e.g. flexbox

CSS animations are rad and the concept is fairly simple. Name the animation, define the movement in `@keyframes` and then call that animation on an element. If you haven't worked with them, you can level up on the syntax right here in the Almanac.

While the concept is simple, there are little tricks to make the animations seem complex and one of those is multi-step transitions. That's what we're going to look at in this post.

A keyframe can be a "step"

If we set up a keyframe animation to change the background color of an element to change from orange to black (because orange *is* the new black, after all) on hover over five seconds, it will do exactly that. It will divide that change up over time and make the transition.

CSS

```css
body {
  background: black;
  animation: color-me-in 5s; /* other keywords like "infinite alternate" can be usef
}

@keyframes color-me-in {
  /* You could think of as "step 1" */
  0% {
    background: orange;
  }
  /* You could think of as "step 2" */
  100% {
    background: black;
  }
}
```

Result ▼

Edit on

We can add as many steps as we like in a keyframe animation. For example, here is blue being added right in the middle of the transition.

CSS

```css
@keyframes color-me-in {
  0% {
    background: orange;
  }
  /* Adding a step in the middle */
  50% {
    background: blue;
  }
  100% {
    background: black;
  }
}
```

Result ▼

Edit on

That's the concept of multi-step animations in a nutshell: more than one change taking place in the animation from start to finish.

#Keyframes are interpolated and tweened, unless you don't want them to be

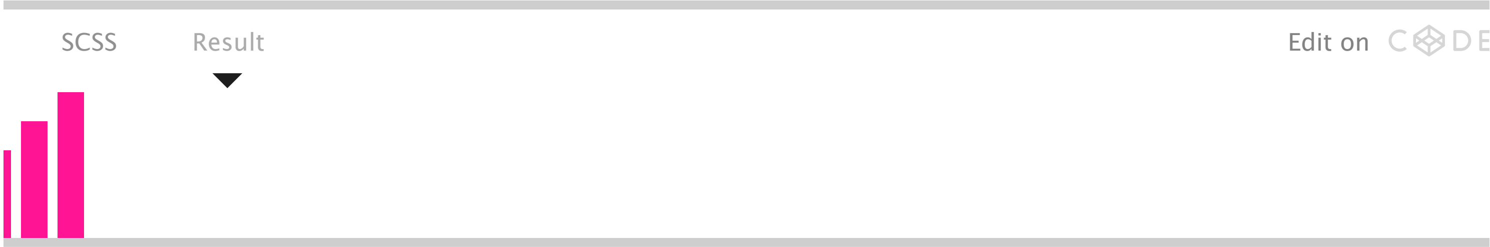
Notice the colors above fade into one another in each of the steps. That's a good default. They even "ease" into each other, as that's the default `animation-timing-function` , another good default as it's a lot smoother and relaxed feeling that than the computer-y feeling `linear` in general, although everything has it's use.

Using the steps() function you can force the interpolation to be an exact number of keyframes. Demo:

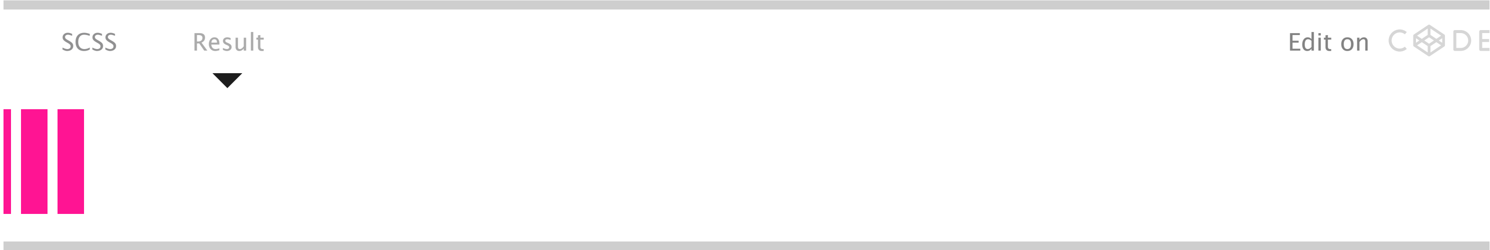
#Putting multi-step animations to use

A good example is the sound equalizer in Apple Music. You've likely seen it or at least something like it: a series of vertical bars that wave up and down in a way that gives the illusion of it dancing to the beat of the music.

