

CSS Animating

Text from: [Learning Web Design - Jennifer Niederst Robbins](#)

CSS TRANSITIONS

CSS Transitions smooth out otherwise abrupt changes to property values between two states over time by filling in the frames in between. Animators call that tweening. When used with reserve, CSS Transitions can add sophistication and polish to your interfaces and even improve usability.

When applying a transition, you have a few decisions to make, each of which is set with a CSS property:

- Which CSS property to change (**transition-property**) (Required)
- How long it should take (**transition-duration**) (Required)
- The manner in which the transition accelerates (**transition-timing-function**)
- Whether there should be a pause before it starts (**transition-delay**)

Transitions require a **beginning state** and an **end state**. The element as it appears when it first loads is the beginning state. The end state needs to be triggered by a state change such as **:hover**, **:focus**, or **:active...**

CSS Animation Selectors

: hover

: focus

: active

transition-property

identifies the CSS property that is changing and that you want to transition smoothly. In our example, it's the background-color. You can also change the foreground color, borders, dimensions, font- and text-related attributes, and many more. TABLE 18-1 lists the animatable CSS properties as of this writing. The general rule is that if its value is a color, length, or number, that property can be a transition property.

Backgrounds	Font and text	Element box measurements
background-color	font-size	height
background-position	font-weight	width
	letter-spacing	max-height
	line-height	max-width
Borders and outlines	text-indent	min-height
border-bottom-color	text-shadow	min-width
border-bottom-width	word-spacing	margin-bottom
border-left-color	vertical-align	margin-left
border-left-width		margin-top
border-right-color	Position	padding-bottom
border-right-width	top	padding-left
border-top-color	right	padding-right
border-top-width	bottom	padding-top
border-spacing	left	
outline-color	z-index	
outline-width	clip-path	
Color and opacity	Transforms	
color	transform	
opacity	transform-origin	
visibility		
		Animateable CSS Properties

Timing Functions

```
.thisAwesomeClass {
```

```
  transition-timing-function :
```

the css property

```
    ease
```

```
    linear
```

```
    ease-in
```

```
    ease-out
```

possible values you can set

```
    ease-in-out
```

```
    step-start
```

```
    step-end
```

```
    steps
```

```
    cubic-bezier(##,##,##,##)
```

```
}
```

The **property** and the **duration** are required and form the foundation of a transition, but you can refine it further. There are a number of ways a **transition** can roll out over time.

For example, it could **start out fast** and then **slow down**, **start out slow** and **speed up**, or **stay the same speed all the way through**, just to name a few possibilities. I think of it as the transition “style,” but in the spec, it is known as the timing function or easing function.

The timing function you choose can have a big impact on the feel and believability of the animation, so if you plan on using transitions and CSS animations, it is a good idea to get familiar with the options.

ease

Starts slowly, accelerates quickly, and then slows down at the end. This is the default value and works just fine for most short transitions.

linear

Stays consistent from the transition's beginning to end. Because it is so consistent, some say it has a mechanical feeling.

ease-in

Starts slowly, then speeds up.

ease-out

Starts out fast, then slows down.

ease-in-out

Starts slowly, speeds up, and then slows down again at the very end. It is similar to ease, but with less pronounced acceleration in the middle.

cubic-bezier(x1,y1,x2,y2)

The acceleration of a transition can be plotted with a curve called a Bezier curve. The steep parts of the curve indicate a fast rate of change, and the flat parts indicate a slow rate of change.

