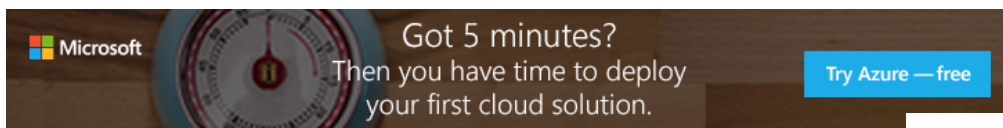


Join the Stack Overflow Community

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.
Join them; it only takes a minute:

[Sign up](#)

Multiple CSS keyframe animations using transform property not working



While working with CSS *keyframe animations* I found that when I give an element two animations like:

```
.element {
  animation: animate1 1000ms linear infinite,
            animate2 3000ms linear infinite;
}
```

If both of the animations animate using the `transform` property only the last one triggers through cascading. But if the `@keyframes` animations are lets say one `margin` or `display` or some other `css` attribute and the other uses `transform` then they both trigger.

[here is a codepen example](#) with the relevant code below.

CSS

```
@keyframes move {
  0%, 100% { transform: translateX(0px); }
  50% { transform: translateX(50px); }
}
@keyframes skew {
  0%, 100% { transform: skewX(0deg); }
  50% { transform: skewX(15deg); }
}
@keyframes opacity {
  0%, 100% { opacity: 1; }
  50% { opacity: .25; }
}

.taco {
  animation: skew 2000ms linear infinite,
            opacity 4000ms linear infinite;
}
```

In the Above they both trigger

```
.burger {
  animation: skew 2000ms linear infinite,
            move 4000ms linear infinite;
}
```

In the above only the last triggers (through cascading)- why?

Anyone have a solution for this without using `js` ? Or is this something that just doesn't work? The example is quite simple and I know that I could combine the animations into one and not have to declare both but this was a test for a more complex animation I was working on.

thanks

[css](#) [css3](#) [css-animations](#)

edited Apr 22 '14 at 16:00



Volker E.

3,084 8 27 45

asked Jan 17 '13 at 17:16



danferth

726 4 10 22

3 Answers

You cannot animate same attribute (here transform attribute) more than once, on a same element, the last one will overwrite other,

You should put your target element into a div, and apply one transform-animation on the div and other transform-animation on the target element....

```
.div_class
{
  animation:animate1 1000ms linear infinite;
}

.element
{
  animation:animate2 3000ms linear infinite;
}
```

answered Jan 19 '13 at 17:34



Swarnendu Paul

2,733 2 15 40

thanks for the idea of wrapping in a div, the two transform animations needed different timings. – danferth Feb 2 '13 at 0:16

"You cannot animate same attribute more than once on a same element" This doesn't seem entirely accurate though, see: jsfiddle.net/j83m5ha5/4. Here the same element chains the same attribute on the same class. Why does this work, but not with transform? – Stephan Muller Apr 6 '15 at 10:06

@StephanMuller , I have checked your animation, and the second animation (out) is overwriting the first one (in). Just increase the duration of first animation , lets say make it 10 sec, and see the effect , you will find that it is a 1 min animation, because 2nd animation is overwriting the 1st one.. – Swarnendu Paul Apr 22 '15 at 17:43



Find your dream job

on a career site built just for developers



stackoverflow
JOBS

Get started

For the same reason that

```
transform: skewX(45deg);
transform: translateX(4em);
```

won't skew the element but will only move it. And if you want to both skew and move it then you need to chain them transform: skewX(45deg) translateX(4em)

You'll have to do something like

```
@keyframes t {
  25% { transform: skewX(15deg); }
  50% { transform: skewX(0deg) translateX(50px); }
  75% { transform: skewX(15deg); }
}
```

You don't need to explicitly specify 0% and 100% keyframes - they'll be automatically generated - see [the CSS Animations spec](#).

And then you can use

```
animation: t 4000ms linear infinite,
  opacity 4000ms linear infinite;
```

One more thing you should be careful about: skewX(angleValue) translateX(lengthValue) happens to be the same as translateX(lengthValue) skewX(angleValue) . However, most of the times, **the order in which you apply transforms matters**. You'll get different results for skewX(angleValue) translateY(lengthValue) and translateY(lengthValue) skewX(angleValue) .

edited Jan 17 '13 at 20:51

answered Jan 17 '13 at 18:56



Ana

26.6k 6 50 92

@StephanMuller's jsfiddle is pretty much the answer, that syntax has worked for me <http://jsfiddle.net/j83m5ha5/4>