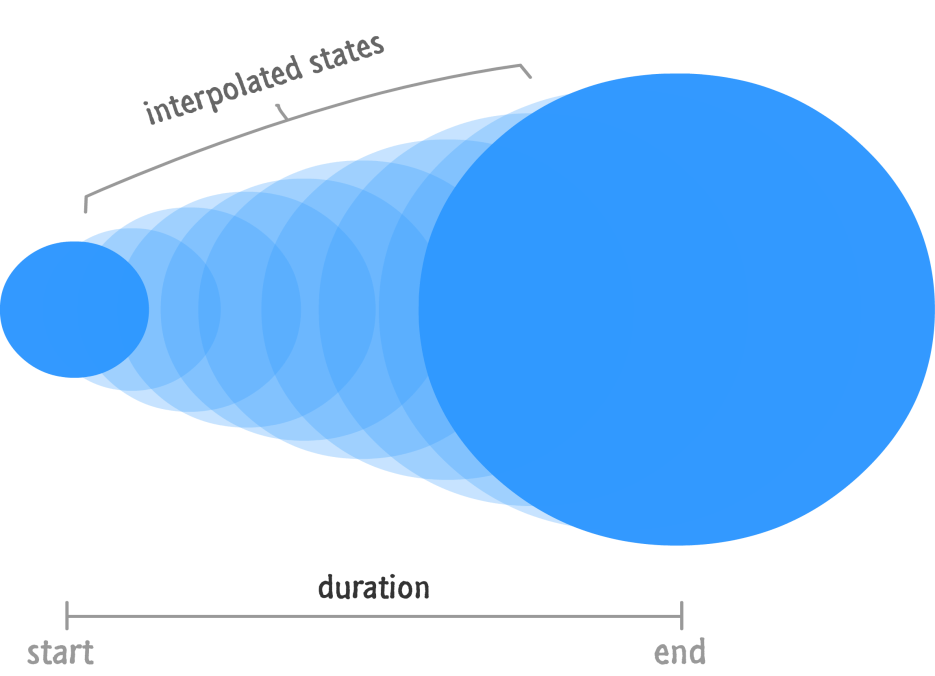
In  [computer animation](https://en.wikipedia.org/wiki/Computer_animation), Interpolation is a method of calculating character poses in the frames between two [keyframes](https://cascadeur.com/help/tools/timeline_tools/keyframes) to create smoother motion.

* Interpolation is the process of estimating unknown values that fall between known values.
* Interpolation animation refers to the technique of creating movement between points.

**Interpolation** is in-between or filling in frames between the key frames. It typically calculates the in-between frames through use of (usually) piecewise [polynomial interpolation](https://en.wikipedia.org/wiki/Polynomial_interpolation) to draw images semi-automatically.

At the beginning,we have our start state. And the end, we have the end state. If we were to play this back, this wouldn't be an animation. In order to make an animation out of what we have, we need a smooth transition that creates all the intermediate states. This creation of the intermediate states is known as **interpolation**.

This interpolation, which occurs over a **period of time that we specify**, would look similar to the following diagram:



For a typical example of 2-D interpolation through key points are

[**nonuniform rational B-spline**](https://en.wikipedia.org/wiki/Nonuniform_rational_B-spline) and  [**Bézier curve**](https://en.wikipedia.org/wiki/B%C3%A9zier_curve)

This is extended to the forming of three-dimensional curves, shapes and complex, dynamic artistic patterns such as used in laser light shows.

**MotionInterpolation**

Interpolation can be extended to motions. The path of an object can be interpolated by providing some key locations, then calculating many in between locations for a smooth motion. In addition to position, the speed or velocity, as well as accelerations along a path, can be calculated to mimic real-life motion dynamics. Where the subjects are too large or complex to move, the camera position and orientation can be moved by this process. This last is commonly called [motion control](https://en.wikipedia.org/wiki/Motion_control).

