**CSS Animation**

When your particular Element or component shift from any particular styling to another styling this process is called animation. In other word we can say that animation is series of complex movements

**The @keyframes Rule**

Keyframes is used to specify the source styling or destination styling where we shift

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.

These are the properties of animation:

* **@keyframes.**
* **animation-name.**
* **animation-duration.**
* **animation-delay.**
* **animation-iteration-count**
* **animation-direction**
* **animation-timing-function**
* **animation-fill-mode**
* **animation**

.Object{

        width: 200px;

        height:100px;

        border: 2px solid black;

        border-radius: 50%;

        background-color: blue;

        margin: 20px;

        position: absolute;

        animation-name: Right\_Move; // Define Animation Name.

        animation-duration: 5s; // Define how many time to take for moving object one place

to another place.

        animation-delay: 2s; // Define how much time to wait for second move of object.

        animation-timing-function: cubic-bezier(0.165, 0.84, 0.44, 1);// Define Different

way to move object.

        animation-iteration-count: 5; // Define how many time to move object.

    }

    @keyframes Right\_Move{

        from{

         top: 0px;

         left: 0px;

        }

        to{

            top: 0px;

            left: 1100px;

        }

    }

**@Keyframes properties:-** Keyframes properties is used to specify the source styling or destination styling where we shift.

**animation-name properties:-** animation-name properties is used to specify the animation name.

**animation-duration properties:-**  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).

**animation-iteration-count properties:-** The animation-iteration-count property specifies the number of times an animation should run

**animation-Delay properties:-** The animation-delay property specifies a delay for the start of an animation

**animation-timing-function properties:-** The animation-timing-function property specifies the speed curve of the animation.

We can use for specify the animation speed with the help of these value

* 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 but end fast.
* ease-out - Specifies an animation with a slow end but start fast
* ease-in-out - Specifies an animation with a slow start and end
* cubic-bezier(n,n,n,n) - Lets you define your own values in a cubic-bezier function

**animation-direction properties:-** The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.

* 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

**animation-fill-mode properties:-** The animation-fill-mode property specifies a style for the target element when the animation is not playing (before it starts, after it ends, or both)

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

* forwards – when we use forwards value the animation will start from the starting point and go toward end point and then stop.
* backwards - when we use backwards value the animation will start from the starting point and go toward end point and then come from end point and stop.
* both - The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions
* none - Default value. Animation will not apply any styles to the element before or after it is executing

**The same animation effect as above can be achieved by using the shorthand animation property:**

/\* @keyframes duration | easing-function | delay |

iteration-count | direction | fill-mode | play-state | name \*/

animation: 3s ease-in 1s 2 reverse both paused slidein;

**CSS transition**

* When we switch one state to another state this entire process is called transitions .
* Transition is the single movement process and animation is series of complex movements so we can say that transition is sub set of animation.

In this chapter you will learn about the following properties

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

**CSS 2D transform**

CSS transforms allow you to move, rotate, scale, and skew elements. When Mouse over the element below to see a 2D transformation.

With the CSS transform property you can use the following 2D transformation methods:

* translate()
* rotate()
* scaleX()
* scaleY()
* scale()
* skewX()
* skewY()
* skew()
* matrix()