COMP 2015

Introduction to JavaScript and jQuery

Contents

DOM Scripting

createElement()

Thus far we have seen how to manipulate HTML elements on the page, however we can also use JavaScript to *create* elements through what is called DOM Scripting.

In order to add an element to the page, we must first *create* the element and then *append* it to the body.

example:

```
var element = document.createElement('p');

document.body.appendChild(element);
```

We can create any type of HTML structure that we would otherwise write in regular HTML. For example, we can *nest* elements before appending them to the body.

example:

```
var container = document.createElement('div');
var paragraph = document.createElement('p');
var span = document.createElement('span');

container.appendChild(paragraph);
paragraph.appendChild(span);
document.body.appendChild(container);
```

which results in the following HTML:

```
<body>
    <div>
        <p><span></span></p>
    </div>
</body>
```

In these examples, we have created what are referred to as Element Nodes (i.e. HTML elements). Just as we add text between, for example, an opening and closing paragraph, we can use DOM Scripting to add text to our page.

To add text to the page, we must first create what is referred to as a Text Node, and then append that Text Node to an Element Node.

createTextNode()

example:

```
var paragraph = document.createElement('p');
var text = document.createTextNode('Hello World');

paragraph.appendChild(text);
```

which results in the following HTML:

```
<p>Hello World</p>
```

Manipulating The DOM Tree

The DOM can be thought of as a tree structure, similar to a family tree, with parents, children and siblings. In the following example, the HTML tag has two children: head and body; the body tag has two children: an image and a paragraph; and the image and paragraph are siblings because they both have the same parent (the body).

```
<html>
    <head>      <!-- child of HTML and sibling of body -->
    </head>
    <body>      <!-- child of HTML, sibling of head and parent to img and p -->

        <img id="im1" src="">  <!-- child of body and sibling of p -->
        <p></p>       <!-- child of body and sibling of img -->

    </body>
</html>
```

Using DOM scripting, we can traverse this tree to find siblings, parents and children of elements. We can even manipulate the tree and move elements around.

example:

```
    var image = document.getElementById('im1');

    if (image.nextElementSibling != null) {
        image.parentNode.insertBefore(image.nextElementSibling, image);
    }
```

The above code does several new things:

- uses the `.nextElementSibling` property to see if the image element has a sibling
- if it does, the code uses the `.parentNode` property to swap the image with its sibling using the `.insertBefore()` method

Two important things to note!

First, only parents can rearrange their children, the children cannot rearrange themselves. That is why we call image.parentNode – to grab the parent (body) element. We can then call .insertBefore on the parent to swap the elements.

Second, there is no `insertAfter()` method, so (although this may be counterintuitive) we must use `insertBefore()` to put the next sibling *before* the image.

After running this script the previous example now looks like this:

```
<html>
    <head>      <!-- child of HTML and sibling of body -->
    </head>
    <body>      <!-- child of HTML, sibling of head and parent to img and p -->

        <p></p>       <!-- child of body and sibling of img -->
        <img id="im1" src="">  <!-- child of body and sibling of p -->

    </body>
</html>
```

DOM Manipulation Examples

Given the following HTML structure, here are a few examples of how we can manipulate the DOM tree.

```
<html>
    <head>
    </head>
    <body>
        <p id="p1">
            <img id="image1" src="" alt="">
            <img id="image2" src="" alt="">
        </p>
        <p id="p2">
            <img id="image3" src="" alt="">
            <img id="image4" src="" alt="">
        </p>
    </body>
</html>
```

Example 1: Move #image4 into #p1

```
    var im4 = document.getElementById('image4');

    if (im4.parentNode.previousElementSibling != null) {
        im4.parentNode.previousElementSibling.appendChild(im4)
    }
```

Example 2: Move #image2 into #p2, placing it before #image3

```
    var im2 = document.getElementById('image2');

    if (im2.parentNode.nextElementSibling != null) {
        var parentSibling = im2.parentNode.nextElementSibling;
        parentSibling.insertBefore(im2, parentSibling.firstElementChild)
    }
```

Example 3: Swap #p2 with #p1

```
    var paragraph2 = document.getElementById('p2');

    if (paragraph2.previousElementSibling != null) {
        paragraph2.parentNode.insertBefore(paragraph2, paragraph2.previousElementSibling);
    }
```

Example 4: Swap #p2 with #p1 – alternate version

```
    var paragraph2 = document.getElementById('p2');
    var parent = paragraph2.parentNode;

    parent.insertBefore(paragraph2, parent.firstElementChild);
```

Next Week

Lab 5 and Assignment 2 due before the beginning of Lesson 6.

Download it from our course section of COMP2015 on D2L http://learn.bcit.ca (COMP2015 > Content > Lesson 5)