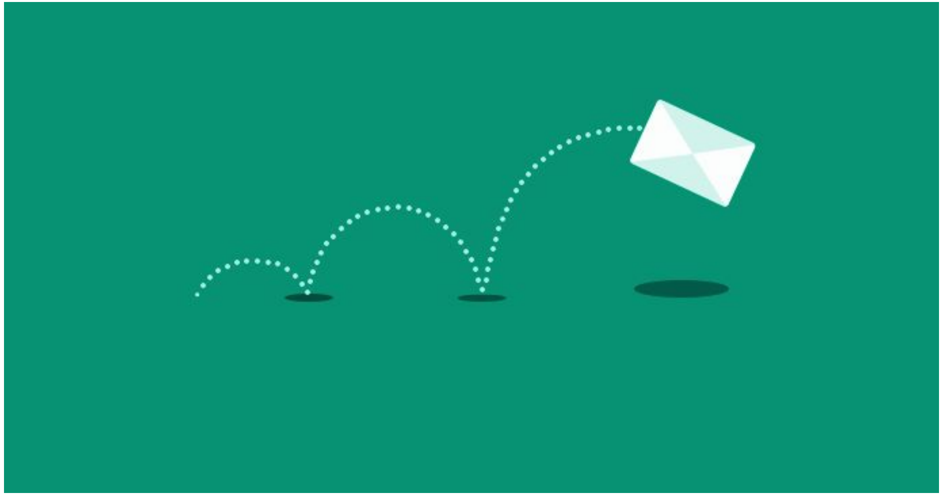
CSS Animations

↓

Replaced animation by Javascript



CSS Animators

- CSS Animations is a technique used to **change** the **appearance** and **behavior** of various elements in web pages.
- It will control the elements by changing their motions or display.
- It has **replaced** the animation created by **Flash** and **JavaScript**.
- The animation is created using the **@keyframe** rule.

  ↳ Time
- It has two parts,
  - **CSS Properties** (describe the animation of the elements)
  - **keyframes (**specific **time intervals** at which the animations have to occur)
- When the animation is created in the **@keyframe** rule, it must have a **selector** otherwise, the animation will have **no effect**.

?

@Keyframes → Time intervals

* Delcaration - foundation of CSS

- Keyframes are the **foundation** of CSS Animations.
- It will control the **intermediate steps** in a CSS animation sequence.
- It defines the display of animation at the corresponding stages in the whole duration.

1st

```
@Keyframes {
0% {color: blue}
25% {color: red}
80% {color: green}
100% {color: orenge}
```

2nd

```
@Keyframes {

from {margin-left: 0px}
to {margin-Left: 200px}

}
```

Animation Properties → mention in declaration

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

Compulsory to apply the property

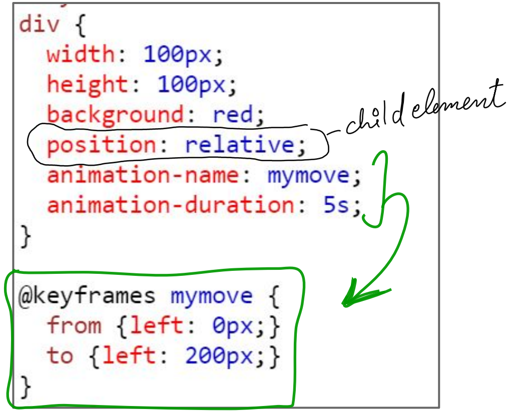
animation : name time (Shorthand)

