

# Using requestAnimationFrame

 [css-tricks.com/using-requestanimationframe/](https://css-tricks.com/using-requestanimationframe/)

Published June 8, 2013 by Chris Coyier

There used to be just one way to do a timed loop in JavaScript: `setInterval()`. If you needed to repeat something pretty fast (but not as-fast-as-absolutely-possible like a `for` loop), you'd use that. For the purposes of animation, the goal is sixty "frames" per second to appear smooth, so you'd run a loop like this:

```
setInterval(function() {  
  // animate something  
}, 1000/60);
```

There is a better alternative to this now. [Paul Irish introduced](#) `requestAnimationFrame` over two years ago. I don't have a whole lot to add to it, I just had never actually used it before and now I have so I thought I'd help spread the word and write about its basic usage.

## Why better?

As Paul explained:

- The browser can optimize it, so animations will be smoother
- Animations in inactive tabs will stop, allowing the CPU to chill
- More battery-friendly

## The Simplest Possible Example

```
function repeatOften() {  
  // Do whatever  
  requestAnimationFrame(repeatOften);  
}  
requestAnimationFrame(repeatOften);
```

Call it once to kick it off, and your function recursively calls itself.

## Start and Stop

`requestAnimationFrame` returns an ID you can use to cancel it, just like `setTimeout` or `setInterval` does. jQuery used here only to demonstrate a simple animation and bind events.

```
var globalID;  
function repeatOften() {  
  $("  globalID = requestAnimationFrame(repeatOften);  
}  
$("#start").on("click", function() {  
  globalID = requestAnimationFrame(repeatOften);  
});  
$("#stop").on("click", function() {  
  cancelAnimationFrame(globalID);  
});
```

Example of this:

[Check out this Pen!](#)

## Browser Support

See the [Can I Use...](#) tables.

The only notable problems are IE 9-, iOS 5-, and Android. But not actually a problem, because:

## Polyfill

Like many fancy web features, it's nice to use it when available and fallback to something that works when you can't. Probably best just to [refer to this Gist](#). Literally just include that chunk anywhere before you use `requestAnimationFrame` or `cancelAnimationFrame`.

Using this, you'll be able to use `requestAnimationFrame` in literally any browser.

## Slightly More Complex Example

I learned about this while making a dumb little demo to learn canvas better:

[Check out this Pen!](#)

What would actually be more complex is several animations running at once using this (that still falls back OK). Feel free to link some of that goodness up in the comments if you know of examples.