CSS-TRICKS

Animating SVG with CSS

BY CHRIS COYIER ON APRIL 17, 2014

ANIMATION, SVO

There isn't just one way to animate SVG. There is the <animate> tag that goes right into the SVG code. There are libraries that help with it like Snap.svg or SVG.js. We're going to look at another way: using inline SVG (SVG code right inside HTML) and animating the parts right through CSS.

I played with this personally recently as my Alma mater was looking to freshen up the advertising graphic we're running here. My latest design around here uses SVG quite a bit and I thought this would be another perfect opportunity to use it some more.

The finished product is pretty simple. Here's it is:



Let's check out how it's done.

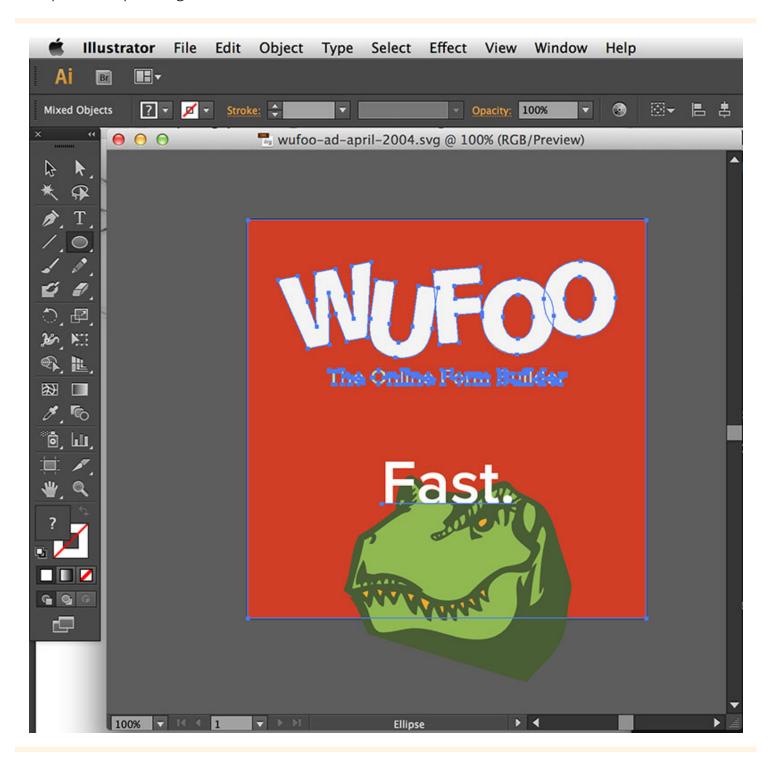
$^{\scriptscriptstyle \#}$ 1. Design the Ad / Have a Plan

This might seem like an how to draw an owl moment, but this article is about animation so let's get there as quickly we can.

My plan for this ad was to make a super simple Wufoo ad with their classic logo, colors, and general branding. Then add a little flair.

- 1. Make the letters kind hop off the page a little. Wufoo is a fun word, let the letters have some fun.
- 2. Back in the day we made a T-Shirt with dinosaurs on the front and on the back it said "Fast. Smart. Formidable." Which are traits of both dinosaurs and Wufoo, not to mention the fun little play on words with FORMidble. Let's make those fade in and out.
- 3. To make the connection with the dinosaur thing complete, we'll have a T-Rex head pop up from the bottom curiously, then zoom away. The word "Fast." will come in as he zooms away which is another nice little connection.

I put all the parts together in Illustrator.

