What DOM animation technique should you use?

Hide tutorial navigation

You've now seen three techniques that let you animate parts of your webpage: `window.setInterval`/`setTimeout`, `window.requestAnimationFrame`, and CSS animations/transitions.

When you're considering what technique to use, you should consider these questions:

- Can it actually do what I want to do?
- How well does this technique perform? (Does it slow down the browser/computer?)
- How accurate does my timing need to be?
- Does it work in all the browsers that I want my webpage to work in?

When making webpages, we're very concerned about performance, so we like to consider that as the most important criteria. However, browser performance does change when new browsers come out, and mobile browsers can perform very differently from desktop browsers, so the technique that performs best today may not be the technique that performs best tomorrow. Currently, CSS animations/transitions are the most performant, then `requestAnimationFrame`, then `setInterval`.

However, CSS can't do everything. For example, to draw pixels and shapes in a `<canvas>` tag, you have to call functions like `fillRect()`, you can't use CSS. You would need to use `requestAnimationFrame` or `setInterval` to call those functions periodicially instead. In fact, that's what we do here on Khan Academy, in our ProcessingJS environment. ProcessingJS is a JS library that writes out to a `<canvas>` tag, and if you define a `draw()` function in your code, then ProcessingJS uses `setInterval` to call that `draw()` function repeatedly based on the `frameRate`.

Sometimes, you want to call JavaScript functions periodically, but not because you want to animate something on the page. You might be polling a server for updates, like Twitter does when it updates its realtime feed. In that case, you can just use `setInterval`, and it doesn't matter that the timing isn't accurate, because you're only calling it every minute or so. That's what we do on Khan Academy on the help requests page, to continuously check for new help requests every 2 minutes.

Of course, you should keep browser compatibility in mind. If you're writing code that needs to work in IE9, then you can't use `requestAnimationFrame` or CSS animations. You need to use a combination of techniques that work across the browsers, or find a library that does that for you, like Velocity.js.

There is a lot more to learn about all of these techniques. Follow these links to learn more:

- CreativeJS: requestAnimationFrame
- CSS Tricks: Using requestAnimationFrame
- MDN: Guide to using CSS animations
- TutsPlus: Beginners intro to CSS animation
- The art of web: CSS animation

- CSS Animation: An interactive guide

Bonus for Doctor Who fans: one of my favorite examples of the power of CSS3 animations is this animated tardis.