Restart CSS Animation

css-tricks.com/restart-css-animation/

Published May 5, 2011 by Chris Coyier

With CSS animations (ala @keyframes) it's not as easy as you might think to "restart" it.

Let's say you set it to run once:

```
.run-animation {
 -webkit-animation: my-fancy-animation 5s 1;
 -moz-animation: my-fancy-animation 5s 1;
 -o-animation: my-fancy-animation 5s 1;
 animation: my-fancy-animation 5s 1;
}
```

You have that run on, say, your logo:

```
<h1 id="logo" class="run-animation">
 Fancy Logo
</h1>
```

Then for some reason you want to have that animation run again, perhaps on a click. You might think CSS provides a way to kick it off again, but as far as I know it doesn't. You might even try removing and re-adding that class name via JavaScript (jQuery):

```
$("#logo").click(function() {
 // not gonna work
 $(this).removeClass("run-animation").addClass("run-animation");
});
```

If you put a tiny delay between removing and re-adding the class (e.g. setTimeout()), it will work, but I think it goes without saying that that isn't ideal. What you really need to do is remove the element from the page entirely and re-insert it. As a quick demo with jQuery, we'll clone it, insert it right after itself, then remove the original.

```
$("#logo").click(function() {

 var el = $(this),
 newone = el.clone(true);

 el.before(newone);

 $("." + el.attr("class") + ":last").remove();
});
```

or the basics with just JavaScript:

```
var elm = this,
var newone = elm.cloneNode(true);
elm.parentNode.replaceChild(newOne, elm);
```

This same problem exists for any type of event. Say you have an animation you want to run immediately, but then again on :hover. Same problem, it won't run again. Oli Studholme has some interesting research on this. One way around it is by having two animations that are exactly the same except for their name, then you can switch which one you use for the :hover event (or whatever) and it will work.

```
img {animation: spin-cat-1 2s;}
img:active {animation: spin-cat-2 2s;}
@keyframes spin-cat-1 {50%{transform:rotateY(180deg);}}
@keyframes spin-cat-2 {50%{transform:rotateY(180deg);}}
```

A CSS preprocessor could help make maintaining that easier by importing a chunk of code so you only have to maintain it in once place, but ultimately it's bloat-y. Unfortunately you also cannot define multiple keyframe animation names in one declaration, which would also be useful (and not just for this, think of more complex animations where you could rewrite just certain keyframes over a base keyframe animation).

Oli also played with a variety of other techniques to get it to work, like altering duration, delays, and iterations instead of changing the name, but in my opinion all of those are even worse than changing the name.

Update: Another JavaScript Method to Restart a CSS Animation

Trigger a reflow in between removing and adding the class name. For example:

```
// retrieve the element
element = document.getElementById("logo");
// reset the transition by...
element.addEventListener("click", function(e) {
 e.preventDefault;

 // -> removing the class
 element.classList.remove("run-animation");

 // -> triggering reflow /* The actual magic */
 // without this it wouldn't work. Try uncommenting the line and the transition won't be retriggered.
 element.offsetWidth = element.offsetWidth;

 // -> and re-adding the class
 element.classList.add("run-animation");
}, false);
```

Check out this Pen!

Other JavaScript

Just for the record, you can also effect the playing state of a CSS animation through JavaScript, like pausing and restarting it.

```
element.style.webkitAnimationPlayState="paused";
element.style.webkitAnimationPlayState="running";
```

Obviously consider the other vendor prefixes properties.

Interestingly, the natural end of a CSS animation doesn't set the play state to "paused". You would have to do that yourself. Say you wanted to restart an animation for every hover and you were cool using jQuery to do it...

```
$("#thing").hover(function() {

 this.style.webkitAnimationPlayState = "running";
 $(this).on('webkitAnimationEnd', function() {
 this.style.webkitAnimationPlayState = "paused";
 });

});
```

Notice how you have to watch for the AnimationEnd event to fire to set it back to "paused" or it won't restart properly.