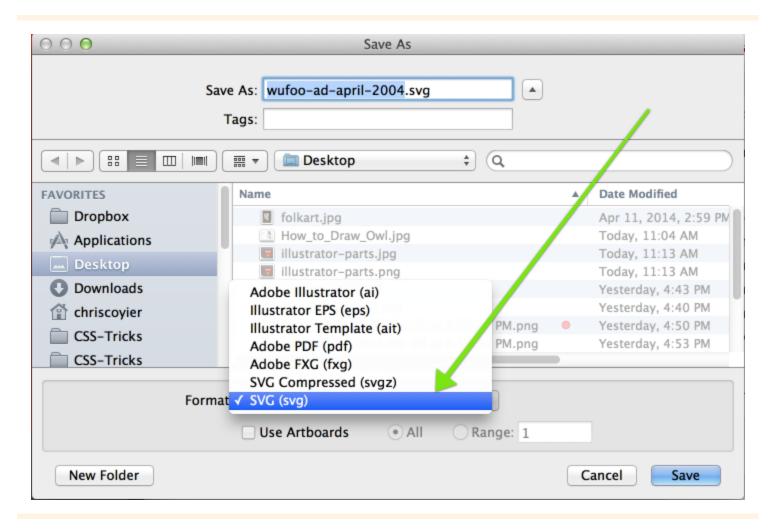


Notice how the logo and tagline text are outlines. That means they are just vector shapes and will render just perfectly as-is in the SVG, as <path> s. The text you see there "Fast." is left as text in Illustrator.

When I save this out from Illustrator, that will be left as a <text> element.

2. Save as SVG

Illustrator can save this directly as SVG:



You can open that SVG file in a code editor and see the SVG code:

```
\Theta \Theta \Theta
                                        wufoo-ad-april-2004.svg
     wufoo-ad-april-2004.svg ×
    <!-- Generator: Adobe Illustrator 16.0.4, SVG Export Plug-In . SVG Version: 6.00
     <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/</pre>
  4 <svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="
     http://www.w3.org/1999/xlink" x="0px" y="0px"
         width="400px" height="400px" viewBox="0 0 400 400" enable-background="new 0 0
         400 400" xml:space="preserve">
  6 <rect fill="#D03E27" width="400" height="400"/>
     <path fill="#F4F4F4" d="M60.858,129.642L30.163,60.579l20.81-4.05l9.402,26.</pre>
     003c2.916,8.116,6.081,17.52,8.701,24.691l0.215-0.041
       c-0.514-8.361-0.703-17.343-0.806-26.452l-0.565-27.5l20.703-4.029l9.933,27.
        012c2.895,8.01,5.3,15.223,7.852,23.188l0.215-0.042
        c-0.493-8.254-0.875-17.086-1.237-25.811L104.5,46.113l19.63-3.82l-3.737,75.764l-
        21.132,4.112L88.923,94.79
       c-2.304-6.121-4.585-12.691-7.221-21.084l-0.214,0.041c0.491,8.812,0.84,15.758,
  10
        0.806,22.44410.02,29.276L60.858,129.642z"/>
     <path fill="#F4F4F4" d="M148.403,62.895l-5.803,41.331c-1.549,11.</pre>
     036,1.356,17.843,9.039,18.921
 12
        c7.79,1.095,12.381-4.108,14.
        007-15.686l5.803-41.33l19.583,2.75l-5.651,40.249c-3.16,22.505-16.
        007,32.285-36.456,29.414
        c-20.016-2.811-28.977-15.211-25.68-38.689l5.575-39.708L148.403,62.895z"/>
 13
     <path fill="#F4F4F4" d="M186.189,51.906l47.135-3.928l1.333,16.</pre>
 14
     0021-27.431,2.28611.134,13.607125.799-2.15111.324,15.894
       l-25.799,2.149l2.323,27.867l-19.704,1.642L186.189,51.906z"/>
```

3. Clean Up the SVG, Give Shapes Classes

You might wanna run it through SVGO to optimize it and remove the DOCTYPE and stuff. But more importantly for this post, you'll want to give the the different shapes class names, that way we can select them in CSS and do stuff!

4. Insert the SVG

You can copy-and-paste that SVG right into the HTML where you want the ad. But that'll just slop up the template probably. In all likelihood you'll just do something like:

#5. Animate!

Now we have these shapes in the DOM that we can target and style like any other HTML element, let's do that.

Let's say we want to do this as a **10 second timeline**.

Words Fade In/Out First

The first thing we want to happen is the Fast. Smart. Formidable. thing. Each word will show for one second. So we'll make an animation where the word shows for 10% of the time:

```
@keyframes hideshow {
    0% { opacity: 1; }
    10% { opacity: 1; }
    15% { opacity: 0; }
    100% { opacity: 0; }
}
```

Then target that first word and have the animation last for 10 seconds (10% of that is 1 second):

```
.text-1 {
   animation: hideshow 10s ease infinite;
}
```

The next two letters will start out hidden (opacity: 0;) and then use the exact same animation, only delayed to start a bit later:

```
.text-2 {
   opacity: 0;
   animation: hideshow 10s 1.5s ease infinite;
}
.text-3 {
   opacity: 0;
   animation: hideshow 10s 3s ease infinite;
}
```

The extra 0.5s on each is to accommodate the fading out time period of the word before it.

