CSS Animation Properties - Complete Summary Table

📋 Animation Properties Overview

Property	Values	Description	Example
animation-name	`none` | `keyframe-name`	Name of @keyframes animation	`animation-name: slide;`
animation-duration	`time`	How long animation takes	`animation-duration: 2s;`
animation-timing-function	`linear` | `ease` | `ease-in` | `ease-out` | `ease-in-out` | `cubic-bezier()`	Speed curve of animation	`animation-timing-function: ease;`
animation-delay	`time`	Delay before starting	`animation-delay: 1s;`
animation-iteration-count	`number` | `infinite`	How many times to repeat	`animation-iteration-count: 3;`
animation-direction	`normal` | `reverse` | `alternate` | `alternate-reverse`	Direction of animation	`animation-direction: alternate;`
animation-fill-mode	`none` | `forwards` | `backwards` | `both`	Styles before/after animation	`animation-fill-mode: both;`
animation-play-state	`running` | `paused`	Play or pause animation	`animation-play-state: paused;`

🎯 1. animation-name

Value	Description	When to Use
`none`	No animation (default)	Disable animation
`keyframe-name`	References @keyframes rule	Link to your animation

Example:

```css
@keyframes slide {
    from { left: 0; }
    to { left: 100px; }
}

.box {
    animation-name: slide;
}
```

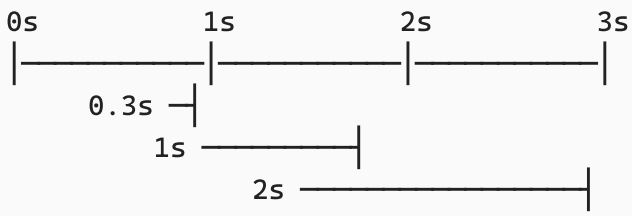
⏱️ 2. animation-duration

Value	Description	Example Use Case
`0s`	Instant (default)	No animation
`0.3s` / `300ms`	Quick animation	Button hover
`1s` / `1000ms`	Normal speed	Card flip
`2s` / `2000ms`	Slow animation	Page transition
`5s+`	Very slow	Background effects

Examples:

```css
.fast { animation-duration: 0.3s; }    /* 300 milliseconds */
.normal { animation-duration: 1s; }    /* 1 second */
.slow { animation-duration: 3s; }      /* 3 seconds */
```

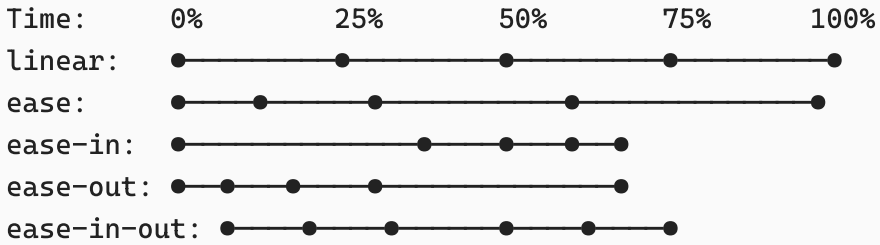
Visual Timeline:



🎨 3. animation-timing-function

Function	Speed Curve	Best For	Visual
`linear`	Constant speed	Loading spinners	▁▁▁▁▁▁▁▁▁▁
`ease`	Slow → Fast → Slow (default)	General animations	▁▂▅▇▅▂▁
`ease-in`	Slow start, fast end	Elements entering	▁▂▅▇▇▇
`ease-out`	Fast start, slow end	Elements exiting	▇▇▇▅▂▁
`ease-in-out`	Slow → Fast → Slow	Smooth transitions	▁▂▅▇▇▅▂▁
`cubic-bezier(x1,y1,x2,y2)`	Custom curve	Special effects	Custom
`steps(n)`	Stepwise (no smooth)	Sprite animations	▇▁▇▁▇▁▇

Speed Comparison:



Examples:

```css
.constant { animation-timing-function: linear; }
.smooth { animation-timing-function: ease; }
.accelerate { animation-timing-function: ease-in; }
.decelerate { animation-timing-function: ease-out; }
.bouncy { animation-timing-function: cubic-bezier(0.68, -0.55, 0.265, 1.55); }
.stepping { animation-timing-function: steps(5); }
```

