

2. Animation

What is an Animation?

* An Animation is the process of creating a sequence of images or moving pictures that give the illusion of motion and change.

* It involves the use of various techniques and software to bring characters, objects, and environments to life, often for the purpose of entertainment, education and/or advertising.

* Animators use their skills to create a variety of different types of animation, including traditional hand-drawn animation, computer-generated animation, stop-motion animation and more.

CSS Animations:-

* CSS Animations are a way to add visual effects to web pages.

* With CSS animations, you can animate the position, sizes and color of HTML elements.

* With CSS animations, you can specify

1. Duration of an animation.

2. Speed of an animation.

3. Number of iterations of an animation.

4. Direction of an animation.

Creating a Simple Animation:-

Steps to Create CSS Animation:

1. Define keyframes with starting state and ending state of the element.

2. Apply the animation properties to an element.

1. Plan your Animation:-

Decide what you want to animate and how you want it to move or change.

2. Create the Start State:-

Draw or create the initial position or appearance of your characters, objects, or environments.

3. Create the End State:-

Draw or create the final position or appearance of your character, objects, or environments after all the movement and changes have taken place.

4. Break down the Animation into Frames:-

Divide the animation into smaller parts by creating frames that show the gradual transition from the start state to the end state.

5. Draw or Create each Frame:-

Draw or create each frame of the animation making sure that each frame shows a small change from the previous frame.

6. Play Back the Animation

Once all the frames are completed, play them back in sequence to see the animation in action.

7. Refine and adjust:-

Review your animation and make any necessary adjustments to improve its flow and timing.

8. Export and Share:-

Export your animation as a video file or GIF and share it with others.

CSS Animation Properties:-

Property	Description
1. @keyframes	Defines the animation behavior for an element.
2. animation-name	Specifies the name of the keyframe animation to be applied.
3. animation-duration	Specifies the duration of an animation to complete one cycle.
4. animation-timing-function	Specifies the speed curve of the animation.
5. animation-delay	Specifies a delay before the animation starts.

6. animation-iteration-count

Specifies the number of times the animation should be repeated.

7. animation-direction

Specifies whether the animation should play forwards, backwards, or alternate back & forth.

8. animation-fill-mode

Specifies what styles are applied to the element before and after the animation.

9. animation-play-state

Specifies whether the animation is running or paused.

10. animation

A shorthand property used for setting all the animation properties.

Animation Property:-

①. Keyframes:-

*This is used to define the animation behavior for an element.

*This is used to combination with other

animation-related CSS properties such as animation-name, animation-duration, animation-direction etc.

CSS Syntax:-

```
@keyframes animation-name {  
    from /* element at the starting state of the  
           animation */;  
    to /* element at the ending state of the  
           animation */;}
```

where, animation-name is a user-defined name for the animation

from keyword is used to define the starting state of the element.

to keyword is used to define the ending state of the element.

* Keyframes can also be defined as a series of percentages (0%, 100%).

CSS Syntax:-

```
@keyframes animation-name {  
    0% /* element at the starting state of the  
           animation */;
```

100% /* element at the ending state of the animation */;

Example 1:-

```
@keyframes changecolor {  
    from {background: red;}  
    to {background: green;}  
}
```

Example 2:-

```
@keyframes changecolor {  
    0% {background: red;}  
    100% {background: green;}  
}
```

Example 3:-

```
@keyframes movebox {  
    from {left: 0px; top: 0px;}  
    to {left: 600px; top: 0px;}  
}
```

animation-name:-

* This specifies the name of the keyframe animation to be applied to the element.

* This is used in combination with other

animation - related CSS properties such as a `animation-duration`, `animation-delay`, and `animation-direction`.

CSS Syntax:-

```
animation-name: name;
```

where, `name` is the name of the animation defined using the `@keyframes` rule.

☞ `animation-duration`:-

* This specifies the duration of an animation to complete one cycle.

CSS Syntax:-

```
animation-duration: time;
```

where, `time` is the duration of the animation. Time can be specified in seconds or milliseconds.