* **Motion interpolation** or **motion-compensated frame interpolation** (**MCFI**) is a form of [video processing](https://en.wikipedia.org/wiki/Video_processing) in which intermediate animation frames are generated between existing ones by means of [interpolation](https://en.wikipedia.org/wiki/Interpolation), in an attempt to make animation more fluid, to compensate for [display motion blur](https://en.wikipedia.org/wiki/Display_motion_blur), and for fake [slow motion](https://en.wikipedia.org/wiki/Slow_motion) effects.

There are many [techniques that animators can use](https://businessofanimation.com/techniques-of-animation-every-freelancer-needs-to-master/)to create motion in their work. One of these techniques is known as interpolation animation.

Animators use interpolation to create movement between points. It is mainly used to smooth out the motion between movements from point A to point B.

**Morphing**

**Morphing** is a [special effect](https://en.wikipedia.org/wiki/Special_effect) in [motion pictures](https://en.wikipedia.org/wiki/Film) and [animations](https://en.wikipedia.org/wiki/Animation) that changes (or morphs) one [image](https://en.wikipedia.org/wiki/Image) or [shape](https://en.wikipedia.org/wiki/Polygon_mesh) into another through a seamless transition. Traditionally such a depiction would be achieved through [dissolving](https://en.wikipedia.org/wiki/Dissolve_(filmmaking)) techniques on film. Since the early 1990s, this has been replaced by computer software to create more realistic transitions. A similar method is applied to audio recordings, for example, by changing voices or vocal lines.

A particular use for morphing effects is modern digital font design. Using morphing technology, called interpolation or [multiple master](https://en.wikipedia.org/wiki/Multiple_master_fonts) tech, a designer can create an intermediate between two styles, for example generating a semibold font by compromising between a bold and regular style, or extend a trend to create an ultra-light or ultra-bold. The technique is commonly used by font design studios.

### ****Types of Interpolation in Animation****

A type of interpolation animation can be described as a rule used to create in-between frames on an interval. Keyframes set start and end positions, but how the object moves between these positions is defined by the [type of interpolation](https://gisresources.com/types-interpolation-methods_3/).

Here are the seven main types of interpolation animation:

1. **Step interpolation**: Here,in-between values are not calculated; animated objects only change their positions and pose in keyframes.
2. **Linear interpolation**: With this type of interpolation animation, objects move between the specified keyframes with constant speed.
3. **Bezier interpolation**: Objects move along a curve defined by their positions in the keyframes, accelerating at the beginning of the curve and decelerating near the end of it.
4. **Bezier viscous interpolation**: This type works in the same way as Bezier interpolation, but previous movements influence subsequent ones to a lesser degree.
5. **Bezier clamped interpolation**: Objects move linearly but with acceleration and deceleration, and any angles in the curves are heavily smoothed. This type of interpolation animation is best used for ground-level movements or swivel points.
6. **Fixed interpolation**: This type of interpolation animation can be used to adjust poses between keyframes without creating additional keys.
7. **Mixed interpolation**: This type of interpolation animation is used for folders that contain several[animation tracks](https://cascadeur.com/help/tools/timeline_tools/animation_tracks) with different types of interpolation.

**2.1.4 Animations in HTML**

We need to use the <image> tag with the src attribute to add an animated image in HTML. The src attribute adds the URL of the image (file destination). Also, we can set the height and width of the image using the height and width attribute. Here we can add video files( .avi) and audio files (.mp3).

To use a floating image in HTML, use the CSS property float. It allows you to float an image left or right. More property values include the following:

|  |  |
| --- | --- |
| **Float Property** | **Value & Description** |
| float :none | None -Not floated |
| float :left | Left-Floats to the left |
| float:right | Right- Floats to the right |
| float:initial | Initial-Default value |

**Example 2.1 Animation in HTML**

<html>

<head>

<title>Animation in HTML</title>

</head>

<body>

<h1>Animation in HTML</h1>

<imgsrc="spinner.gif" width="190" height="84" style="float:right;"/>

</body>

</html>

**2.2 CSS Animations**

**2.2.1 Introduction CSS Animations**

CSS Animation is the process of animating the objects (or elements) on a web page. Earlier to CSS Animations, it was done with the help of[**JavaScript**](https://www.lambdatest.com/blog/automation-testing-with-selenium-javascript/) and its libraries which as a developer,we would know that it unnecessarily complicated the matters. Animation in CSS bring pre-defined properties that are easier to apply and faster to establish the end-goals with multiple options packed inside it.