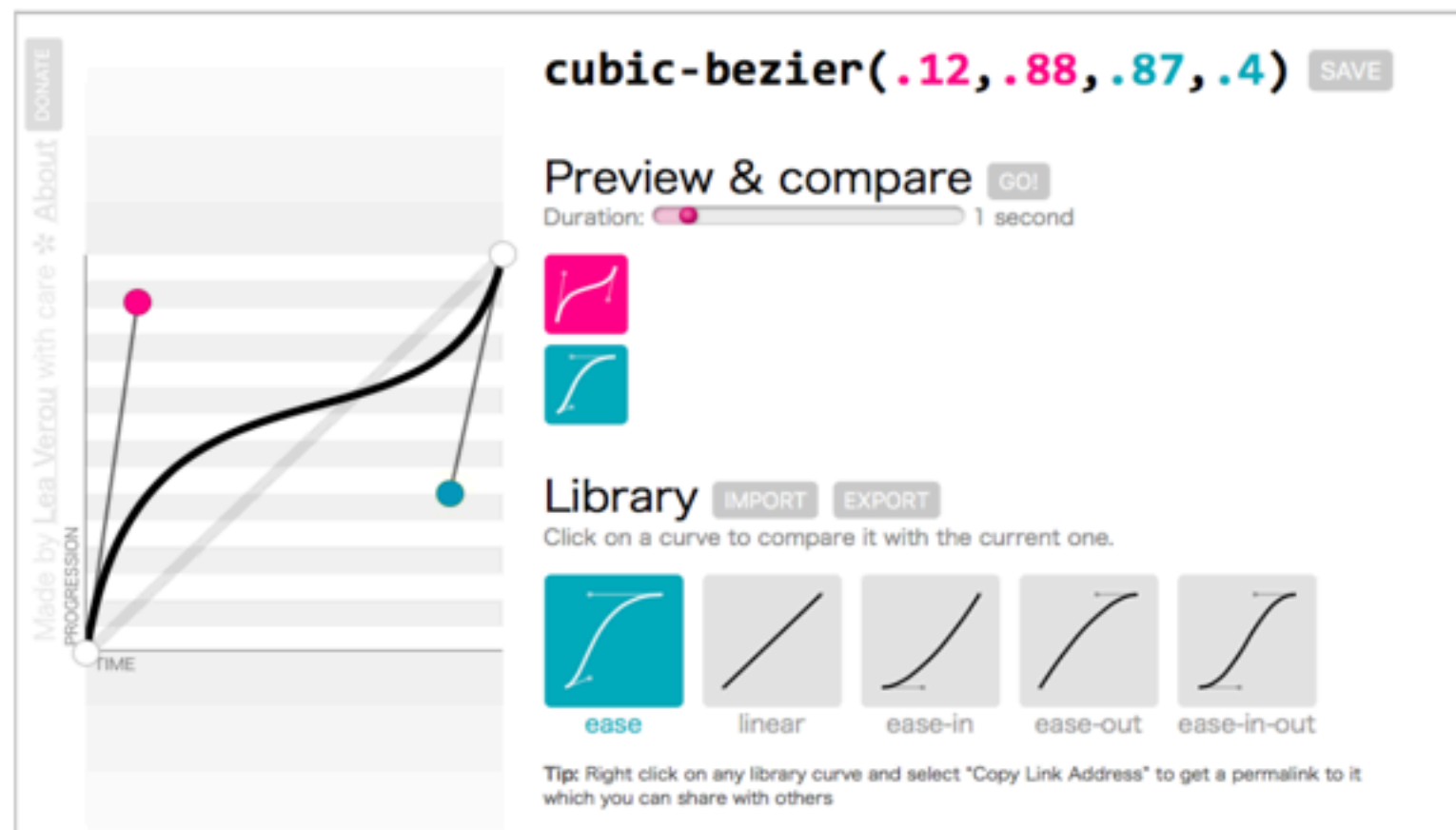


FIGURE 18-2. Examples of Bezier curves from Cubic-Bezier.com. On the left is my custom curve that starts fast, slows down, and ends fast.

You can see that the ease curve is a tiny bit flat in the beginning, gets very steep (fast), then ends flat (slow). The linear keyword, on the other hand, moves at a consistent rate for the whole transition.

You can get the feel of your animation just right by creating a custom curve. The site [Cubic-Bezier.com](https://cubic-bezier.com) is a great tool for playing around with transition timing and generating the resulting code. The four numbers in the value represent the x and y positions of the start and end Bezier curve handles (the pink and blue dots).