Letter Pops

As soon as those letters are done animating, we'll have the letters in WUFOO do their jiggle jump thing, like this:



The trick here is that we'll make the animation only 5 seconds long, but run it once forwards and once backwards. That way it matches our 10 second timeline, is placed right in the middle where we want it, and we only need to scale in one direction, because when it reverses it will scale back.

Each letter has a slight bit of delay so they happen off-kilter a bit:

```
.wufoo-letter {
  animation: kaboom 5s ease alternate infinite;
 &:nth-child(2) {
    animation-delay: 0.1s;
  }
 &:nth-child(3) {
    animation-delay: 0.2s;
  }
 &:nth-child(4) {
    animation-delay: 0.3s;
  }
 &:nth-child(5) {
    animation-delay: 0.4s;
  }
@keyframes kaboom {
  90% {
    transform: scale(1.0);
  }
  100% {
    transform: scale(1.1);
  }
}
```

The above is in SCSS just for brevity, and does not include any prefixing (as you would need in production).

I feel like animation-delay is a property that would benefit from native randomization in CSS. Would be neat to see the letters be randomly delayed just a bit each time.

Dinosaur Last

As soon as the words are done, the dinosaur will peak its head up. Even though the dinosaur is made up of lots of <path> s, we can target them all together by targeting the <g> (group) tag that wraps them all.

Because using translate to animate position is better, we'll do that in the keyframes:

```
@keyframes popup {
    0% {
        transform: translateY(150px);
    }
    34% {
        transform: translateY(20px);
    }
    37% {
        transform: translateY(150px);
    }
    100% {
        transform: translateY(150px);
    }
}
```

We wanted this animation to "last" about 3 seconds. It actually runs in 10 second loops, but you'll only see it do stuff for 3 seconds. When the translateY(150px) is in effect, the dinosaur is moved so far below, you don't see anything. By 37% of that animation (around 3 seconds) you'll have seen it move slowly up, then quickly back down).

When we apply this animation, we'll make sure that:

The dinosaur is hidden at first

The animation is delayed so it starts right after the words are done with their fade in/out dance.

```
CSS
```

```
.trex {
   transform: translateY(150px);
   animation: popup 10s 6.5s ease infinite;
}
```

The dinosaur pops down right at the last second, which is when the word "Fast." pops back onto the screen (because all the animations are set to infinite, which re-runs them forever). That's a bit of fun synergy there.

6. Making it a responsive / clickable ad

One of the beautiful things about SVG is that it's scaleable to any size without losing quality. To make an inline SVG like this scale while maintaining its aspect ratio, we can use the ol' padded box technique.

```
.wufoo-ad-wrap {
  height: 0;
  padding-top: 100%;
  position: relative;
}
.wufoo-ad {
  position: absolute;
  top: 0;
  left: 0;
```

```
width: 100%;
height: 100%;
}
```

The idea here is that the "wrap" will always be a perfect square, relative to its width. Then we absolutely position the SVG within that perfect square, which will happily resize itself.

Since this is an ad (which of course should be clickable), rather than using a <div> for the wrap, you could use an , just make sure you set that to be display: block; .

Final thing, for reference.

I think there is a future for this kind of thing in display advertising, particularly because of the CSS control and easy, sharp resizing.

Related Posts ... powered by Jetpack!

The Jetpack WordPress plugin runs on this site, powering not just the related posts below, but the social sharing links above, security and backups, Markdown support, site search, the comment form, positing to social network connections, and more!



A Compendium of SVG Information

A huge pile of information about SVG. How to Use SVG These are overview articles covering lots of stuff relating to

SVG Animation and CSS Transforms: A Complicated Love Story

The following is a guest post by Jack Doyle, author of the GreenSock Animation Platform (GSAP). Jack has

Snap Animation States

There are many ways to make icons for a website. Inline SVG is scalable, easy to modify with CSS, and can even be

Comments



Steven Vachon

APRIL 17, 2014

Animating with CSS has always been an issue because it's percent based and not time based. This is one of the reasons why GSAP is so popular. SVG animation (SMIL) on the other hand does a pretty good job but has no support in IE at all. I'm in the process of writing smil2css to circumvent this problem. It essentially converts SMIL to CSS animations, much like your article is suggesting.