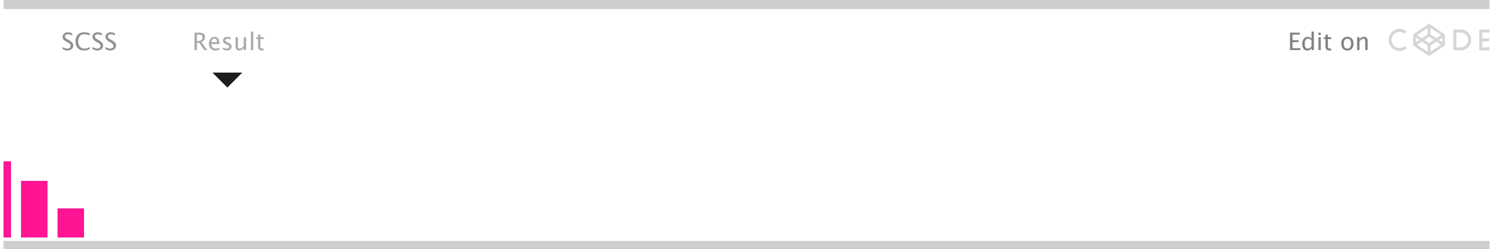
Here's a non-moving version of what we're talking about:



Five vertical bars and want to animate them moving up and down at various heights. Let's create an animation called `equalize` with 25 keyframe steps. That's one step for every `4%` of the animation, which is then called to the `.bar` element:

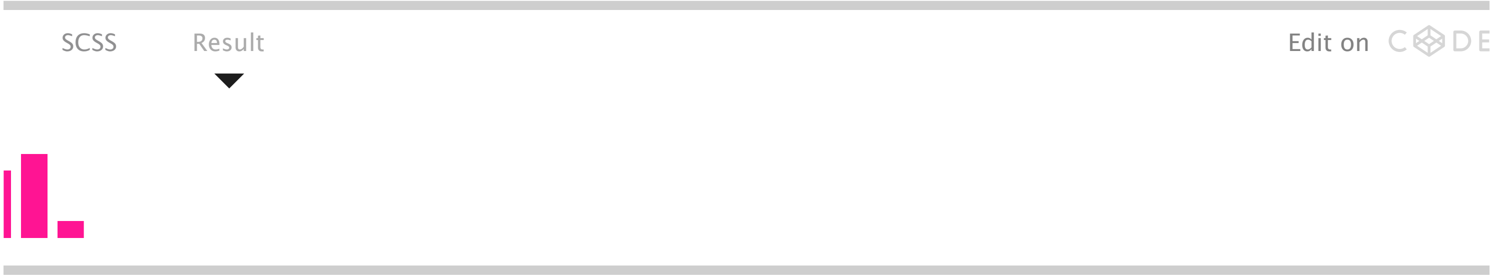


Great, but they're all moving at the same pace at the same time. Not exactly a cool equalizer to look at. However, we can add a different `animation-duration` to each `.bar` when the animation is called to give the effect that they are moving at different paces.



There we have it! An equalizer that, when active, looks as though it could be moving to the rhythm of a song, even though we are using the same multi-step animation on each element.

And here's an alternative that plays with `animation-delay` (negative values, so they all still start at the same time) instead:



#But wait, there's also multi-step `transitions`!

`Transition`s are just like simpler animations. They move from one fixed set of properties to another. The shorthand:

CSS

```
.move-me {
  transition: [transition-property] [transition-duration] [transition-timing-functic
}
```

And, just like `animation` we can animate the same set of properties.