. / # name , h2 , body etc.

```
Selector {
animation : name time
animation-iteration-count : 3
animation-delay : 3S
}
```

animation-name

- The animation-name property is used to describe the name of the **@keyframe** that has the CSS animation sequence.
- Syntax: ***animation-name: animation _name;***
- Example:



```
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation-name: mymove;
  animation-duration: 5s;
}

@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
}
```

child element

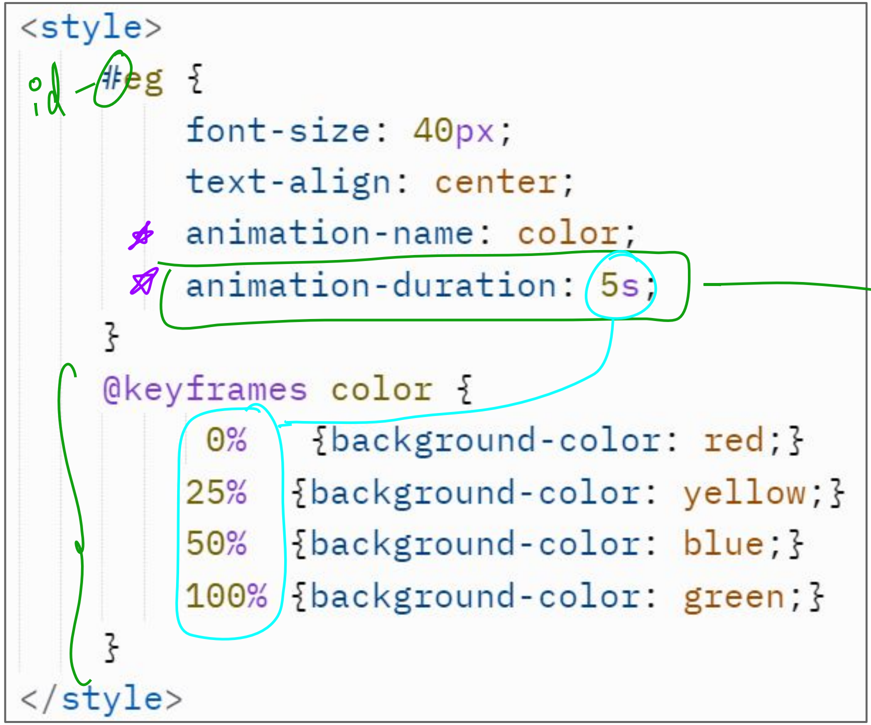
animation-duration

- The animation-duration property specifies the **time duration** of the animation to complete **one** cycle.

- If animation-duration is **not mentioned**, **no animation** will occur because the **default** value is **0 seconds**.

- We can specify animation-duration by using the keywords "**from**" and "**to**" (which represents 0% (start) and 100% (complete)). Instead, we can also use **percent**.

animation-duration



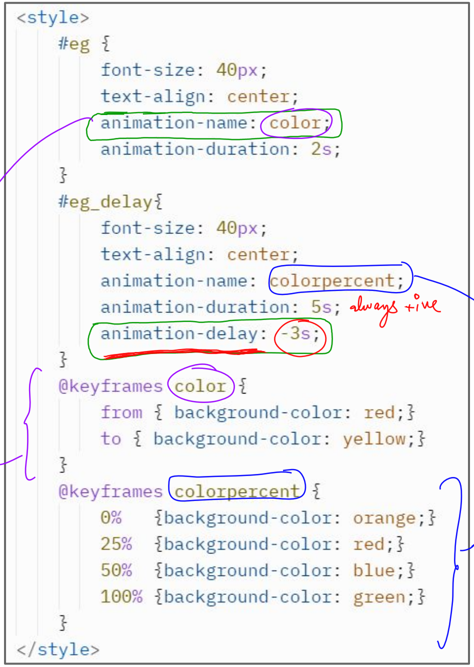
```
<style>
    #eg {
        font-size: 40px;
        text-align: center;
        animation-name: color;
        animation-duration: 5s;
    }
    @keyframes color {
          0%  {background-color: red;}
        25%  {background-color: yellow;}
        50%  {background-color: blue;}
        100% {background-color: green;}
    }
</style>
```

id

animation-delay – Animation थोड़ा late start होगा

- The animation-delay specifies the **delay** when the animation should start.