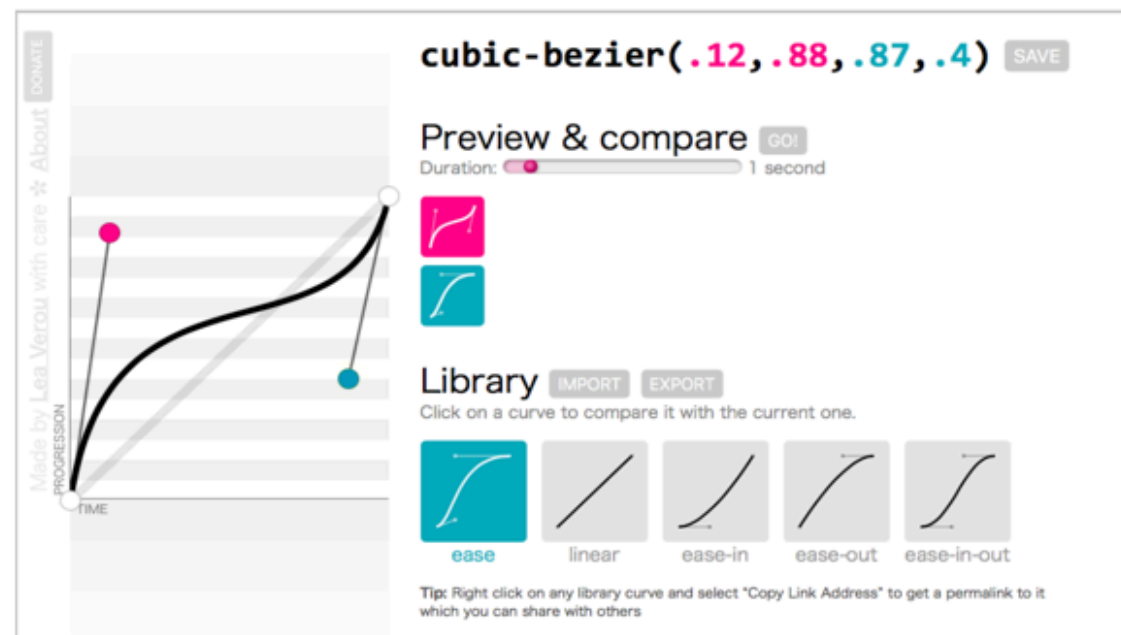


FIGURE 18-2. Examples of Bezier curves from Cubic-Bezier.com. On the left is my custom curve that starts fast, slows down, and ends fast.

steps(#, start|end)

Divides the transitions into a number of steps as defined by a stepping function. The first is the number of steps, and the **start** and **end** keywords define whether the change in state happens at the beginning (**start**) or end of each step. Step animation is especially useful for keyframe animation with sprite images. For a better explanation and examples, I recommend the article "Using Multi-Step Animations and Transitions," by Geoff Graham on CSS-Tricks (css-tricks.com/using-multi-step-animations-transitions/).

step-start

Changes states in one step, at the beginning of the duration time (the same as **steps(1, start)**). The result is a sudden state change, the same as if no transition had been applied at all.

step-end

Changes states in one step, at the end of the duration time (the same as **steps(1, end)**).

transition-delay

The transition-delay property, as you might guess, delays the start of the animation by a specified amount of time.

The Shorthand transition Property

The authors of the CSS3 spec had the good sense to give us the shorthand transition property to combine all of these properties into one declaration. You've seen this sort of thing with the shorthand border property. Here is the syntax:

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

```
.theClass {
```

```
    transition: background-color 0.3s ease-in-out 0.2s;
```

```
}
```

The values for each of the **transition-*** properties are listed out, separated by character spaces. The order isn't important as long as the **duration** (which is required) appears before **delay** (which is optional). If you provide only one time value, it will be assumed to be the duration.