⏳ 4. animation-delay

Value	Effect	Timeline
`0s`	Start immediately (default)	▶▇▇▇▇▇

Value	Effect	Timeline
`1s`	Wait 1 second	`→████`
`2s`	Wait 2 seconds	`——→██`
`-1s`	Start 1s into animation	`████—`

Positive Delay (wait before starting):

```
.delayed {
    animation-delay: 2s;
}
```

```
Timeline:
0s      1s      2s      3s      4s      5s
|———————|———————|———————|———————|———————|
        Wait            Animation runs
        ————————————    ————————————————
```

Negative Delay (skip beginning):

```
.skip-start {
    animation-delay: -1s;
}
```

```
Timeline:
Animation starts at 1s mark (skips first 1s)
|██████████████|
  ↑ Starts here
```

Staggered Animations:

```
.item:nth-child(1) { animation-delay: 0s; }
.item:nth-child(2) { animation-delay: 0.1s; }
.item:nth-child(3) { animation-delay: 0.2s; }
.item:nth-child(4) { animation-delay: 0.3s; }
```

🔄 5. animation-iteration-count

Value	Description	Visual
`1`	Once (default)	
`2`	Twice	
`3`	Three times	
`infinite`	Forever	
`0.5`	Half cycle	

Examples:

```
.once { animation-iteration-count: 1; }        /* Default */
.three-times { animation-iteration-count: 3; }
.loop { animation-iteration-count: infinite; } /* Never stops */
.half { animation-iteration-count: 0.5; }      /* Stops at 50% */
```

Visual Timeline:

```
iteration-count: 1
Start ──────────→ End
       Animation

iteration-count: 3
Start ──────────→ ──────────→ ──────────→ End
        Cycle 1      Cycle 2      Cycle 3

iteration-count: infinite
Start ──────────→ ──────────→ ──────────→ ...
        Cycle 1      Cycle 2      Cycle 3    (never ends)
```

↔️ 6. animation-direction

Direction	Behavior	Visual
`normal`	0% → 100% (default)	`A ──→ B`
`reverse`	100% → 0%	`A ←── B`
`alternate`	0%→100%, 100%→0%, repeat	`A ──→ B ←── A`
`alternate-reverse`	100%→0%, 0%→100%, repeat	`A ←── B ──→ A`

Detailed Breakdown:

normal (default)

```
Iteration 1: A ————→ B
Iteration 2: A ————→ B
Iteration 3: A ————→ B
```

Always starts at beginning (0%), ends at 100%

reverse

```
Iteration 1: A ←———— B
Iteration 2: A ←———— B
Iteration 3: A ←———— B
```

Always starts at end (100%), goes to 0%

alternate

```
Iteration 1: A ————→ B (forward)
Iteration 2: A ←———— B (backward)
Iteration 3: A ————→ B (forward)
Iteration 4: A ←———— B (backward)
```

Alternates direction each cycle

alternate-reverse

```
Iteration 1: A ←———— B (backward)
Iteration 2: A ————→ B (forward)
Iteration 3: A ←———— B (backward)
Iteration 4: A ————→ B (forward)
```

Starts backward, then alternates

Examples:

```css
@keyframes slide {
    from { left: 0; }
    to { left: 100px; }
}
```

```css
.normal {
    animation: slide 2s normal infinite;
    /* Jumps back to 0, slides to 100px, repeat */
}

.reverse {
    animation: slide 2s reverse infinite;
    /* Jumps to 100px, slides to 0, repeat */
}

.alternate {
    animation: slide 2s alternate infinite;
    /* 0→100px, 100px→0, 0→100px, smooth! */
}

.alternate-reverse {
    animation: slide 2s alternate-reverse infinite;
    /* 100px→0, 0→100px, 100px→0, smooth! */
}
```

📌 7. animation-fill-mode

Mode	Before Animation	During Delay	After Animation	Use Case
`none`	Normal styles	Normal styles	Normal styles	Temporary effects
`forwards`	Normal styles	Normal styles	**Final state**	Keep end result
`backwards`	**First keyframe**	**First keyframe**	Normal styles	Apply start immediately
`both`	**First keyframe**	**First keyframe**	**Final state**	Both effects

