Unit-2

CSS animation property

CSS Animation: CSS Animations is a technique to change the appearance and behavior of various elements in web pages. It is used to control the elements by changing their motions or display. It has two parts, one contains the CSS properties which describe the animation of the elements and the other contains certain keyframes which indicate the animation properties of the element and the specific time intervals at which those have to occur.

What is a Keyframe?

Keyframes are the foundations with the help of which CSS Animations work. They define the display of the animation at the respective stages of its whole duration. For example: In the first example code, the paragraph changes its color with time. At 0% completion, it is red, at 50% completion it is of orange color and at full completion i.e. at 100%, it is brown.

Syntax:

/*property-name*/: /*value*/;

Animation Properties:

There are certain animation properties given below:

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

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation-name: mymove;}
@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
 }
```

```
</style>
</head>
<body>
<h1>The animation-name Property</h1>
<p>Specify a name for the @keyframes animation.</p>
<div></div>
</body>
</html>
```

- animation-duration: It is used to specify the time duration it takes animation to complete one cycle.

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove infinite;
  animation-duration: 3s;
}
@keyframes mymove {
  from {top: 0px;}
  to {top: 200px;}
}
</style>
</head>
<body>
<h1>The animation-duration Property</h1>
<p>Specify that the animation should complete one cycle in 3 seconds:</p>
<div></div>
</body>
</html>
```

- animation-timing-function: 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.

- **ease** - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- **linear** - Specifies an animation with the same speed from start to end
- **ease-in** - Specifies an animation with a slow start
- **ease-out** - Specifies an animation with a slow end

- **ease-in-out** - Specifies an animation with a slow start and end

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 5s infinite;
  animation-timing-function: linear;
}
@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
}
</style>
</head>
<body>
<h1>The animation-timing-function Property</h1>
<p>Play an animation with the same speed from beginning to
end:</p>
<div></div>
</body>
</html>
```

- animation-delay: It specifies the delay of the start of an animation.

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 5s infinite;
  animation-delay: 2s;
}
@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
}
```

```
</style>
</head>
<body>
<h1>The animation-delay Property</h1>
<p>Start the animation after 2 seconds:</p>
<div></div>
</body>
</html>
```

- animation-iteration-count: This specifies the number of times the animation will be repeated.

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 3s;
  animation-iteration-count: 2;
}
@keyframes mymove {
  from {top: 0px;}
  to {top: 200px;}
}
</style>
</head>
<body>
<h1>The animation-iteration-count Property</h1>
<p>Play the animation two times:</p>
<div></div>
</body>
</html>
```

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

- **normal** - The animation is played as normal (forwards). This is default
- **reverse** - The animation is played in reverse direction (backwards)
- **alternate** - The animation is played forwards first, then backwards
- **alternate-reverse** - The animation is played backwards first, then forwards

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: myfirst 5s 2;
  animation-direction: alternate;
}
@keyframes myfirst {
  0%   {background: red; left: 0px; top: 0px;}
  25%  {background: yellow; left: 200px; top: 0px;}
  50%  {background: blue; left: 200px; top: 200px;}
  75%  {background: green; left: 0px; top: 200px;}
  100% {background: red; left: 0px; top: 0px;}
}
</style>
</head>
<body>
<h1>animation-direction: alternate</h1>
<p>Play the animation forwards first, then backwards:</p>
<div></div>
</body>
</html>
```

- animation-play-state: This property specifies whether the animation is running or paused.

