Handout

CSS Animations
Lesson 1 - Step 5
Donovan Hutchinson

A question I'm often asked is why I choose CSS for animations when so much can be done with JavaScript?

One of the first reason is browser support. We have upcoming specifications from the W3C on the JavaScript Animation API which is very powerful and useful, and is going to bring a lot to what browsers can do. But while this is being developed, there's still a lot the browsers can do with CSS. Two properties, the `transition` and `animation`, allow us to introduce a lot of movement and animation to our UI without reaching for any external plugins or code. Browsers have progressed now to the point where transitions and animations are well supported.

Another consideration is the weight of our projects. Adding extra libraries adds to the page weight, which can make them slower and take longer for projects to render, particularly mobile browsers. Being able to put in animation or transitions in CSS is generally very lean, and adds minimal code to a site.

It's also simpler, at least, if you're coming at this from the perspective of someone not fluent in JavaScript. I tend to prefer the simpler choice whenever possible. I usually begin with CSS where I can. Having said that, it's not always so simple. Older browsers may not support animations terribly well, and there are tools such as GreenSock's GSAP, that do a terrific job of bringing advanced animation to older browsers.

Performance, which used to be a big deal when the choice was to use CSS or jQuery, is less of an issue now. Most JavaScript tools use hardware accelerated CSS properties when appropriate, and do a great job of performing well in browser.

In the end though, I recommend knowing what we can achieve with just CSS. When you hit the limits of what can be done, that's the time to think about bringing in more advanced tools and libraries. There is a lot we can do with just a bit of CSS. So let's start with transitions.