Moving the animation in CSS provides better performance than JavaScript as the browser takes control and optimizes the animation for the performance. The properties that come with the CSS animations can be divided into three major categories:

1. **Transformation** – Transforming the dimensions, rescaling the objects, moving them from point A to B, etc.
2. **Transitions** – Performing the transformations smoothly.
3. **Keyframes**– Changing the animation (property, value, etc.) at a given time or state.

**CSS Animations**

To make CSS Animations we need,

1. HTML element to animate

2. CSS rule to animate

3. key frames that defines start and end of the animation

To enhance Animation delay ,speed properties are used

**2.2.2 Detailed Look at the CSS Animation Property**

CSS Animations is a technique to change the appearance and behavior of various elements in web pages. By the use of CSS Animation property, we can control the elements by changing their motions or display. CSS Animation property applies an animation between styles. The following table shows the CSS animation properties .

|  |  |
| --- | --- |
| **Value** | **Description** |
| [*animation-name*](https://www.w3schools.com/cssref/css3_pr_animation-name.php) | Specifies the name of the keyframe you want to bind to the selector |
| [*animation-duration*](https://www.w3schools.com/cssref/css3_pr_animation-duration.php) | Specifies how many seconds or milliseconds an animation takes to complete |
| [*animation-timing-function*](https://www.w3schools.com/cssref/css3_pr_animation-timing-function.php) | Specifies the speed curve of the animation |
| [*animation-delay*](https://www.w3schools.com/cssref/css3_pr_animation-delay.php) | Specifies a delay before the animation will start |
| [*animation-iteration-count*](https://www.w3schools.com/cssref/css3_pr_animation-iteration-count.php) | Specifies how many times an animation should be played |
| [*animation-direction*](https://www.w3schools.com/cssref/css3_pr_animation-direction.php) | Specifies whether or not the animation should play in reverse on alternate cycles |
| [*animation-fill-mode*](https://www.w3schools.com/cssref/css3_pr_animation-fill-mode.php) | Specifies what values are applied by the animation outside the time it is executing |
| [*animation-play-state*](https://www.w3schools.com/cssref/css3_pr_animation-play-state.php) | Specifies whether the animation is running or paused |
| initial | Sets this property to its default value. [Read about initial](https://www.w3schools.com/cssref/css_initial.php) |
| inherit | Inherits this property from its parent element. [Read about inherit](https://www.w3schools.com/cssref/css_inherit.php) |

[**animation-name**](https://www.geeksforgeeks.org/css-animation-name-property/)

It is used to specify the name of the @keyframes describing the animation.

[**animation-duration**](https://www.geeksforgeeks.org/css-animation-duration-property/)

It is used to specify the time duration it takes animation to complete one cycle. we can specify the time in seconds(***s***) or in milliseconds(***ms***).

[**animation-timing-function**](https://www.geeksforgeeks.org/css-animation-timing-function-property/)

It specifies how animations make transitions through keyframes. There are several presets available in CSS which are used as the value for the animation-timing-function like **linear**, **ease,ease-in**,**ease-out**, and **ease-in-out.**

[**animation-delay**](https://www.geeksforgeeks.org/css-animation-delay-property/)

It specifies the delay of the start of an animation. we can specify the time in seconds(***s***) or in milliseconds(***ms***).

[**animation-iteration-count**](https://www.geeksforgeeks.org/css-animation-iteration-count-property/)

This specifies the number of times the animation will be repeated.

[**animation-direction**](https://www.geeksforgeeks.org/css-animation-direction-property/)

It defines the direction of the animation. animation direction can be **normal**, **reverse**, **alternate**, and **alternate-reverse**.