Program to change background color of an element from red to green using animation name and `animation-duration` properties.

```
<html>
```

```
<head>
```

```
<style>
```

```
 @keyframes changecolor {  
   from {background-color: red;}  
   to {background-color: green;}  
 }
```

```
div {
```

```
 width: 100px;
```

```
 height: 100px;
```

```
 background: red;
```

```
 animation-name: changecolor;  
 animation-duration: 5s;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
 <div></div>
```

```
</body>
```

```
</html>
```

iv). animation-timing-function:-
* This specifies the speed of animation.

CSS Syntax:-

```
animation-timing-function:linear|ease|ease-in  
    ease-out|ease-in-out|;
```

animation-timing-function table:-

Value	Description
linear	The animation has the same speed from start to end.
ease	Default value. The animation has a slow start, then fast, before it ends slowly.
ease-in	The animation has a slow start.
ease-out	The animation has a slow end.
ease-in-out	The animation has both a slow start and a slow end.

v). animation-delay:-
* This specifies a delay before the animation starts.

CSS Syntax:-

```
animation-delay:time;
```

while, time is the time delay before the animation starts.
time can be specified in seconds(s) or milliseconds(ms).

Program to move a box from left to right using
animation-timing-function and animation-delay
properties.

```
<html>  
  <head>  
    <style>  
      @keyframes movebox {  
        0% { left: 0px; top: 0px; }  
        100% { left: 600px; top: 0px; }  
      }  
      div {  
        width: 100px;  
        height: 100px;  
        background: red;  
        position: relative;  
        animation-name: movebox;  
        animation-duration: 10s;  
      }  
    </style>  
  </head>  
  <body>  
    <div></div>  
  </body>  
</html>
```

```
        animation-timing-function: linear;  
        animation-delay: 5s; }  
</style>  
</head>  
<body>  
    <div></div>  
</body>  
</html>
```

v). animation-iteration-count

*This specifies the number of times the animation should be repeated.

*CSS Syntax:-

```
animation-iteration-count: number | infinite;
```

Value	Description
number	A number that defines how many times an animation should be played. Default value is 1.

Value	Description
infinite	Specifies that the animation should be played infinite times (forever).

vii). animation-direction:-

*This specifies the direction of an animation.
CSS Syntax:-

```
animation-direction: normal | reverse | alternate  
                     alternate-reverse;
```

Value	Description
normal	Default value. The animation is played as normal (forwards).

Value	Description
reverse	The animation is played in reverse direction (backwards).

Value	Description
alternate	The animation is played forwards first, then backwards.

Value	Description
alternate-reverse	The animation is played backwards first, then forwards.

Program to rotate a box infinitely in anti-clockwise direction using animation-iteration-count and animation-direction properties.

```
<html>
<head>
<style>
@keyframes rotatebox{
    0% {top:0px; left:0px;}
    25% {top:0px; left:100px;}
    50% {top:100px; left:100px;}
    75% {top:100px; left:0px;}
    100% {top:0px; left:0px;}
}
div {
    width:100px;
    height:100px;
    background:red;
    position:relative;
    animation-name:rotatebox;
    animation-duration:8s;
    animation-iteration-count:infinite;
    animation-direction:reverse;
}
</style>
```

```
</head>
<body>
<div></div>
</body>
</html>
```

viii. animation-fill-mode:

* This specifies what styles are applied to the element before and after the animation.

CSS Syntax:

animation-fill-mode:none|forward|backwards

Value	Description
none	Default value. Animation will not apply any styles to the element before or after it is executing.
forward	The element will retain the style values that is set by the last keyframe.
backward	The element will get the style values that is set by the first keyframe, and this during the animation-delay period.

ii). animation-play-state:-

* This specifies whether the animation is running or paused.