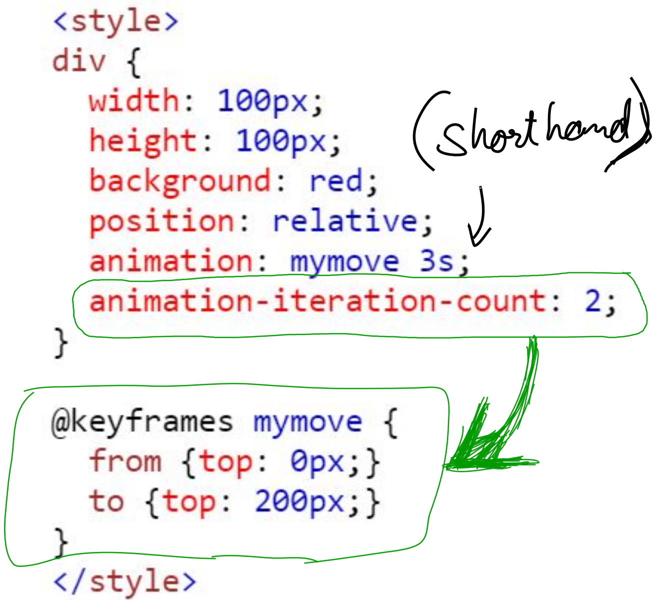
Ques It allows **Negative** values. If using negative values, the animation will be playing as if it has started **already** before *N* seconds.



```
<style>
    #eg {
        font-size: 40px;
        text-align: center;
        animation-name: color;
        animation-duration: 2s;
    }
    #eg_delay{
        font-size: 40px;
        text-align: center;
        animation-name: colorpercent;
        animation-duration: 5s; always +ive
        animation-delay: -3s;
    }
    @keyframes color {
        from { background-color: red;}
        to { background-color: yellow;}
    }
    @keyframes colorpercent {
        0%   {background-color: orange;}
        25%  {background-color: red;}
        50%  {background-color: blue;}
        100% {background-color: green;}
    }
</style>
```

animation-iteration-count

- The animation-iteration-count property specifies the **number of times** an animation should run.
- If we specify the animation-iteration-count value as **infinite**, the animation will repeat **indefinitely**.



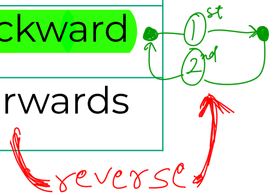
```
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 3s;
  animation-iteration-count: 2;
}

@keyframes mymove {
  from {top: 0px;}
  to {top: 200px;}
}
</style>
```

(Shorthand)

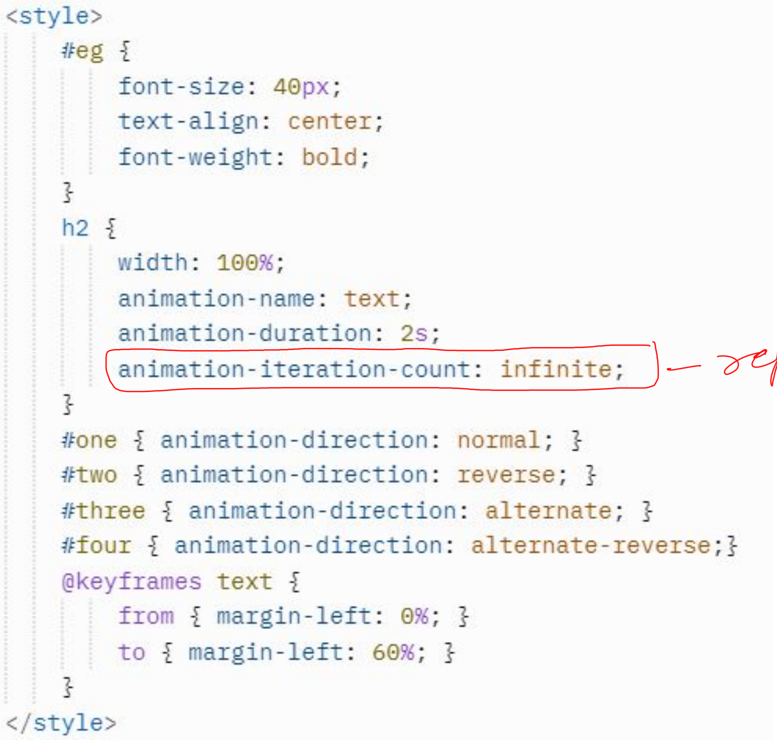
animation-direction

- The animation-direction property specifies the direction of the animation.