[**animation-fill-mode**](https://www.geeksforgeeks.org/css-animation-fill-mode-property/)

It defines how styles are applied before and after animation. The animation fill mode can be **none**, **forwards**,**backwards**, or **both**.

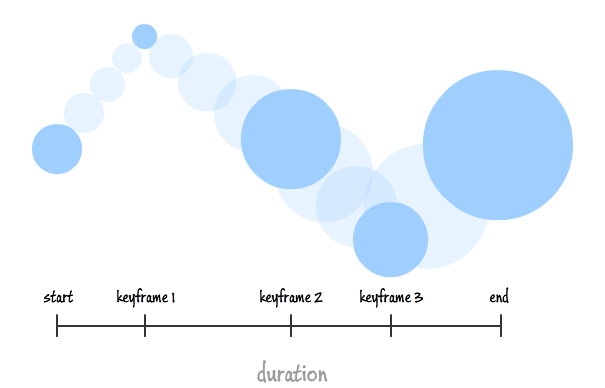
[**animation-play-state**](https://www.geeksforgeeks.org/css-animation-play-state-property/)

This property specifies whether the animation is running or paused. It takes the values ***paused*** or ***running.***

**Note:** Always specify the [***animation-duration***](https://www.w3schools.com/cssref/css3_pr_animation-duration.php) property, otherwise the duration is 0, and will never be played.The **animation-delay** property specifies a delay for the start of an animation.The animation-delay value is defined in seconds (s) or milliseconds (ms).

**2.2.3 Keyframes**

To use CSS animation, we must first specify some keyframes for the animation.Keyframes hold what styles the element will have at certain times.



In the above example, the blue circle isn't simply sliding to the right and getting larger. The individual keyframes adjust the circle's size and vertical position of the ball.

**The @keyframes Rule**

Keyframes are the foundations of CSS Animations . They define the display of the animation at the respective stages of its whole duration.

When we 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, we must bind the animation to an element.Keyframes takes the general format,

***@keyframes animationname***

***{***

***keyframes-selector***

***{css-styles;}***

***}***

Where

Animationname → name of the animation

keyframes-selector→Percentage of the animation duration. It may take values ***0-100%*** ( or)  
**from - to** (same as 0%- 100%)

css-styles→One or more legal CSS style properties

For example,

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

Here, in the example above, binds the "***example"*** animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow".

**Example 2.2 Simple Animation**

<html>

<head>

<style>

/\* The animation code \*/

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

/\* The element to apply the animation to \*/

div {

width: 120px;

height: 50px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

p

{

font-size:20;

}

</style></head>

<body bgcolor=powderblue>

<div><p> MULTIMEDIA</div>

<hr>

<div><p> ANIMATION </div>

</body>

</html>

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

In the example below, style is changed by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

Here, the background-color of the <div> element will be changed in 25%, 50% and 100% of animation is completed.

**Example 2.3 Simple Animation**

<html>

<head>

<style>

/\* The animation code \*/

@keyframes example {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

/\* The element to apply the animation to \*/

div {

width: 125px;

height: 50px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

p

{

font-size:20;

}

</style></head>

<body bgcolor= dodgerblue>

<div><p> MULTIMEDIA</div>

<hr>

<div><p> ANIMATION </div>

</body>

</html>

**Example 2.4 Simple Animation**

<html>

<head>

<style>

/\* The animation code \*/

@keyframes example {

0% {background-color:red; left:0px; top:0px;}

25% {background-color:yellow; left:200px; top:0px;}

50% {background-color:blue; left:200px; top:200px;}

75% {background-color:green; left:0px; top:200px;}

100% {background-color:red; left:0px; top:0px;}

}

/\* The element to apply the animation to \*/

div {

width: 150px;

height: 40px;

position: relative;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

p

{

font-size:20;