Visual Representation:

none (default)

```
Delay     Animation     After
─────     ─────────     ─────
Normal    Animated      Normal
```

forwards

```
Delay    Animation    After
_____    _________    _______________

Normal   Animated     STAYS AT END!
```

backwards

```
Delay         Animation    After
_________     _________    _____

AT START!     Animated     Normal
```

both

```
Delay         Animation    After
_________     _________    _____________

AT START!     Animated     STAYS AT END!
```

Practical Example:

```
@keyframes fadeOut {
    from { opacity: 1; }
    to { opacity: 0; }
}

/* Without forwards - element becomes visible again! */
.disappear-temporary {
    animation: fadeOut 1s none;
    /* opacity: 1 → 0 → 1 (visible again!) */
}

/* With forwards - element stays hidden */
.disappear-permanent {
    animation: fadeOut 1s forwards;
    /* opacity: 1 → 0 (stays hidden!) */
}
```

With Delay:

```
@keyframes slideIn {
    from { transform: translateX(-100px); }
    to { transform: translateX(0); }
```

```
}

/* Without backwards - element jumps at delay end */
.jump {
    animation: slideIn 1s 2s none;
}
/* Timeline:
   0-2s: Normal position
   2s: JUMP to -100px, then animate
*/

/* With backwards - element at start position during delay */
.smooth {
    animation: slideIn 1s 2s backwards;
}
/* Timeline:
   0s: IMMEDIATELY at -100px
   0-2s: Stays at -100px
   2s-3s: Animates to 0
*/
```

⏯️ 8. animation-play-state

State	Description	Use Case
`running`	Animation plays (default)	Normal state
`paused`	Animation paused	Stop on hover

Examples:

```
.box {
    animation: spin 2s linear infinite;
    animation-play-state: running;
}

/* Pause on hover */
.box:hover {
    animation-play-state: paused;
}
```

Interactive Control:

```css
.loading {
    animation: rotate 1s linear infinite;
}

/* Pause when page is loading */
.page-loaded .loading {
    animation-play-state: paused;
}
```

Visual:

```
running: ►──────►──────►──────►
paused:  ►──╫  (stopped at current position)
```

📝 Animation Shorthand

Full Syntax

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

Property Order (can be any order, but common convention):

Position	Property	Required?
1	`name`	✅ Required
2	`duration`	✅ Required
3	`timing-function`	❌ Optional (default: ease)
4	`delay`	❌ Optional (default: 0s)
5	`iteration-count`	❌ Optional (default: 1)
6	`direction`	❌ Optional (default: normal)
7	`fill-mode`	❌ Optional (default: none)
8	`play-state`	❌ Optional (default: running)

Examples:

Minimal (name + duration only)

```
animation: slide 2s;
```

Common Usage

```
animation: slide 2s ease-in-out;
animation: fade 1s ease 0.5s;
animation: spin 2s linear infinite;
```

With Delay

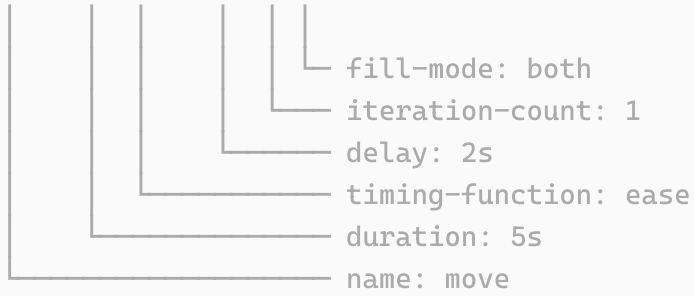
```
animation: bounce 1s ease 2s;
/*         name   dur  func delay */
```

Complete

```
animation: move 2s ease-in 1s 3 alternate both running;
/*         name dur timing  del it dir      fill play   */
```

From Your Code

```
/* Your shorthand example */
animation: move 5s ease 2s 1 both;
/*
```




```
*/
```

Multiple Animations

```
animation:
    fadeIn 1s ease-out,
    slideUp 1s ease-out 0.2s,
    bounce 0.5s ease-in 1s;
```