Amelia BR

APRIL 18, 2014

Sounds neat, but it won't help with current versions of IE — they don't support CSS animations applied to SVG, either.



Steven Vachon

APRIL 18, 2014

I haven't tested with IE11 yet as I don't have Win8, but it works fine in IE10 on Win7. There are a lot of browser issues with SVG right now. This one, for example still needs to make its way into a Chrome release.



Noufal Binu

APRIL 18, 2014

IE Vendor prefix not support?



Tom

APRIL 17, 2014

Never thought about using SVG as a code snippet; always just linked to them like I would a regular png or jpg . Now I can see how easy it is, I'm going to be using them like this all the time.

Great little tutorial, Chris. Thanks!



Steven Vachon

APRIL 17, 2014

You can also load them with background-image, which my favourite way.



Tom

APRIL 17, 2014

Good tip, Steven. Can you still target the class and id parameters in your CSS if you load them using background-image?



Steven Vachon

APRIL 17, 2014

Tom, I haven't tested it, but I'd imagine that when the SVG is loaded in as a style asset, it cannot access its parent document's stylesheet. It will definitely work, though, if you embed the relevant CSS into a <style> within the SVG file.

Shaw

APRIL 17, 2014

Good post! Thanks for the tips.

I was messing around with SVG / CSS animation last week and it took a while to figure out the syntax. Though I did manage to implement a variation of the popular "line drawing" animation: http://codepen.io/shshaw/pen/JDEsG

My main struggle was with how to handle the animation-delay s for individual elements in an elegant way, but by separating out the different stages of the animation (line drawing, stroke fade out, fill fade in), I was able to stagger the elements' animations the way I wanted.

Elaine

APRIL 18, 2014

Wow, Shaw, that is quite cool. Great job.

Michael

APRIL 17, 2014

Unfortunately only the three words animate on chrome mobile. The dinosaur and wufoo do not, or is this intended?

Timothy

APRIL 17, 2014

Nice work! I've been doing a lot of SVG animation lately, good to keep this technique in mind. Another advantage: you can tweak the animation for different screen sizes using media queries. Can't do that with just inline SVG animations, as far as I know.

Steven Vachon

APRIL 18, 2014

SVG animations (SMIL) can animate CSS properties, so you can animate with percent values. Media queries might get a bit messy for positioning, but would work fine for hiding/showing. It's give and take, really, because CSS can't do everything that SMIL can. One example would be animating along a path.

Peter Marosi

APRIL 18, 2014

Why U re not coloring with classes?

You can remove fill="FFFFFF" and make it like this

svg
.class
fill: #FFFFFF

Arafin Shaon

APRIL 18, 2014

This is the thing I've been trying to learn from last couple of months but its quite complex:

Martin

APRIL 18, 2014

I've used this illustrator technique to create svgs, which also spits out a doctyoe for the svg. I'm under the assumption that this is needed. Am I correct in my thinking?

Steven Vachon

APRIL 18, 2014

Sounds like you are trying to optimize the file size of your SVG file. If so, you might want to take the shortcut and just run it through SVGO.

Amelia BR

APRIL 18, 2014

Illustrator (and Inkscape) creates stand-alone SVG files that are fully XML compliant with doctypes and namespaces.

If you're only going to be using the image in the browser (and if your server is set to always give the correct MIME type for .svg files), you can usually skip the doctype. If you're going to be cutting & pasting your code into an html5 page (or importing it server-side via php), then you can also skip the namespace declarations.

