

JavaScript: how to slice DOM elements

A DOM collection (such as an `HTMLNodeList`) works much like an array but, unfortunately, doesn't have the standard JavaScript array methods. For that reason, if we want to slice a set of DOM elements we have to call the corresponding array method directly from its prototype object. Let's see how.

We have the following DOM structure:

```
<ol id="test">
  <li>A</li>
  <li>B</li>
  <li>C</li>
  <li>D</li>
  <li>E</li>
  <li>F</li>
  <li>G</li>
  <li>H</li>
</ol>
```

Implementing a `slice()` utility function requires the use of the `call()` method provided by the `Function` object:

```
var slice = function(elements, start, end) {
  var sliced = Array.prototype.slice.call(elements,
start, end);
  return sliced;
}
```

The first parameter is the DOM set, followed by the two offsets that will be used to slice the collection. Here's a practical example:

```
var elems =
document.getElementById('test').getElementsByTagName('li
');
var html = '<ol>';
var slices = slice(elems, 2, 6),
    len = slices.length,
    i = 0;
for(i; i < len; i++) {
    html += '<li>' + slices[i].firstChild.nodeValue +
'</li>';
}

html += '</ol>';

var output = document.createElement('div');
output.innerHTML = html;
document.body.appendChild(output);
```

You can see the demo below.

Demo

[Live demo](#)