}

</style></head>

<body bgcolor=slateblue>

<div><p> MULTIMEDIA</div>

<div><p> ANIMATION </div>

</body>

</html>

**Example 2.5 To demonstrate animation-timing-function**

<html>

<head>

<style>

#jk{

width: 400px;

height: 100px;

position: relative;

animation-name: EASTPT;

animation-duration: 5s;

animation-timing-function: linear;

animation-delay: 1s;

animation-iteration-count: infinite;

animation-direction: alternate;

}

@keyframes EASTPT {

0% {

left: 0px;

top: 0px;

}

25% {

left: 200px;

top: 200px;

}

50% {

left: 200px;

top: 0px;

}

75% {

left: 0px;

top: 200px;

}

100% {

left: 0px;

top: 0px;

}

}

</style>

</head>

<body bgcolor=orange>

<h1 id="jk" style="color:blue;" > Bangalore</h1>

</body>

</html>

**Example 2.6Demonstration of animation-timing-function property**

<!DOCTYPE html>

<html>

<head>

<style>

.eastpt {

font-size: 40px;

text-align: center;

font-weight: bold;

color: #090;

padding-bottom: 5px;

font-family: Times New Roman;

}

.bca {

font-size: 17px;

font-weight: bold;

text-align: center;

color: #008;

font-family: Times New Roman;

}

h2 {

width: 350px;

animation-name: text;

animation-duration: 4s;

animation-iteration-count: infinite;

background-color: rgb(255, 210, 85);

}

#one { animation-timing-function: ease;}

#two { animation-timing-function: linear;}

#three {animation-timing-function: ease-in;}

#four { animation-timing-function: ease-out;}

#five { animation-timing-function: ease-in-out;}

@keyframes text {

from { margin-left: 60%; }

to { margin-left: 0%; }

}

</style>

</head>

<body>

<div class="eastpt">Computer Multimedia and Animation</div>

<div class="bca">CSS Animation</div>

<h2 id="one">This text is ease.</h2>

<h2 id="two">This text is linear.</h2>

<h2 id="three">This text is ease-in.</h2>

<h2 id="four">This text is ease-out.</h2>

<h2 id="five">This text is ease-in-out.</h2>

</body>

</html>

**Example 2.7Demonstration of animation-delay property**

<html>

<head>

<style>

.east {

font-size: 40px;

text-align: center;

font-weight: bold;

color: #090;

padding-bottom: 5px;

font-family: Times New Roman;

}

.bca {

font-size: 17px;

font-weight: bold;

text-align: center;

font-family: Times New Roman;

}

#one {

animation-name: example;

animation-duration: 10s;

}

#two {

animation-name: example;

animation-duration: 10s;

animation-delay: 10s;

}

@keyframes example {

from { background-color: orange; }

to { background-color: white; }

}

</style>

</head>

<body>

<div class="east">CSS Animation</div>

<div class="bca"> IV SEMESTER</div>

<h2 id="one">Text animation without delay.</h2>

<h2 id="two">Text animation with 10 second delay.</h2>

</body>

</html>

**Example 2.8 Demonstration of animation-iteration-count property**

<html>

<head>

<style>

.east {

font-size: 40px;

text-align: center;

font-weight: bold;

color: #090;

padding-bottom: 5px;

font-family: Times New Roman;

}

.bca {

font-size: 17px;

font-weight: bold;

text-align: center;

font-family: Times New Roman;

}

#one {

animation-name: example;

animation-duration: 2s;

animation-iteration-count: 2;

}

#two {

animation-name: example;

animation-duration: 2s;

animation-iteration-count: infinite;

}

@keyframes example {

from { background-color: orange; }

to { background-color: white; }

}

</style>

</head>

<body>

<div class="east">East Point College of Higher Education</div>

<div class="bca">Bachelors in Computer Application</div>

<h2 id="one">This text changes its color two times</h2>

<h2 id="two">This text changes its color infinite times</h2>

</body>

</html>

**Example 2.9Demonstration of Animation-directionproperty**

<html>

<head>

<style>

.jk {

font-size: 40px;

text-align: center;

font-weight: bold;

color: #090;

padding-bottom: 5px;

font-family: Times New Roman;

