A Drip of JavaScript

Ditching jQuery with `querySelectorAll`

Originally published in the [A Drip of JavaScript newsletter](#).

For many developers, jQuery serves as their first introduction to JavaScript. And jQuery does a great job of easing the learning curve by hiding away browser inconsistencies and using an intuitive chaining syntax. But probably the most distinctive feature of jQuery is its use of CSS selectors to choose which elements to manipulate.

For example, consider an HTML page that looks like this:

```html
<!doctype html>
<html>
    <head>
        <title>Programming Languages</title>
    </head>
    <body>
        <h1>Programming Languages</h1>

        <ul>
            <li>JavaScript</li>
            <li>CoffeeScript</li>
            <li>Ruby</li>
            <li>Python</li>
        </ul>

        <script src="example.js"></script>
    </body>
</html>
```

If you want to use jQuery to add a class to the first and last `li`, that's as simple as:

```
$("li:first-child, li:last-child").addClass("list-end");
```

What many people don't realize is that modern web browsers actually support a native DOM method that can use CSS selectors in exactly the same way. Let's see how that works.

```
var selector = "li:first-child, li:last-child";
var listEnds = document.querySelectorAll(selector);

var listEndsArr = [].slice.call(listEnds);

listEndsArr.forEach(function (el) {
    el.className += "list-end";
});
```

As you can see, `querySelectorAll` will return a list of matching elements. However, the list that it returns is a `NodeList`, one of those "array-like" objects that isn't really an array. That means that we can't directly use some array methods like `forEach`.

To get around that problem, we use `slice.call` to create a real array of the matching elements. And once we have a real array we use `forEach` to add the class to each matching element.

There are some issues to keep in mind when using `querySelectorAll`. First, the CSS selectors you can use will be limited by the selectors that the browser supports. Second, while it enjoys good support among modern browsers, `querySelectorAll` is not available in IE7 and below.

Of course, `querySelectorAll` isn't a full replacement for jQuery, but it does get you one step closer to jQuery independence.