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 5s;
  animation-play-state: paused;
}
@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
}
```

```
</style>
</head>
<body>
<h1>The animation-play-state Property</h1>
<p>Pause an animation:</p>
<div></div>
</body>
</html>
```

CSS transition-property

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

How to Use CSS Transitions?

To create a transition effect, you must specify two things:

- ✓ the CSS property you want to add an effect to
- ✓ the duration of the effect

- **transition-property:** It specifies the CSS properties to which a transition effect should be applied.
- **transition-duration:** It specifies the length of time a transition animation should take to complete.
- **transition-timing-function:** It specifies the speed of transition.
- **transition-delay:** It specifies the transition delay or time when the transition starts.

CSS Multiple Transition Effect

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition-property: width;
  transition-duration: 2s;
  transition-timing-function: linear;
  transition-delay: 1s;
}
```

```
div:hover {
  width: 300px;
}
</style>
</head>
<body>
<h1>The transition Properties Specified One by One</h1>
<p>Hover over the div element below, to see the transition effect:</p>
<div></div>
</body>
</html>
```

Shorthand Properties v/s Longhand Properties

- **Background:** The CSS Background property is used to set the background on a web page. The background can be applied to any element like the body, h1, p, div, etc. There are many properties available with a background such as color, image, position, etc. Some of them are used in the below code.
  - **Longhand way:**

```
background-color:#000;
background-image: url(images/bg.png);
background-repeat: no-repeat;
background-position:left top;
```

  - **Shorthand way:**

```
background:#000 url(images/bg.png) no-repeat left top;
```

- **Font:** The CSS font property is used to apply different fonts to the text on the webpage. The various attributes that can be set using the font are font-family, font-size, font-weight, etc. Some of them are used in the below code.
  - **Longhand way:**

```
htmlfont-style:italic;
font-weight:bold;
font-size:18px;
line-height:150%;
font-family:Arial,sans-serif;
```

  - **Shorthand way:**

```
font: italic bold 18px/150% Arial, sans-serif;
```

- **Border:** The CSS border property is used to apply a border to different elements of a web page. There are many attributes of the border like width, style, color, etc.
  - **Longhand way:**

```
border-width: 1px;
border-style: solid;
border-color: #000;
```

  - **Shorthand way:**

```
border: 1px solid #000;
```

- **Margin:** The CSS margin properties are used to create space around elements, outside of any defined borders. We can define margin for all 4 sides i.e. top, bottom, left and right.
  - **Longhand way:**

```
margin-top: 10px;
margin-right: 5px;
margin-bottom: 10px;
margin-left :5px;
```

  - **Shorthand way:**

```
margin : 10px 5px 10px 5px;
```

- **Padding:** The CSS padding properties are used to generate space around an element's content, inside of any defined borders. Padding can also be applied as top, bottom, left and right padding.
  - **Longhand way:**

```
padding-top: 10px;
padding-right: 5px;
padding-bottom: 10px;
padding-left :5px;
```

  - **Shorthand way:**

```
padding : 10px 5px 10px 5px;
```

HTML5 – SVG

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X,Y coordinate system etc.

SVG became a W3C Recommendation 14. January 2003 and you can check latest version of SVG specification at SVG Specification.

Advantages of SVG: Advantages of using SVG over other image formats (like JPEG and GIF) are:

- SVG images can be created and edited with any text editor.
- SVG images can be searched, indexed, scripted, and compressed.
- SVG images are scalable.
- SVG images can be printed with high quality at any resolution.

Differences between HTML SVG and HTML Canvas:

- SVG is a language for describing 2D graphics in XML whereas Canvas draws 2D graphics, on the fly with JavaScript.
- If attributes of an SVG object are changed, the browser can automatically re-render the shape whereas Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser.
- SVG is resolution independent whereas CANVAS is resolution-dependent.
- SVG supports event handlers whereas CANVAS doesn't have support for event handlers.