}

h2 {

width: 100%;

animation-name: text;

animation-duration: 2s;

animation-iteration-count: infinite;

}

#one { animation-direction: normal; }

#two { animation-direction: reverse; }

#three { animation-direction: alternate; }

#four { animation-direction: alternate-reverse; }

@keyframes text {

from { margin-left: 60%; }

to { margin-left: 0%; }

}

</style>

</head>

<body>

<div class="jk">Animation-direction Property</div>

<h2 id="one">This text is normal.</h2>

<h2 id="two">This text is reverse.</h2>

<h2 id="three">This text is alternate.</h2>

<h2 id="four">This text is alternate-reverse.</h2>

</body>

**Example 2.10Demonstration of animation-fill mode property**

<!DOCTYPE html>

<html>

<head>

<style>

.east {

font-size: 40px;

text-align: center;

font-weight: bold;

color: #090;

padding-bottom: 5px;

font-family: Times New Roman;

}

h2 {

width: 400px;

background-color: orange;

animation-name: text;

animation-duration: 3s;

}

#one {

animation-fill-mode: none;

}

#two {

animation-fill-mode: forwards;

}

#three {

animation-fill-mode: backwards;

animation-delay: 2s;

}

#four {

animation-fill-mode: both;

animation-delay: 2s;

}

@keyframes text {

from {

margin-left: 0%;

background-color: #aaaaaa;

}

to {

margin-left: 60%;

background-color: #008000;

}

}

</style>

</head>

<body>

<div class="east">Animation fill -mode</div>

<h2 id="one">none</h2>

<h2 id="two">forwards</h2>

<h2 id="three">backwards</h2>

<h2 id="four">both</h2>

</body>

</html>

**2.2.4 Declaring Multiple Animations**

We can define multiple animation for a single element. Here in this example,

***animation: animScale 2000ms ease-in-out infinite,***

***animOpacity 2000ms ease-in-out infinite;***

**Example 2.11Demonstration of Multiple Animation**

<html>

<head>

<style>

@keyframes animScale{

from{ transform: scale(0.2, 0.2); }

to{ transform: scale(1, 1); }

}

@keyframes animOpacity{

from{ opacity: 0; }

to{ opacity: 1; }

}

#green{

width: 200px;

height: 50px;

background: green;

animation: animScale 2000ms ease-in-out infinite,

animOpacity 2000ms ease-in-out infinite;

}

#saffron{

width: 200px;

height: 50px;

background: #FAC430;

animation: animScale 2000ms ease-in-out infinite,

animOpacity 2000ms ease-in-out infinite;

}

</style>

<body>

<div></div>

<p id="saffron"><h1>I LOVE INDIA</h1><p id="green">

</body>

<html>

**2.2.5 Wrap-up**

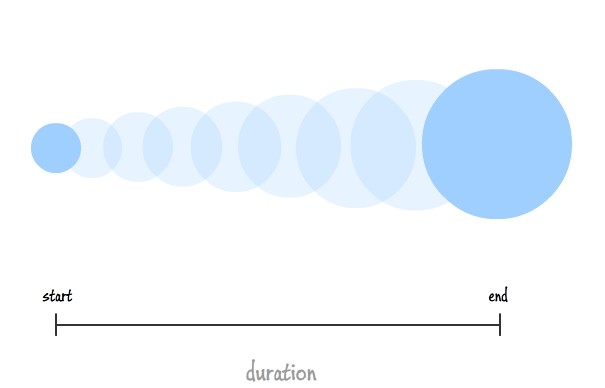
* The word-wrap property allows long words to be able to be broken and wrap onto the next line.
* [This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box. This property defines the breaks in the word to avoid the overflow when a word is too long to fit in the container.](https://www.w3schools.com/cssref/playdemo.php?filename=playcss_word-wrap)

**2.3 CSS Transitions**

* + Transitions in CSS allow us to control the way in which transition takes place between the two states of the element.
  + CSS transitions allows us to change property values smoothly, over a given duration.

