

UI BASICS

INTERACTION – ANIMATION

CSS ANIMATION

ANIMATION

- The **animation** property in CSS can be used to animate many other CSS properties such as [color](#), [background-color](#), [height](#), or [width](#).
- Each animation needs to be defined with the **@keyframes**. Each **@keyframes at-rule** defines what should happen at specific moments during the animation.

```
.element {  
  animation-name: stretch;  
  animation-duration: 1.5s;  
  animation-timing-function: ease-out;  
  animation-delay: 0s;  
  animation-direction: alternate;  
  animation-iteration-count: infinite;  
  animation-fill-mode: none;  
  animation-play-state: running;  
}
```

```
@keyframes stretch {  
  /* declare animation actions here */  
}
```

```
<div class="element"></div>
```

```
.element {  
  animation:  
    stretch  
    1.5s  
    ease-out  
    0s  
    alternate  
    infinite  
    none  
    running;  
}
```

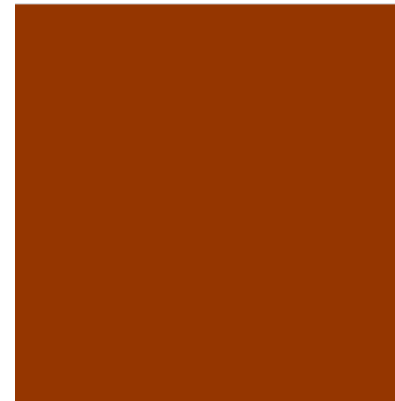
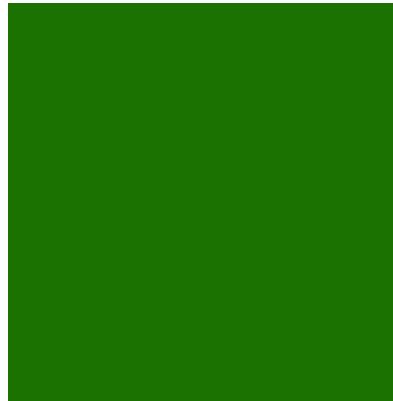
```
@keyframes stretch {  
  /* declare animation actions here */  
}
```

ANIMATION

```
.element {  
  width: 300px;  
  height: 300px;  
  animation: pulse 5s infinite;  
}  
  
@keyframes pulse {  
  from {  
    background-color: green;  
  }  
  to {  
    background-color: red;  
  }  
}
```

```
<div class="element"></div>
```

from



to



ANIMATION

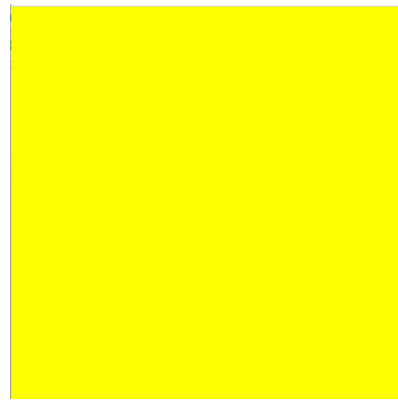
```
.element {  
  width: 300px;  
  height: 300px;  
  animation: pulse 5s infinite;  
}  
  
@keyframes pulse {  
  0% {  
    background-color: green;  
  }  
  25% {  
    background-color: yellow;  
  }  
  50% {  
    background-color: blue;  
  }  
  100% {  
    background-color: red;  
  }  
}
```

```
<div class="element"></div>
```

0%



50%



75%



100%



```
.element {  
  animation-name: stretch;  
  animation-duration: 1.5s;  
  animation-timing-function: ease-out;  
  animation-delay: 0s;  
  animation-direction: alternate;  
  animation-iteration-count: infinite;  
  animation-fill-mode: none;  
  animation-play-state: running;  
}
```

```
@keyframes stretch {  
  /* declare animation actions here */  
}
```

```
.element {  
  animation:  
    stretch  
    1.5s  
    ease-out  
    0s  
    alternate  
    infinite  
    none  
    running;  
}
```

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-direction
- animation-iteration-count
- animation-play-state

- **animation-duration:** - The length of time it takes for an animation to complete one cycle.
 - It defines how long time an animation should take to complete.
 - **0s** by default

animation-duration

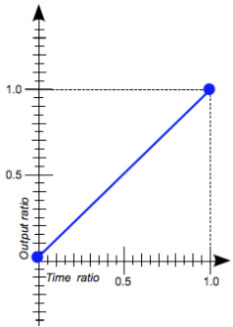
Xs or Xms

NOTE : If the **animation-duration** property is not specified, **no animation will occur**, because the **default value** is **0s** (0 seconds).

