

Lesson:

Animation



Topics Covered

- What is Animation?
- Style Declaration
- Animation Sequence - Keyframes
- Animation Examples
- Animation vs Transition

What is CSS animation?

With CSS animations, it becomes feasible to animate transitions between different CSS style configurations.

These animations typically comprise two parts:

- a **style declaration** that outlines the CSS animation,
- a collection of **keyframes** that establish the initial and final states of the animation's style, along with any potential intermediary points.

Style Declaration

To define animation, the first part is style declaration which specifies the following properties,

1. animation-name: specifies the name of the animation.

```
animation-name: animation1  
animation-name: animation2  
animation-name: animation3
```

2. animation- duration: determines how long the animation will run, and is typically expressed in seconds or milliseconds.

```
animation-duration: 1s  
animation-duration: 1000ms
```

3. animation-timing-function: controls the speed of the animation, and can be set to linear, ease, ease-in, ease-out, or ease-in-out, among others.

```
animation-timing-function: ease-in  
animation-timing-function: ease-out  
animation-timing-function: ease-in-out
```

4. animation-delay: sets a delay before the animation starts, and can be used to synchronize multiple animations or to create a more gradual effect.

```
animation-delay: 1s  
animation-delay: 1000ms
```

5. animation-iteration-count: specifies how many times the animation should repeat, and can be set to a specific number or to infinite.

```
animation-iteration-count: 2
animation-iteration-count: 5
```

6. animation-direction: determines the direction of the animation, and can be set to normal, reverse, alternate, or alternate-reverse.

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

7. animation-fill-mode: controls how the animation behaves before and after it runs and can be set to forwards, backward, both, or none.

```
animation-fill-mode: forwards;
animation-fill-mode: backwards;
animation-fill-mode: both;
```

8. animation-play-state: determines whether the animation is running or paused, and can be set to running or paused.

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

Animation Shorthand Syntax

We can write all animation properties in one line using animation shorthand syntax.

```
animation: name duration timing-function delay
iteration-count direction fill-mode;
```

Animation Sequence

Once you have set up the basic animation properties of your animation, the next step is to specify its visual appearance. To achieve this, you must create one or multiple keyframes using the @keyframes rule. Each keyframe outlines the desired rendering of the animated element at a particular moment within the sequence of the animation.

Syntax

```
@keyframes animation_name {
  from {
    /* initial styles written here */
  }

  75% {
    /* intermediate styles written here */
  }

  to {
    /* final styles written here */
  }
}
```

Animation Examples

Let's try to create a few simple animations through what we have learned in previous sections.

Example 1:-

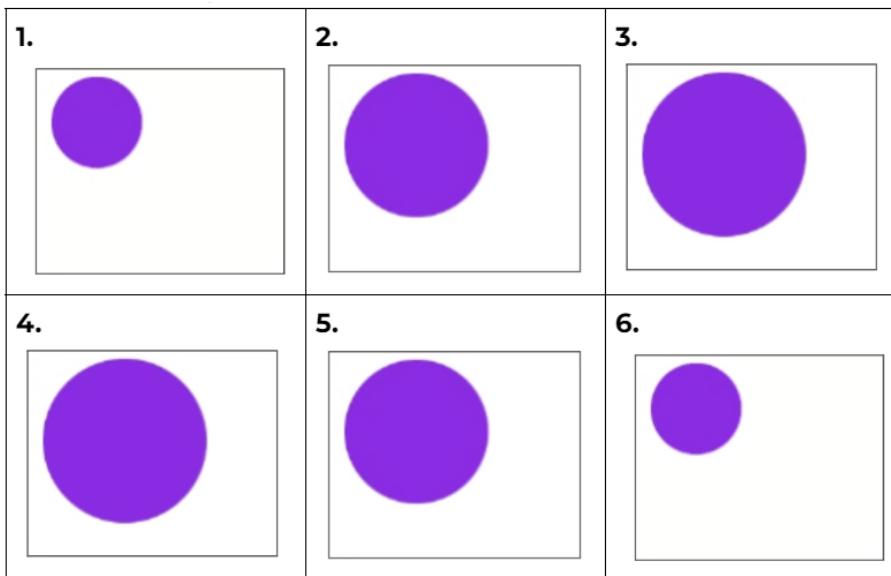
```
<!DOCTYPE html>
<html lang="en-US">
<head>
  <title>Animation</title>
</head>
<style>
  .box {
    position: relative;
    background-color: blueviolet;
    border-radius: 50px;
    animation: move 1s ease-in-out 0s 100 alternate;
  }

  @keyframes move {
    from {
      width: 100px;
      height: 100px;
    }

    to {
      width: 50px;
      height: 50px;
    }
  }
</style>
<body>
  <div class="box"></div>
</body>
</html>
```

Browser Output:-

Animation Sequence



After 6, It will repeat again from 1.