CSS Syntax:-

```
animation-play-state:running|paused;
```

Value	Description
paused	Specifies that the animation is paused.

running	Default value. Specifies that the animation is running.
---------	---

Program to move a box from left to right and right to left using animation-fill-mode and animation-play-state properties.

```
<html> position: absolute;
```

```
<head>
```

```
<style> margin-left: 100px;
```

```
@keyframes movebar{from {top:0px;left:0px;}
```

```
to {top:0px;left:100px;}}
```

```
body{position: relative; height: 100px; width: 100px;}
```

```
div{position: absolute; top: 0; left: 0;}
```

```
width: 100px; height: 100px;}
```

```
background-color: red; border: 1px solid black;}
```

background-image:

position: relative;

animation-name: movebar;

animation-duration: 5s;

animation-iteration-count: 3;

animation-direction: alternate;

animation-fill-mode: forwards;

animation-play-state: running; }

```
<style>
```

```
</head>
```

```
<body>
```

```
<div></div></body>
```

```
</html>
```

iii) Animation:-

* This is a shorthand property, used for setting all the properties, except the properties except the animation-play-state and the animation-fill-mode property.

* The animation is a shorthand property for:-

i). animation-name

ii). animation-duration

iii). animation-timing-function

iv). animation-delay

① animation-iteration-count
② animation-direction

CSS Syntax -

animation: [name] [duration] [timing-function]
[delay] [iteration-count] [direction];

Two types of animation in CSS properties -

1. Longhand Property
2. Shorthand Property

1. Longhand Property -

Longhand Property, individual CSS properties can be set separately.

2. Shorthand Property -

Shorthand Property, multiple CSS properties can be set with a single line of code

Longhand

animation-name: color; animation-duration: 5s;
animation-direction: linear; animation-timing-function: linear;
animation-delay: 1s;
animation-iteration-count: infinite;
animation-direction: normal;

Shorthand

animation: color 5s linear infinite normal;

Program to change background color of a box
using shorthand property

(HTML)

<head>

<style>

```
@keyframes changeColor {  
    from {background: red;}  
    to {background: green;}  
}
```

div {

```
width: 100px;  
height: 100px;  
background: red;  
position: relative;  
animation: changeColor 5s linear infinite normal;  
}
```

```
</style>
<head>
<body>
  <div></div>
</body>
</html>
```

Declaring Multiple Animation:

- * Multiple Animations refer to the ability to apply more than one animation to an element.
- * This can be achieved by specifying multiple animation properties in the CSS code, each with its own set of animation properties.
- * The animations can be defined using different @keyframes rules or can be variations of the same animation with different durations, timing functions, or other properties.
- * Multiple animations can add complexity and visual interest to web pages and are commonly used in interactive designs and games.

* It is important to use them judiciously as they can also slow down page loading times and affect performance.

* It can apply multiple animations to a single element using the animation property.

Longhand	Shorthand
animation-name: changecolor	animation: changecolor 5s
animation-duration: 5s	linear 5s infinite normal,
animation-timing-function: linear	movecolor 5s linear 1s infinite
animation-delay: 2s	normal.
animation-iteration-count: infinite	
animation-direction: normal	

animation-name: movebox
animation-duration: 5s
animation-timing-function: linear
animation-delay: 1s
animation-iteration-count: infinite
animation-direction: normal

Program to change background color of a box and move the box from left to right using shorthand property.

```
<html>
  <head>
    <style>
      @keyframes changeColor {
        from {background:red;}
        to {background:green;}
      }

      @keyframes moveBox {
        from {left:0px; top:0px;}
        to {left:100px; top:0px;}
      }

      div {
        width:100px;
        height:100px;
        background:red;
        position:relative;
        animation:changeColor 5s linear 1s
          infinite normal,
          moveBox 5s linear 1s
          infinite normal;
      }
    </style>
  </head>
```

```
<body>
  <div></div>
</body>
</html>
```

CSS Transitions:-

- * CSS transitions are a way to make web page elements change smoothly from style to another.
 - * A transition effect could typically occur when a user hovers over an element.
 - * It's used to add animation effects to HTML elements, such as changing the color, size, position, or opacity of an element.
 - * CSS Transition works by defining the starting and ending states of an element.
 - * Then specifying the duration and timing function of the transition.
 - * CSS transition can be applied to a wide range of HTML elements, including text, images, buttons, and links.
- With CSS transitions, you can specify-
1. Duration of transition.
 2. Speed of transition.
 3. CSS property for transition.