- **animation-timing-function**: - It establishes preset acceleration curves such as ease or linear.

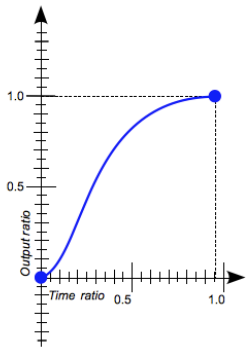
`animation-timing-function`

`ease, ease-out, ease-in, ease-in-out, linear, cubic-bezier(x1, y1, x2, y2)` (e.g. `cubic-bezier(0.5, 0.2, 0.3, 1.0)`)



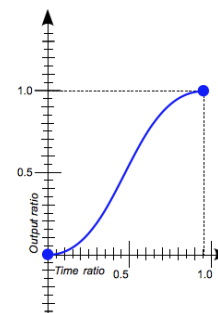
linear : The animation moves from beginning to end at a constant rate.

`cubic-bezier(0.0, 0.0, 1.0, 1.0)`



ease : The animation starts slowly, accelerates sharply, and then slows gradually towards the end. **(by default)**

`cubic-bezier(0.25, 0.1, 0.25, 1.0)`



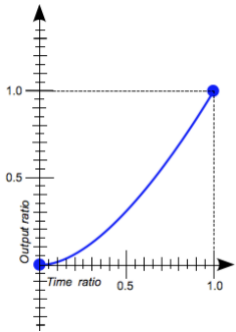
ease-in-out : The animation starts slowly, speeds up, and then slows down towards the end.

`cubic-bezier(0.42, 0, 0.58, 1.0)`

- **animation-timing-function**: - It establishes preset acceleration curves such as ease or linear.

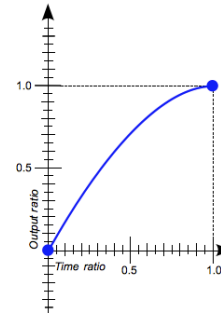
animation-timing-function

ease, ease-out, ease-in, ease-in-out, linear, cubic-bezier(x1, y1, x2, y2) (e.g. cubic-bezier(0.5, 0.2, 0.3, 1.0))



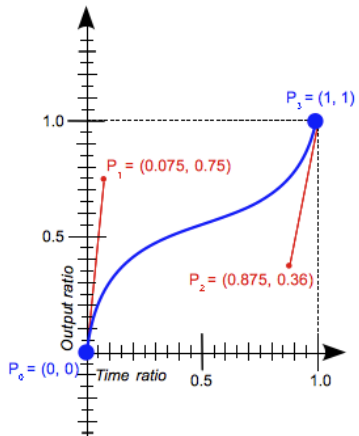
ease-in : The animation starts slowly, and then progressively speeds up until the end, at which point it stops abruptly.

cubic-bezier(0.42, 0, 1.0, 1.0)



ease-out : The animation starts abruptly, and then progressively slows down towards the end.

cubic-bezier(0, 0, 0.58, 1.0)



The **cubic-bezier()** functional notation defines a [cubic Bézier curve](#). As these curves are continuous, they are often used to smooth down the start and end of the animation.

P₀ is (0, 0) and represents the initial time and the initial state, **P₃** is (1, 1) and represents the final time and the final state.

- **animation-delay:** - The time between the element being loaded and the start of the animation sequence.
 - **0s** by default

`animation-delay`

Xs or Xms

NOTE : Negative values are also **allowed**. If using negative values, the animation will start **as if it had already been playing for X seconds**.

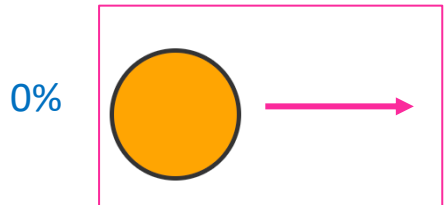
- **animation-direction**: It sets the direction of the animation after the cycle. Its default resets on each cycle.

animation-direction : *normal* , *reverse* , *alternate* , *alternate-reverse* ;