Values	Description
normal (default)	The animation is played **forward** →
reverse	The animation is played in the **reverse** direction i.e. backward ←
alternate	The animation is played forwards first, and then backward
alternate-reverse	The animation is played backward first, and then forwards



animation-direction



```
<style>
    #eg {
        font-size: 40px;
        text-align: center;
        font-weight: bold;
    }
    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: 0%; }
        to { margin-left: 60%; }
    }
</style>
```

— repeat continuously

animation-timing-function

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

Values	Description
ease (default)	The animation starts slowly, then fast, and then finally ends slowly
linear	The animation plays with the same speed from start to end
ease-in	The animation plays with a slow start
ease-out	The animation plays with a slow end
ease-in-out	The animation starts and ends slowly
cubic-bezier(n,n,n,n)	Lets you define your own values in a cubic-bezier function

ease
↓
slow

?

animation-fill-mode

Tutorial 36 - Harry
Time: 5:35

- CSS animations **will not** affect an element before the first keyframe is played or after the last keyframe is played. This behavior can be **overridden** by the **animation-fill-mode** property.
- It is used to specify the **style** for the element when the animation is **not playing**.

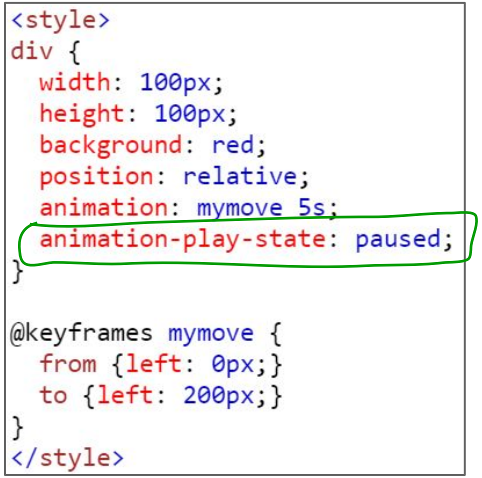
animation-fill-mode

Values	Description
none (default)	The animation **will ~~not~~** apply any styles to the element before or after it is executing
forwards → & stop	The element will retain the style values that are set by the **last keyframe** (depends on **animation-direction** and **animation-iteration-count**)
backwards ← & stop	The element will get the style values that are set by the **first keyframe** (depends on **animation-direction**), and retain this during the **animation-delay** period
both	The animation will follow the rules for **both forwards** and **backwards**, extending the animation properties in both directions

alternate move forward and come back to same position

animation-play-state

- The animation-play-state property allows you to **play/pause** the animation.
- The possible values are: paused, running.
- Syntax: **animation-play-state: paused|running;**



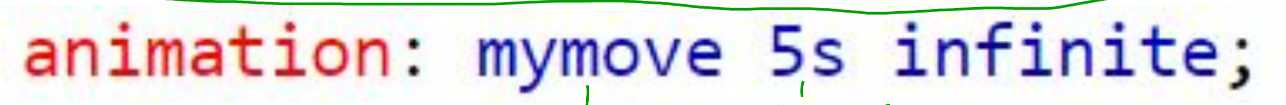
```
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 5s;
  animation-play-state: paused;
}

@keyframes mymove {
  from {left: 0px;}
  to {left: 200px;}
}
</style>
```

Animation Shorthand property

- A **shorthand property** for setting all the animation properties.
- The properties should be in the following order.
- Syntax:

animation: [animation-name] [animation-duration] [animation-timing-function] [animation delay] [animation-iteration-count] [animation-direction] [animation-fill-mode] [animation-play-state];

- Example:



```
animation: mymove 5s infinite;
```

name duration