Adding a Transition:-

Steps to create CSS Transitions:

1. Define starting state of the element.
2. Define ending state of the element
- 3 Select the properties of the element.

Looking at transition details:-

* CSS transition allows to smoothly animate changes in CSS properties over a period of time.

* It can use transitions to create effects like fading in and out, sliding, scaling, rotating, and more.

Ex:-

```
div {  
background-color: red;  
transition: background-color 1s;  
}
```

```
div : hover {  
background-color: blue;  
transition: background-color 2s;  
}
```

Transition Properties/Property:-

* Transition Property is used to specify the CSS properties that should be transitioned and the duration of the transition.

Property	Description
transition	A shorthand property for setting the four transition properties into a single property.

transition-delay	Specifies a delay(in seconds) for the transition effect.
------------------	--

transition-duration	Specifies how many seconds or milliseconds a transition effect takes to complete.
---------------------	---

transition-property	Specifies the name of the CSS property the transition effect is for.
---------------------	--

transition-timing-function	Specifies the speed curve of the transition effect.
----------------------------	---

i) CSS transition - property:-

* This specifies which property of an element should have the transition effect.

Ex:-

* A Transition effect could typically occur when a user hover over an element.

CSS Syntax:-

```
transition-property: property;
```

where, property, is the property to which the transition effect should be applied.

ii) CSS transition - duration:-

* This specifies the duration of transition effect.

CSS Syntax:-

```
transition-duration: time;
```

where, time is the duration of the transition effect.

time can be specified in seconds or

milliseconds.
1 second is equal to 1000 milliseconds.

Program to change background color of a box from red to yellow when it is being hovered over.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        width: 100px;
        height: 100px;
        background: red;
      }
      div:hover {
        background: yellow;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

iii) transition-timing-function:-

*This specifies the speed of a transition effect.

CSS Syntax:-

```
transition-timing-function: linear | ease | ease-in |  
ease-out | ease-in-out;
```

Value	Description
ease	Default value. Specifies a transition effect with a slow start, then fast, then end slowly.
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 start and end.

iv) transition-delay:-

*This specifies a delay before the transition starts.

CSS Syntax:-

```
transition-delay time
```

where, time is the time delay before the transition starts.
time can be specified in seconds(s) or milliseconds(mil)

Program to change width of a box from 100 pixels
to 200 pixels when it is being hovered over,

```
<html>  
  <head>  
    <style>  
      div {  
        width: 100px;  
        height: 100px;  
        background: blue;  
      }  
      div:hover {  
        width: 200px;  
      }  
    </style>  
  </head>  
</html>
```

```
transition-duration: 5s;  
transition-timing-function: linear;  
transition-delay: 1s;  
}  
</style>  
<head>  
<body>  
<div></div>  
</body>  
</html>
```

1. Longhand Properties:-

In Longhand properties, individual CSS properties can be set separately.

2. Shorthand Properties:-

In Shorthand properties, multiple CSS properties can be set with a single line of code.

3) CSS transition property:-

*The transition is a shorthand property for-

1. transition-property.
2. transition-duration.
3. transition-timing-function
4. transition-delay.

CSS Syntax:-

transition: [property] [duration] [timing-function] [delay];

Eg:-

CSS Longhand	CSS Shorthand
transition-property: width;	transition: width 5s linear;
transition-duration: 5s;	
transition-timing-function:	linear
transition-delay: 1s;	

Program to change width of a box from 100 pixels to 200 pixels when it is being hovered using shorthand property.

```
<html>  
<head>  
<style>  
div{  
width: 100px;  
height: 100px;  
background: blue;  
}  
div:hover{  
width: 200px;  
}
```

```
div {transition: width 5s linear; }  
<style>  
</head>  
<body>  
  <div></div>  
  </body>  
</html>
```

Country

卷之三

1988-1989

2008: Hito