Example 2:-

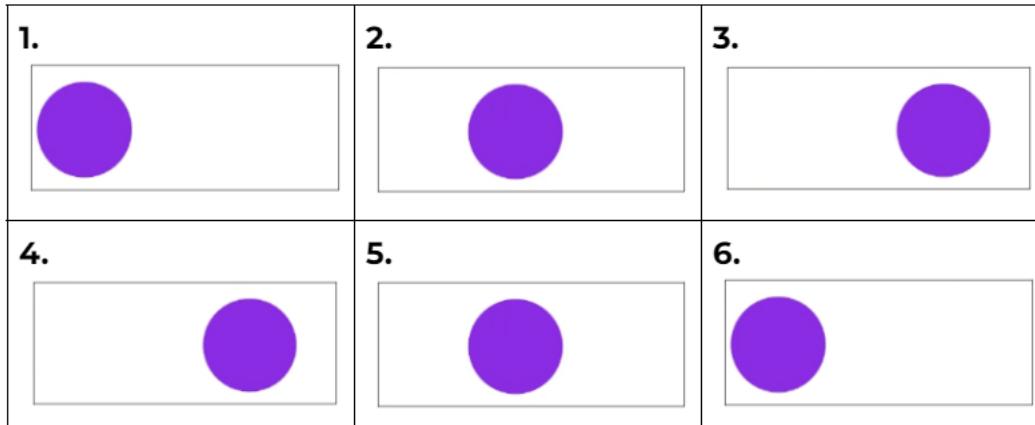
```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <title>Animation</title>
  </head>
  <style>
    .box {
      position: absolute;
      width: 100px;
      height: 100px;
      background-color: blueviolet;
      border-radius: 50px;
      animation: move 1s ease-in-out 0s 1 alternate;
    }

    @keyframes move {
      from {
        left: 0px;
      }

      to {
        left: 100px;
      }
    }
  </style>
  <body>
    <div class="box"></div>
  </body>
</html>
```

Browser Output

Animation Sequence



After 6, It will repeat again from 1.

Example 3:-

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <title>Animation</title>
  </head>
  <style>
    .box {
      position: relative;
      background-color: blueviolet;
      border-radius: 50px;
      height: 100px;
      width: 100px;
    }

    .box1 {
      animation: move 1s ease-in-out 0s 100 alternate;
    }

    .box2 {
      animation: move 1s ease-in-out 500ms 100 alternate;
      top: 100px;
    }

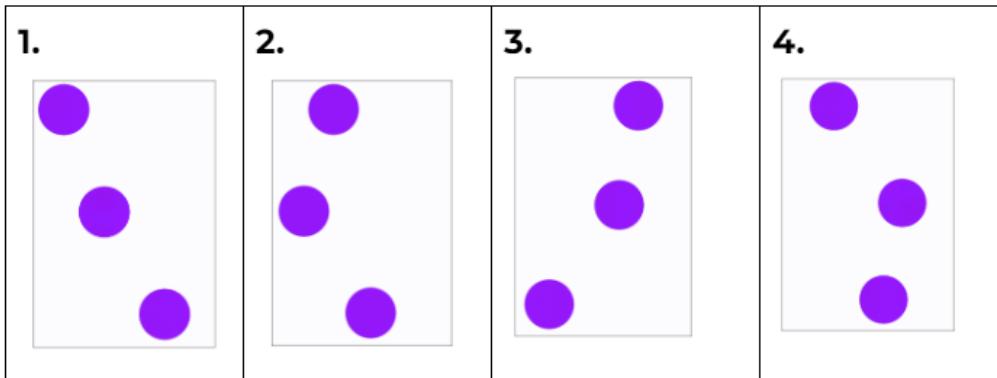
    .box3 {
      animation: move 1s ease-in-out 1000ms 100 alternate;
      top: 200px;
    }

    @keyframes move {
      from {
        left: 0px;
      }

      to {
        left: 200px;
      }
    }
  </style>
  <body>
    <div class="box box1"></div>
    <div class="box box2"></div>
    <div class="box box3"></div>
  </body>
</html>
```

Browser Output

Animation Sequence



It will repeat itself after 4.

Animation vs Transition

Animation	Transition
It moves from the initial to the final state, along with intermediate steps.	Can only move from the initial to the final state.
We can set loop count using the animation-iteration-count property	Run only once
We can run automatically or with a trigger	Run on the trigger (like hover)
Runs forward, in reverse, or alternate directions	On trigger, run forwards, and on leaving the trigger run reverse
Easy for creating a complex series of movements.	Helpful in creating simple movements.