



Creating arrays from node lists

Using [querySelectorAll](#) to get one or more items and then process them as an array of items is not as straight forward as I initially thought it would be, at least in the browsers I use for testing

Part of the problem is that `querySelectorAll` returns a [node list](#) and some of them will not work with node lists and will require the conversion to an array before we do anything.

We'll look at three ways to work with node lists as arrays. The last one works with node lists directly. The examples will each take all the paragraphs on a page and then print the content of each paragraph to the console. They will also indicate what version of the ECMAScript specification the feature was first introduced in.

Convert to array with spread syntax

The first way I found to convert a node list to an array is to use [spread syntax](#) to include the `rawData` iterable in the array.

I always do this as a two-step process to make sure I do it right. The code for this version of the example looks like this.

```
const rawData = document.querySelectorAll("p");
const myParagraphs = [...rawData];

array.forEach
myParagraphs.forEach((paragraph) => {
  console.log(paragraph.innerText);
});
```

Spread syntax is available since ES2015/ES6. It should be safe to use.

Convert to array with array.from

Another way to create an array from our rawData node list is to use [array.from](#) to generate the array.

The code for this example is almost identical to the spread syntax example; the only difference is how we generate the myParagraphs array.

```
const rawData = document.querySelectorAll("p");
const myParagraphs = array.from(rawData);

//array.forEach
myParagraphs.forEach((paragraph) => {
  console.log(paragraph.innerText);
});
```

array.from is also available since ES2015/ES6.

nodeList.forEach

The biggest surprise when I was researching this post is that the nodeList interface also has a [forEach](#) method.

This means that, in the end, we don't need to convert the node list to an array if all we want to do is run forEach on it.

This example uses nodeList.forEach to run the same example. The code is less complex since we don't have to convert the node list to an array.

```
const rawData = document.querySelectorAll("p");

// NodeList.forEach
rawData.forEach((paragraph) => {
  console.log(paragraph.innerText);
});
```

This method was introduced in ES2016/ES7 so it should be safe to use.

As usual with code, test it on your target browsers to make sure they work.