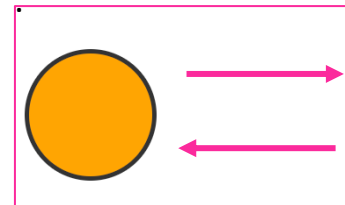
0% → 100% : forwards

100% → 0% : backwards

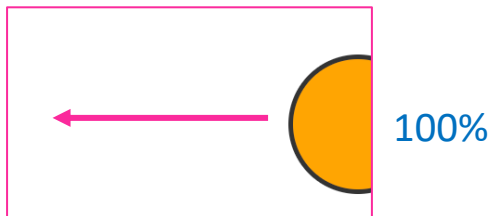
normal : The animation plays **forwards** each cycle. – **by default**



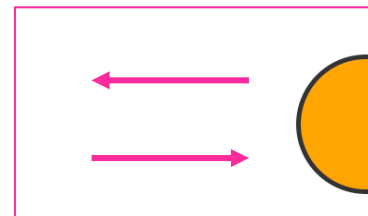
alternate : The animation **reverses direction each cycle**, with the **first iteration** being played **forwards**.



reverse : The animation plays **backwards** each cycle.



alternate-reverse : The animation **reverses direction each cycle**, with the **first iteration** being played **backwards**.



- **animation-iteration-count**: The number of times the animation should be performed.

```
animation-iteration-count : <number> , infinite;
```

- **infinite** : The animation will repeat **forever**.
- **<number>** : - The **number of times** the animation will repeat.
 - This is **1** - by default.
 - You may specify non-integer values to play part of an animation cycle:
for example : **0.5** will play **half of the animation cycle**. **Negative values are invalid**.

- **animation-play-state**: pause/play the animation.

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

- **running** : The **animation** is currently **playing** - **by default**
- **paused** : The **animation** is currently **paused**.

CSS TRANSITION

TRANSITION

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

- **transition-property** - by default **< all >**
- **transition-duration** - by default **< 0s >**
- **transition-timing-function** - by default **< ease >**
- **transition-delay** - by default **< 0s >**

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

```
.transition1 {  
  width: 100px;  
  height: 100px;  
  background: ■lightblue;  
  transition: all 6s, width 2s, height 4s;  
}  
.transition1:hover {  
  width: 300px;  
  height: 300px;  
  background: ■red;  
}
```


NOTE : If the **transition-duration** part is not specified, the transition will **have no effect**, because the **default** value is **0s**.

TRANSITION

- **transition-property** - **< all >**
- **transition-duration** - **< 0s >**
- **transition-timing-function** - **< ease >**
- **transition-delay** - **< 0s >**

```
.transition1 {  
  width: 100px;  
  height: 100px;  
  background: ■ lightblue;  
  transition: all 6s, width 2s, height 4s;  
  margin: 50px;  
}  
.transition1:hover {  
  width: 300px;  
  height: 300px;  
  background: ■ red;  
  margin: 100px;  
}
```

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

- 
- all
 - <property name>

TRANSITION

- transition-property - < all >
- transition-duration - < 0s >
- transition-timing-function - < ease >
- transition-delay - < 0s >

```
.transition1 {  
  width: 100px;  
  height: 100px;  
  background: lightblue;  
  transition: all 6s, width 2s, height 4s;  
  margin: 50px;  
}  
.transition1:hover {  
  width: 300px;  
  height: 300px;  
  background: red;  
  margin: 100px;  
}
```

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

Xs or X00ms

NOTE : If the **transition-duration** part is not specified, the transition will **have no effect**, because the **default** value is **0s**.

TRANSITION

- transition-property - < all >
- transition-duration - < 0s >
- transition-timing-function - < ease >
- transition-delay - < 0s >

```
.transition1 {  
  width: 100px;  
  height: 100px;  
  background: lightblue;  
  transition: all 6s, width 2s, height 4s;  
  margin: 50px;  
}  
  
.transition1:hover {  
  width: 300px;  
  height: 300px;  
  background: red;  
  margin: 100px;  
}
```

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



- ease - 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 slow start and end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

TRANSITION

- transition-property - < all >
- transition-duration - < 0s >
- transition-timing-function - < ease >
- transition-delay - < 0s >

```
.transition1 {  
  width: 100px;  
  height: 100px;  
  background: ■ lightblue;  
  transition: all 6s, width 2s, height 4s;  
  margin: 50px;  
}  
.transition1:hover {  
  width: 300px;  
  height: 300px;  
  background: ■ red;  
  margin: 100px;  
}
```

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

Xs or X00ms

