Web Technologies – Exercises: Week 4

I hope you've started the JavaScript reading … referring to those chapters and the full set of lecture slides should make this exercise easy!

Outcomes

By the end of the fourth week, you should be able to:

1. Work with JavaScript dates and **`document.write`**.
2. Tell the differences in **`navigator`** property values between browsers.

Task 1: Do the quiz

Find it in the 'Exercises' folder on StudySpace. As usual, your best mark before the end of the next lab session contributes to the 10% "quiz mark" – aim for 100% :-)

Task 2: Examine the various **`Date()`** object methods:

1) Make sure you're familiar with the slides from Week 4 concerning the **`Date()`** methods.
2) Create a new XHTML document in a file called `week4date.htm` with the **`title`** '**`Web Technologies: Week 4 – date object methods`**'.
   - You could use the new empty-JS file that on StudySpace in the exercises section as a template.
3) Add a **`<script>`** block to the **`<head>`**, including the appropriate **`type`** attribute.
4) Declare a new variable called **`today`** containing today's **`Date()`** (or in Java-style O-O speak "a new instance of the **`Date`** object"). **`today`** will be a global variable as it's declared at the top level (with or without the **`var`** keyword).
5) Use the JavaScript **`alert`** method to pop up a dialog box that displays the variable when the page loads.
   - If you have not done so before, **test your page in a browser now!** This is (sort-of) what you should see (except with the current date/time) in a recent version of Firefox:

6) Next create a **table** with 6 rows and two columns. Like the table below, in each row display the **name** of the *method* in the first column and then the **JavaScript output** from each of the methods (in the table below) applied to the **today** date object. A typical fragment of HTML and JavaScript code *for each row* might be:

```
<tr>
   <td>toDateString</td>
   <td
        <script type="text/javascript">
             document.write( today.toDateString() );
        </script>
   </td>
</tr>
```

This uses the **write** method of the **document** object to display the output from the **toDateString** method of the date object **today**.

- Your table structure should be something like the following:

toDateString()	(some date text)
toTimeString()	(some time text)
toLocaleString()	(some date/time text)
toUTCString()	(some more date/time)
toString()	(some more date/time text?)
valueOf()	(a number)

  - Now you can see why **get{Hours,Minutes,Seconds,Date,Month,FullYear}()** methods are needed when displaying dates within pages! Developers also need to rely on the **getUTC** variants to cope with page views between time zones.

7) Add a level 1 heading above the table to describe what you've just displayed and add another row to the table with appropriate column headings using table heading tags.

8) Add an **<address>** block below and add the usual three lines of text to it listing (i) your k-number, (ii) your name and (iii) your course title and level.

9) Use CSS in a **<style>** block in the **<head>** to right-align the **<address>** block.

10) Make sure your XHTML is *valid*.

PTO...

11) Add more *style* to the page, including at least (but not limited to) the following:

- A 6 pixel border on the table and 2 pixel borders on the cells, collapsed together.
- Some space **inside** the cells (between the text and the border).
- A background and text colour to the column headings in the table.
- A mono-spaced font like "Courier"or "Consolas" for the names of the `Date()` methods in the first column of the table but not the second column.
  - **Hint**: You can do this easily with a CSS *class* or the *adjacent CSS selector* "+".
    If you're confused there's a picture of the basic page (with old text & different styles) on StudySpace but ***do not copy the text from mine***.
    Do add your own styles in addition to those above!

12) Make sure it's still valid XHTML & CSS.

Task 3: Examine the various navigator object properties

In this exercise you're aiming to produce a page like the picture on StudySpace. We've talked briefly about the so-called "Browser Object Model". Now investigate for yourselves the properties of the `window.navigator` object (that you can safely refer to as `navigator` since `window` is the top level object.)

1) Create a new XHTML document called `week4nav.html` with title
   'Web Technologies: Week 4 – navigator object properties'.
2) Add a heading element to describe the page.
3) Create an XHTML `<table>` with one header row and rows for each of the following read-only properties of the `navigator` object (so the cells in the right-hand column each contain a `<script>` block and a `document.write` JS statement that prints out the value of these properties):

`navigator` property name	Value
appCodeName	(text written by JavaScript `document.write` statements describing aspects of the browser in separate cells)
appName	
appVersion	
cookieEnabled	
userAgent	

4) Add some CSS formatting to the page (have some fun … but don't use *blink*!)
   a) *E.g.* to make the rows stand out, use *one* CSS class to apply different coloured backgrounds to *alternate* content rows.
   b) Use another background colour for the headers.
      - You could use a **<style>** block in the **<head>** for this *or* an external style sheet (call it `week4nav.css`).
   c) Centre the table but not the heading.

NB Don't just copy the style in my picture -- yours should be different!

5) Make sure your document is valid XHTML and CSS and add your name, k-number and course title to the page on separate lines at the bottom-right of the page.
6) Open `week4nav.html` in at least 3 different browsers (*e.g.* Internet Explorer, Opera and Firefox) and appreciate the differences between them.
7) Copy the text from the three browsers into a new web page `week4browsers.html` containing one table like `week4nav.html`, but with 4 columns, where the 3 data columns are labelled with the browser name and contain the text from each.
8) Make sure it's also valid XHTML & CSS.

This concludes the fourth week's exercises … *before next week* make sure you have done all of today's mandatory tasks *and the reading…*

Appendix: HTML **<table>**

If you don't know how to build a *table* in HTML then read on and/or look here! (Index DOT Html is a good site to know for basics in addition to W3Schools…)

An XHT**ML** table like the one in "Task 3" would be marked-up (**ML** ="markup language") using XHTML tags as follows:

```
<table>
```

Indicates the start of a table

```
 <tr>
```

Start of a table row -- remember white space is ignored/treated as one space in HTML so you can make the code look pretty with sensible indenting

```
  <th scope="row">
    toDateString()
  </th>
```

A table heading cell opened, some *text*, and the closing tag

(For proper semantic markup, where you use appropriate tags to present information, any table header rows should have cells marked-up with **<th>** instead of **<td>**. **scope** defaults to "**col**").

```
  <td>
    <script
    type="text/javascript">
      //some JavaScript
      document.write(…);
    </script>
  </td>
```

Another cell within the same row which contains a <script> block -- **document.write** will write into the cell in the browser.

```
 </tr>
```

Always mark the end of the row!

```
 <tr><th scope="row">…</th>
     <td>…</td></tr>
```

Add as many other rows as you need, with as many cells as you need… (XHTML tables need not be rectangular 'arrays' they can be ragged…)

```
</table>
```

Finally, close the table!