



Footnotes on web pages

As I was working on my last post about paged media I discovered one thing we can do to make footnotes for the web a little more user friendly. Paged Media has a way to hide the text of the footnote from being displayed on the page... wouldn't it be nice to do that with web content as well without having to change the document we're working with?

CSS on its own will take you part of the way there... so let's explore how can Javascript help. What we want to accomplish:

1. Hide all the elements with class footnotes
2. Add a number corresponding to the number of the footnote
 1. If possible make it a hyperlink
3. Create an ordered list
4. Create a link for each footnote pointing to the footnote number

First we define the code we'll use to generate the footnotes. By using a span with an associated class

```
<span class='footnote'>Another footnote</span>
```

On the Javascript side we capture all the footnotes using [querySelectorAll](#) to get a list of all the elements that match the `.footnote` selector.

Even though the `querySelectorAll` returns a [nodeList](#), not an array. While we can still use the `forEach` method, it has none of the other array methods. If we need other array methods we can covert the list into an array using [destructuring assignments](#) or [array.from](#) to create an array you can operate on. To be on the safe side, I used destructuring to convert the list into an array.

```
const footnotes = document.querySelectorAll('.footnote');  
fnArray = [...footnotes];
```

Now that we have the array, the first step is to generate the footnote marker and link. We use `(i+1)` instead of just `i` because arrays start at 0 instead of 1.

Once we have the footnote marker we insert it into the document adjacent to

the footnote text; only then we can remove the footnote text from the document, otherwise we would have nowhere to insert the marker.

```
fnArray.forEach(function(elem, i) {
  let reference = document.createElement('sup');
  reference.setAttribute('id', i + 1);

  'id'// Create the link and attributeset link = document.createElement('a');
  link.setAttribute('href', '#fn' + (i + 1));
  link.innerText = i + 1;

  // A'a'// Append the link to the reference element
  reference.appendChild(link);

  // Insert the linked reference to the page after
  // the footnote text.insertAdjacentElement('afterend', reference);

  // Hide'afterend'// Hide the footnote text
  elem.style.display = 'none';
});
```

With the markers in place and the footnote text removed we can build the list of footnotes at the end of the document.

We first create an hr element to display a horizontal rule to distinguish the footnote list from the body of the text.

We then open an ol element for the numbered/ordered list that will contain the footnotes.

```
// Create hr element and append it to the page
const hr = document.createElement('hr');
document.documentElement.appendChild(hr'hr'// Create ol element and append
const ol = document.createElement('ol');
document.documentElement.appendChild(ol);
```

For each footnote:

- We create the `li` element that will contain the footnote
- We create an `a` element that will link to the footnote marker
 - We set an `href` attribute pointing to the footnote marker ID
 - Set the content of the link to be the content of the footnote element we removed
- We append the link to the list item element
- We append the list item to the `ol` object

```
// For each footnote in the array
fnArray.forEach(function(elem, i) {
  // Create the list item
  let item = document.createElement('li');
  'li'// Create the link, set href attribute and the textet link = document
  link.setAttribute('href', '#' + (i + 1));
  link.innerText = elem.textContent;

  // A'a'// Append the link to the list item
  item.appendChild(link);
  // Append the list item to the list
  ol.appendChild(item);
});
```

One thing we need to do is research the performance implications of using this script inline versus calling it from an async script tag.

A working demo is at this [Codepen](#)