Let's transition a red box to an orange rectangle on hover by chaining the `background-color` and `width` properties:

CSS

```css
.box {
  width: 150px;
  height: 150px;
  background-color: red;
  transition: 1s;
}

.box:hover {
  width: 300px;
  background-color: orange;
}
```

This tells our `.box` element to transition both the `width` and `background-color` properties on hover and to make the transition in one second.

SCSS Result

Edit on

Not exactly multi-step yet, but it's possible!

#Adding multiple steps to transitions

We can make things a little more interesting by chaining our transitions together using commas, then playing with the duration and delay of them to create the same sort of multi-step movement effect that is possible in keyframe animation.

Let's take the above example and make it change width *then after that's done* change color.

```
CSS

.box {
  transition:
    /* step 1 */
    width      1s,
    /* step 2 */
    background 0.5s 1s;
}
```

SCSS Result Edit on CODE

We can get much fancier here and choreograph additional movements.

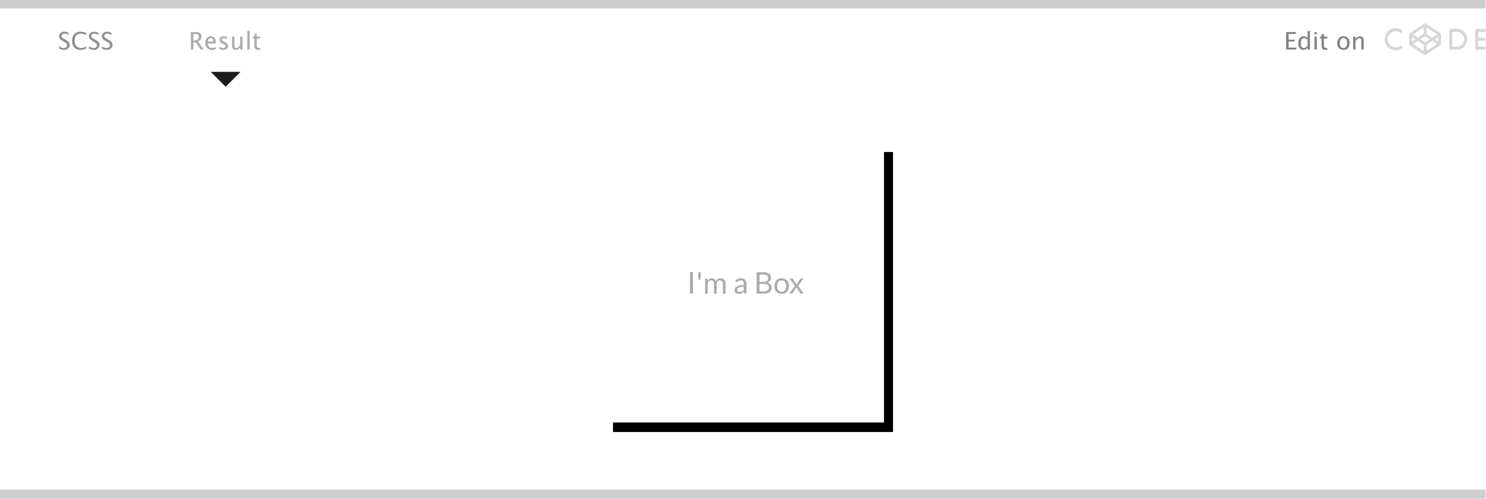
- Change the width from `150px` to `300px` immediately on hover for 1 second
- Change the `background-color` from red to orange immediately for 1 second
- Add a `box-shadow` and change the direction of it immediately on hover for 1 second
- Add one line of text that fades out and is pushed away to the left after the `width` and `height have transitioned`
- Add another line of text that appears and swoops in from the right after the first line of text has disappeared

We can do this by chaining our transitions on each element where there are multiple properties to transition!