SVG Circle

```
<html>
<body>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
  stroke="green" stroke-width="4" fill="yellow" />
Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

- The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are omitted, the circle's center is set to (0,0)

- The r attribute defines the radius of the circle

SVG Rectangle

```
<html>
<body>
<svg width="400" height="100">
  <rect width="400" height="100"
  style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The width and height attributes of the <rect> element define the height and the width of the rectangle
- The style attribute is used to define CSS properties for the rectangle
- The CSS fill property defines the fill color of the rectangle
- The CSS stroke-width property defines the width of the border of the rectangle
- The CSS stroke property defines the color of the border of the rectangle

SVG Rounded Rectangle

```
<!DOCTYPE html>
<html>
<body>
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

SVG Ellipse - <ellipse>

The <ellipse> element is used to create an ellipse.

An ellipse is closely related to a circle. The difference is that an ellipse has an x and a y radius that differs from each other, while a circle has equal x and y radius:

```
<!DOCTYPE html>
<html>
<body>
<svg height="140" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50"
style="fill:yellow;stroke:purple;stroke-width:2" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The cx attribute defines the x coordinate of the center of the ellipse
- The cy attribute defines the y coordinate of the center of the ellipse
- The rx attribute defines the horizontal radius
- The ry attribute defines the vertical radius

SVG Line - <line>

The <line> element is used to create a line:

```
<!DOCTYPE html>
<html>
<body>
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-
width:2" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The x1 attribute defines the start of the line on the x-axis
- The y1 attribute defines the start of the line on the y-axis
- The x2 attribute defines the end of the line on the x-axis
- The y2 attribute defines the end of the line on the y-axis

SVG Polygon - <polygon>

The <polygon> element is used to create a graphic that contains at least three sides.

Polygons are made of straight lines, and the shape is "closed" (all the lines connect up).

```
<!DOCTYPE html>
<html>
<body>
<svg height="210" width="500">
  <polygon points="200,10 250,190 160,210"
style="fill:lime;stroke:purple;stroke-width:1" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The points attribute defines the x and y coordinates for each corner of the polygon

SVG Star

```
<!DOCTYPE html>
<html>
<body>
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
Sorry, your browser does not support inline SVG.
</svg>
 </body>
</html>
```

SVG Polyline - <polyline>

The <polyline> element is used to create any shape that consists of only straight lines (that is connected at several points):

```
<!DOCTYPE html>
<html>
<body>
<svg height="200" width="500">
  <polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
style="fill:none;stroke:black;stroke-width:3" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The points attribute defines the list of points (pairs of x and y coordinates) required to draw the polyline

SVG Gradients

A gradient is a smooth transition from one color to another. In addition, several color transitions can be applied to the same element.

There are two main types of gradients in SVG:

- Linear
- Radial

SVG Linear Gradient - <linearGradient>

The <linearGradient> element is used to define a linear gradient.

The <linearGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

Linear gradients can be defined as horizontal, vertical or angular gradients:

- Horizontal gradients are created when y1 and y2 are equal and x1 and x2 differ
- Vertical gradients are created when x1 and x2 are equal and y1 and y2 differ
- Angular gradients are created when x1 and x2 differ and y1 and y2 differ

```
<!DOCTYPE html>
<html>
<body>
<svg height="150" width="400">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The id attribute of the <linearGradient> tag defines a unique name for the gradient
- The x1, x2, y1,y2 attributes of the <linearGradient> tag define the start and end position of the gradient
- The color range for a gradient can be composed of two or more colors. Each color is specified with a <stop> tag. The offset attribute is used to define where the gradient color begin and end
- The fill attribute links the ellipse element to the gradient

SVG Radial Gradient - <radialGradient>

The <radialGradient> element is used to define a radial gradient.

The <radialGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

```
<!DOCTYPE html>
<html>
<body>
<svg height="150" width="500">
  <defs>
    <radialGradient id="grad1" cx="50%" cy="50%" r="50%" fx="50%"
fy="50%">
      <stop offset="0%" style="stop-color:rgb(255,255,255);stop-opacity:0" />
      <stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />
    </radialGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

- The id attribute of the <radialGradient> tag defines a unique name for the gradient
- The cx, cy and r attributes define the outermost circle and the fx and fy define the innermost circle
- The color range for a gradient can be composed of two or more colors. Each color is specified with a <stop> tag. The offset attribute is used to define where the gradient color begin and end
- The fill attribute links the ellipse element to the gradient