For example, when hovering your mouse over a button, we can change the background color of the element with help of CSS selector and [pseudo-class](https://www.geeksforgeeks.org/css-pseudo-classes/).

We can change any other or combination of properties. The transition allows us to determine how the change in color takes place. We can use the transitions to animate the changes and make the changes visually appealing to the user and hence, giving a better user experience and interactivity.



**2.3.1 All About CSS Transitions**

**Adding a Transition**

To create a transition effect, we must specify two things

* the CSS property ,we want to add an effect
* the duration of the effect

 If the duration part is not specified, the transition will have no effect, because the default value is 0.

**2.3.3 Looking at Transitions in Detail**

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

* transition
* transition-delay
* transition-duration
* transition-property
* transition-timing-function

**1. transition-property**

 This property allows us to select the CSS properties which we want to animate during the transition(change).

**Syntax:**

***transition-property: none | all | property | property1, property2, ..., propertyN;***

***where***

**none** 🡪used to specify that no property should be selected.

**all** 🡪used to specify all the properties to be selected, though not all properties are animate-able, only the properties which are animate-able will be influenced.

We can specify a single **property** or a set of comma-separated properties **property1, property2, …, propertyN**.

**2. transition-duration**

This property allows us to determine how long it will take to complete the transition from one CSS property to the other.

**Syntax:**

***transition-duration: time;***

Here, **time** can be in seconds(s) or milliseconds(ms), you should use ‘s’ or ‘ms’ after the number (without quotes).

**3. transition-timing-function**

 This property allows you to determine the speed of change and the manner of change, during the transition. Like, the change should be fast at the beginning and slow at the end, etc.

**Syntax:**

***transition-timing-function: ease|ease-in|ease-out|ease-in-out|linear| step-start|step-end;***

Note, there are other values that this transition-timing-function can take, only the most frequent and simple are mentioned here.

**4. transition-delay**

This property allows you to determine the amount of time to wait before the transition actually starts to take place.

**Syntax:**

***transition-delay: time;***

Here, again, **time** can be in seconds(s) or milliseconds(ms), and you should use ‘s’ or ‘ms’ after the number (without quotes).

**Example 2. 11 Demonstration of Transition effect with delay**

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background: red;

transition: width 3s;

transition-delay: 1s;

}

div:hover {

width: 300px;

}

</style>

</head>

<body>

<h1>The transition-delay Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

<p><b>Note:</b> The transition effect has a 1 second delay before starting.</p>

</body>

</html>

**Example 2. 12 Transition effect with transform**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background: red;

transition: width 2s, height 2s, transform 2s;

}

div:hover {

width: 300px;

height: 300px;

transform: rotate(180deg);

}

</style>

</head>

<body>

<h1>Transition + Transform</h1>

<p>Hover over the div element below:</p>

<div></div>

</body>

</html>

**Example 2. 13 Transition effect Example without delay**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background: red;

transition: width 2s;

}

div:hover {

width: 300px;

}

</style>

</head>

<body>

<h1>The transition Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

</body>

</html>

**Example 2. 14 Transition effect Examples**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background: slateblue;

transition: width 2s, height 4s;

}

div:hover {

width: 300px;

height: 300px;

}

</style>

</head>

<body>

<h1>The transition Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

</body>

</html>

**Specifying the Speed Curve of the Transition**

The transition-timing-function property specifies the speed curve of the transition effect. The transition-timing-function property can have the following values:

* ***ease*** - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
* ***linear*** - specifies a transition effect with the same speed from start to end
* ***ease-in*** - specifies a transition effect with a slow start
* ***ease-out*** - specifies a transition effect with a slow end
* ***ease-in-out*** - specifies a transition effect with a slow start and end
* ***cubic-bezier(n,n,n,n)*** - lets you define your own values in a cubic-bezier function

**Example 2. 15 Transition effect with speed curve**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background: powderblue;

transition: width 2s;

}

#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;}

div:hover {

width: 300px;