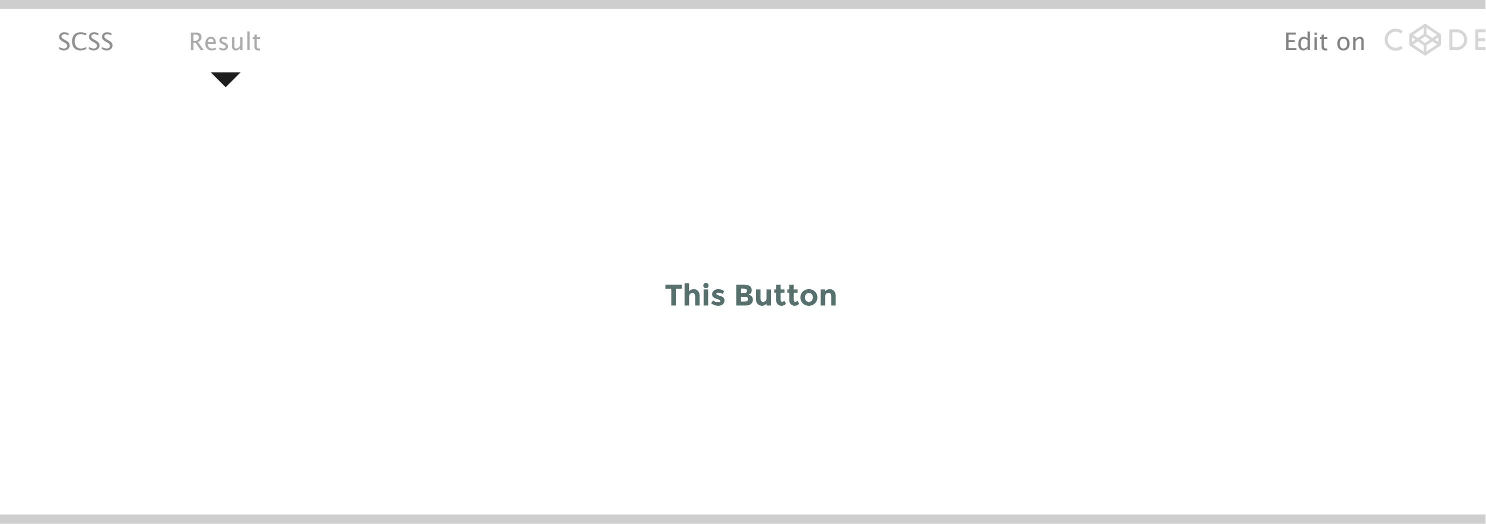
CSS

```
/* Our box element */
.box {
  width: 150px;
  height: 150px;
  background-color: red;
  box-shadow: 5px 5px 25px #000;
  transition: all 1s;
}

/* On hover... */
.box:hover {
  /* Increase width */
  width: 300px;
  /* Change color */
  background-color: orange;
  /* Move the shadow */
  box-shadow: -5px 5px 25px #000;
}

/* The first line of text */
.box__blurb {
  /* Start with full opacity and centered */
  opacity: 1;
  transform: translateX(45px);
  /* Then transition these chained properties */
  transition:
    opacity 0.5s 2s,
    transform 0.5s 0.5s;
}

/* On .box hover... */
.box:hover .box__blurb {
  /* Fade out */
  opacity: 0;
  /* Move over to the right */
```

Here's what we've got. Some flourishes were added for presentation:



Here's another example of a button where all the hover effects are clearly staged into steps:



Wrapping up

Multi-step animations and transitions are fun little tricks we have at our disposal to create rich, dynamic movement in CSS. The example of an equalizer in this post is a practical application but there are many other ways that multi-step transitions can be used. In fact, Ana Tudor used them to make a mind-blowing animation in a post she published here on CSS-Tricks. Rémi Lacorne has a sweet example of transitions as well.

Where have you used multi-step animations and transitions in your CSS animations? Have you made something super and complex that shows off your animation chops? Or maybe a subtle touch that enhances the user experience? Let us know in the comments.