Meet requestAnimationFrame

Making your animations run smoothly depends on a lot of factors. It depends on what else is going in your page. It depends on what other animations might be running in parallel. It depends on whether the user is interacting with the page by clicking or typing. It depends on what browser you are using and when it decides to repaint or update what is shown on the screen.

Traditionally, you may have used a function like `setInterval` or its funnier cousin `setTimeOut` to power your animation loop. The problem with these two functions is simple. They don't understand the subtleties of working with the browser and getting things to paint at the right time. They have no awareness of what is going on in the rest of the page. These qualities made them very inefficient when it came to powering animations because they often request a repaint/update that your browser simply isn't ready to do. You would often end up with skipped frames and other horrible side effects.

If `setInterval` and `setTimeOut` went out on a date, they would probably be heckled quite a bit. Despite how bad `setInterval` and `setTimeOut` were for working with animations, you had no real alternatives. You had to use them. This lack of better alternatives left animations created via JavaScript looking a little less polished when compared to animations created in CSS or via a dedicated runtime like Flash.

Fortunately, things changed. Given how important animations are for creating great looking applications and sites, the major browser vendors (starting with Mozilla) decided to address this problem in an elegant way. Instead of burdening you with having to deal with all of the issues related to creating a smooth animation in JavaScript using `setInterval` or `setTimeOut`, they created a function called `requestAnimationFrame` that handles all of those issues for you.

What makes `requestAnimationFrame` so awesome is that it doesn't force the

browser to do a repaint that may never happen. Instead, it asks the browser nicely to call your animation loop when the browser decides it is time to redraw the screen. This results in no wasted work by your code on screen updates that never happen. Skipped frames are a thing of the past. Best of all, because `requestAnimationFrame` is designed for animations, your browser optimizes the performance to ensure your animations run smoothly depending on how much system resources you have available, whether you are running on battery or not, whether you switch away to a different tab, and so on.

Words haven't been invented in the English language to describe how awesome the `requestAnimationFrame` function is.