Kris Bulman

APRIL 18, 2014

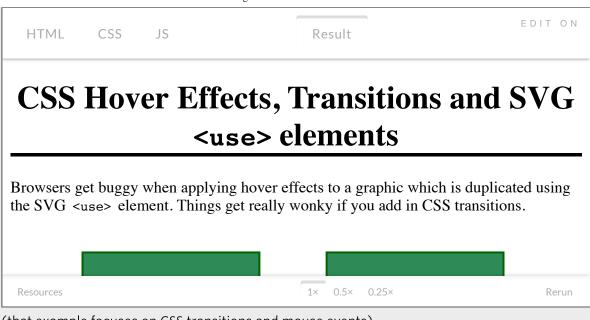
s/peak/peek/:P

Amelia BR

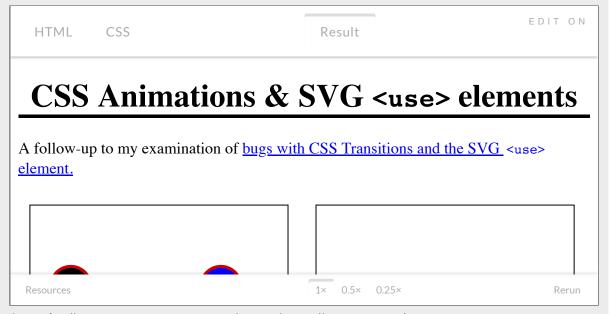
APRIL 18, 2014

Great step-by-step approach.

One warning for anyone trying to use CSS animations and transitions with SVG: avoid the use of the "element, or things become very buggy very fast!



(that example focuses on CSS transitions and mouse events)



(specifically on animations — not as buggy, but still inconsistent)

Amelia BR

APRIL 18, 2014

Urgh. The "in the warning should be **<use>.** Must remember to use the preview option.

P.S. Nice auto-embeds, Chris & co.!

Orangestar

APRIL 18, 2014

Expanding on what's already been said, SVGs seem to be supported poorly on mobile browsers. Most of the SVGs on this site are completely blank squares over here on Mobile Safari (iOS

6.1.6) and animations seem to be right out.

Arden

APRIL 22, 2014

SVG's, including animated ones (both JS and pure CSS) work like a charm on my nexus devices (5 & 7). It also seems to work fine on the xcode iDevice simulator, though of course, that doesn't say much;).

Mummy Ninja

APRIL 20, 2014

Very nice. Is there already a software that can easily generate the css code for you. Just select the animation you want, and poof! There you have it. Everything is neatly coded [HTML+CSS] after exporting/saving it.:D

Mohit Bumb

APRIL 21, 2014

Sorry dummy question but not good in designing is illustrator is required or we can do this in photoshop too?

Arden

APRIL 22, 2014

because of the nature of SVG (Vector Graphics) you'll need a program that can use and edit vectors straight out of the box; photoshop can't. I'm sure there are other options than illustrator out there though!

Andy

MAY 23, 2014

Photoshop: No. You could use Inkscape, which is free and open source.

Martin Lenngren

APRIL 22, 2014

This is great Chris, **again!** If your keyframe end (the 100% part) on the same position as it starts (the 0% part) you don't need to define the end at all, that's the default behaviour.

Bryan Hoffman

APRIL 22, 2014

I used this technique on a recent redesign of http://spigotdesign.com using a variety of transitions and transforms.

Works great on *most* modern browsers. In Firefox it's pretty ugly. Nice to see the Wufoo animates correctly in FF.

Andy

MAY 23, 2014

I'm posting this here because this technique might create a big mess with older webkit browsers. There is a weird bug in webkit with elements wrapped in -tags. Non-relative links break css-animation. My old Safari (5.1.10) even refuses to display the page completely, when it contains an inline svg animated by css, and wrapped in a link. I'm still exploring this, but it seems that a workaround might be to add the link via javascript. Then Chrome (35.0.1916.114) will do the animation, and old Safari will display the page, just without animation.

meghraj

JUNE 1, 2014

have you found workaround for this? if possible share with us.

Andy

MAY 23, 2014

Ooops, sorry! That should have read: ...with <svg> elements wrapped in <a>-tags...

lightnosugar

#JUNE 18, 2014

You can save out of Illustrator CC with classes and IDs in the SVG code.

CLASS: To give an object a named class, apply a Graphic Style with that name.

ID: To apply an ID, name a layer what you want your ID and place your object on it.

When you save your SVG file, go to "Advanced Options" and choose these settings:

profile: SVG1.1

CSS properties: Style Elements

decimal places: 3

Check the box for "Output fewer elements"

Leave the rest unchecked

When you open in a text editor, Illustrator will add inline styles which you can go in and delete, and instead use a CSS documnet to style.

xy75

#JUNE 27, 2014

define atomic reusable unstyled components group them into more complex components using "use" with styles and transformations clone complex components/whole svgs to achieve the independency

This comment thread is closed. If you have important information to share, please contact us.