}

</style>

</head>

<body>

<h1>The transition-timing-function Property</h1>

<p>Hover over the div elements below, to see the different speed curves:</p>

<div id="div1">linear</div><br>

<div id="div2">ease</div><br>

<div id="div3">ease-in</div><br>

<div id="div4">ease-out</div><br>

<div id="div5">ease-in-out</div><br>

</body>

</html>

**2.3.4 The Longhand Properties**

**Mentioning the properties of element in individual line is referred as Longhand properties.**

p {

animation-duration: 3s;

animation-name: slidein;

animation-iteration-count: infinite;

animation-direction: alternate;

}

is equivalent to

p {

animation: slidein3s infinite alternate;

}

**2.3.5 Longhand Properties vs. Shorthand Properties**

**Mentioning the properties of element in individual line is referred as Longhand properties. We can combine more than one properties into one single line is referred as shorthand properties.**

**Shorthand in Animation Property**

 It is a shorthand way of implying the animation properties for a quicker code. The properties should be in the following order

***animation: [animation-name] [animation-duration] [animation-timing-function] [animation-delay]***

***[animation-iteration-count] [animation-direction] [animation-fill-mode]***

***[animation-play-state];***

**Shorthand in Transition Property**

We can combine all the four transition properties mentioned above, into one single shorthand property, according to the syntax given below. This saves us from writing long codes and prevents from getting messy. The ordering of property is important.

***transition: (property name) | (duration) | (timing function) | (delay);***

This property must be placed with other CSS properties, if any, of the initial state. We should use at least, **property name** and **duration** to get any animate-able effect. Also, the ordering of the values matters. The first value is of the property name, second for the duration and so on, as listed above. So, if only one number is mentioned, it will be taken up as duration, and not as a delay

### *Example I* Binding an animation to a <div> element, using the shorthand property

div {  
  animation: mymove 5s infinite;  
}

### Example IIBinding an animation to a <div> element, using the shorthand property

p {

animation-duration: 3s;

animation-name: slidein;

animation-iteration-count: infinite;

animation-direction: alternate;

}

***is equivalent to***

p {

animation: slidein3s infinite alternate;

}

**2.3.6 Working with Multiple Transitions**

 To have multiple transitions on an element, basically we have two ways.

* One is specifying the **properties** to be transitioned, the **time duration** of the transition, and the **timing function** of the transition separately and let that work.
* Another way to implement this is to add all the details about the transition in a **shorthand** form where we add the property, time duration, and timing function of the properties and **separate them with commas**.

**Transition keywords**

* **transition:** This keyword can be used with a CSS property in inline, internal, or external CSS. This needs the **property (transition-property)** which will be transitioned, the **time duration (transition-duration)** of the transition, **timing(transition-timing-function)** function of the transition. We can give those values individually to the property or we can use the shorthand technique to add all of them at the same time.
* **transition-property:**This is used to specify the properties to be transitioned.
* **transition-duration:** This is used to specify the time duration for which the properties will be transitioned.
* **transition-timing-function:**This is used to specify the time duration for which the properties will be transitioned.

**Example 2. 15 Transition effect with speed curve**

<html>

<head>

<style>

h3 { color: brown !important; }

.container { display: flex !important; }

div {

font-family: "Lucida Sans", "Lucida Sans Regular";

font-size: 1rem;

margin: 2rem;

justify-content: center;

display: flex;

border: 10px solid green;

width: 18rem;

height: 7rem;

padding-bottom: 20px;

padding-top: 20px;

transition: color 1s ease-out, padding-top 1s ease-out,

padding-bottom 1s ease-out, font-size 2s ease-out;

}

div:hover {

color: rebeccapurple;

border: 10px solid brown;

padding-top: 100px;

padding-bottom: 30px;

font-size: 1.8rem;

}

</style>

</head>

<body>

<h1 style="color: green; margin: 2rem;">

BCA IV SEM PEARLS

</h1>

<h3 style="margin: 2.2rem; margin-top: -2rem">

multiple CSS transitions on an element

</h3>

<div>Computer Multimedia and Animation</div>

</body>

</html>