🎬 Complete Example with All Properties

```css
.box {
    /* Individual properties */
    animation-name: move;
    animation-duration: 5s;
    animation-timing-function: ease;
    animation-delay: 2s;
    animation-iteration-count: 3;
    animation-direction: alternate;
    animation-fill-mode: both;
    animation-play-state: running;
}

/* Equivalent shorthand */
.box {
    animation: move 5s ease 2s 3 alternate both running;
}

/* Pause on hover */
.box:hover {
    animation-play-state: paused;
}

/* The keyframes */
@keyframes move {
    0% {
        transform: translateX(0);
        background-color: red;
    }
    50% {
        transform: translateX(300px);
        background-color: blue;
    }
    100% {
        transform: translateX(0);
        background-color: red;
    }
}
```

⚡ Quick Reference Cheat Sheet

Want to...	Use...
Name the animation	`animation-name: myAnimation;`
Set speed	`animation-duration: 2s;`
Change speed curve	`animation-timing-function: ease;`
Wait before starting	`animation-delay: 1s;`
Loop forever	`animation-iteration-count: infinite;`
Reverse direction	`animation-direction: reverse;`
Ping-pong effect	`animation-direction: alternate;`
Keep final state	`animation-fill-mode: forwards;`
Apply start during delay	`animation-fill-mode: backwards;`
Pause animation	`animation-play-state: paused;`

🎯 Common Patterns

1. Simple Fade In

```
@keyframes fadeIn {
    from { opacity: 0; }
    to { opacity: 1; }
}

.fade {
    animation: fadeIn 1s ease;
}
```

2. Infinite Spinner

```
@keyframes spin {
    from { transform: rotate(0deg); }
    to { transform: rotate(360deg); }
}

.spinner {
    animation: spin 2s linear infinite;
}
```

3. Bounce Effect

```css
@keyframes bounce {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-50px); }
}

.ball {
    animation: bounce 1s ease alternate infinite;
}
```

4. Pulse (Heartbeat)

```css
@keyframes pulse {
    0%, 100% { transform: scale(1); }
    50% { transform: scale(1.1); }
}

.heart {
    animation: pulse 1.5s ease-in-out infinite;
}
```

5. Delayed Sequence

```css
.item1 { animation: slideIn 0.5s ease 0s forwards; }
.item2 { animation: slideIn 0.5s ease 0.2s forwards; }
.item3 { animation: slideIn 0.5s ease 0.4s forwards; }
.item4 { animation: slideIn 0.5s ease 0.6s forwards; }
```

🎓 Summary Table - All Values

Property	Default	Common Values
`animation-name`	`none`	`keyframe-name`
`animation-duration`	`0s`	`0.3s`, `1s`, `2s`, `5s`
`animation-timing-function`	`ease`	`linear`, `ease-in`, `ease-out`, `ease-in-out`
`animation-delay`	`0s`	`0s`, `0.5s`, `1s`, `2s`, `-1s`
`animation-iteration-count`	`1`	`1`, `2`, `3`, `infinite`
`animation-direction`	`normal`	`reverse`, `alternate`, `alternate-reverse`

Property	Default	Common Values
`animation-fill-mode`	`none`	`forwards`, `backwards`, `both`
`animation-play-state`	`running`	`paused`

💡 Best Practices

Performance

✅ **DO:** Animate `transform` and `opacity`

```
/* Good - GPU accelerated */
animation: slide 1s;
@keyframes slide {
    from { transform: translateX(0); }
    to { transform: translateX(100px); }
}
```

❌ **DON'T:** Animate `width`, `height`, `margin`, `top`, `left`

```
/* Bad - triggers layout recalculation */
animation: expand 1s;
@keyframes expand {
    from { width: 100px; }
    to { width: 200px; }
}
```

Timing

- **Quick interactions:** 200-300ms
- **Normal animations:** 500ms-1s
- **Emphasis effects:** 1-2s
- **Background effects:** 3s+

Fill Mode

- Use `forwards` to keep final state
- Use `both` when you have a delay and want smooth start

Iteration

- Use specific numbers for effects (1, 2, 3)
- Use `infinite` only for loaders/spinners
- Avoid infinite on many elements (performance)

This is your complete animation reference guide! 🚀