CSCU9A2 Spring 2017

Web Design Practical 3: JavaScript

AIM: A BRIEF INTRODUCTION TO CREATING INTERACTIVE WEB PAGES WITH JAVASCRIPT.

This sheet contains <u>two</u> checkpoints. Deadline: Thursday 30th March

FURTHER DOCUMENTATION

There is a short summary of JavaScript on the Web Design page in the References section of the CSCU9A2 website. For more detail, consult Chapter 15 of *Creating a Website: the Missing Manual*. This practical is based in part on examples taken from this book. See also the W3Schools tutorial on JavaScript at `http://www.w3schools.com/js/`.

PART ONE: JAVASCRIPT BASICS

The best way to learn JavaScript is from examples. Copy the folder:

```
My Computer\Groups (V:)\CSCU9A2\JS\
```

into your Web folder. Navigate into the subfolder Part One. Open the document `MessageBox.htm`, first in a web browser and then in TextPad. This is a very simple example showing how to pop up an alert box. Next, examine the document `CurrentDate.htm`, first in a browser and then in TextPad. This example shows you how to use the Date object to find out the current date. It also shows you how to use `document.write()` to alter the contents of the web page.

Now for your first task. Make a copy of `CurrentDate.htm`, calling it `Birthday.htm`. Alter this so that on your birthday, it pops up an alert box wishing you a Happy Birthday. (For testing purposes you can pretend that today is your birthday.) Here are some hints to help:

You can find out the current month and date by using the `today.getMonth()` and `today.getDate()` methods. Note that `getMonth()` returns a value between 0 and 11.

You will need to use an `if` statement. The good news is that JavaScript's syntax for `if` statements and logical conditions is virtually identical to that of Java. If you have any difficulty figuring it out, ask for help.

Next we will look at using functions in JavaScript. These are similar to methods in Java. As in Java, functions are usually passed some values (called *arguments* or *parameters*), operate on these values, and may return some new values (usually called *return values*). Functions are declared within scripts in the head of the HTML page (that is between the `<head>` and `</head>` tags) and may be called from scripts anywhere in the page.

Open the file `Functions.htm`, first in a browser and then inside TextPad so that you can examine the code. In this example there are two functions, `ShowAlertBox` and `MultiplyNumbers`, both of which take parameters. The example shows how these functions can be invoked by scripts elsewhere in the body of the page. If there is anything you do not understand, ask for help.

Now open the document `MoreFunctions.html`. This file illustrates one of the commonest use of functions in JavaScript: to cause something to happen when a button is pressed. After examining this document in a browser, open it in TextPad and Save a copy of it as `MyFunctions.html`.

In this example, pressing the first button calls the function `newText()`, (a function which is not passed any parameters, hence the empty brackets) which changes the text in the adjacent text form. This function doesn't return any values either: it simply alters something on the page.

Alter this part of the program (i.e. the function `newText()`) so that

- When the button is pressed, the text "DO NOT PRESS THIS BUTTON AGAIN!" is displayed
- When the button is pressed for a second time, and subsequently, the text "I told you not to do that!" is displayed.

Hints: you will need to have some way of distinguishing whether the button is being pressed for the first time or not. One good way of doing this is to have a variable which is initialised to 0 when it is declared

The code:

```
<script>
   var whichTime = 0 ;
</script>
```

will declare such a variable, but note that the variable needs to be declared before the function is encountered, not inside the function. Why?

You can the test this variable, and if it is 0 (`whichtime == 0`) the first text should be displayed, and the variable incremented. If it is non-zero (`whichtime !=0`) the second text is displayed.

As you will see from the rest of the example, functions can be used to implement a counter, counting how often a button has been pressed. Look at the code of the function `updateCount()`. Alter this code so that instead of counting upwards, pressing the button causes the count to be decreased, starting from 10. When the value is 0, display the word "Ignition" instead of the number. When the button is pressed again, display the word "Lift-off".

CHECKPOINT [JAVASCRIPT1]

Show the demonstrators the documents Birthday.html and MyFunctions.html, displaying the behaviour described above.

PART TWO: DYNAMIC IMAGES AND EVENTS

Navigate into your folder `H:\web\JS\Part Two`. This folder contains some examples of how JavaScript can be used to make web pages respond to "events" such as the mouse being clicked or rolled over a page element. The response to such an event is often to cause an image to change (dynamic images).

Start by looking at the file `Events.htm`, first in a browser and then in TextPad. This is a simple example showing how to cause something to happen when the mouse is rolled over an element in the page, in this case, a link. The element is given an attribute called `onmouseover` whose value is the action to be performed when the mouse is moved over that element. In this case the action is to pop up an alert box.

To see how this can be used to create a dynamic image, examine the document `ImageRollover.htm`. Look at it in the browser first to see what it does, then open it in TextPad. This document contains a function, `ChangeImage`, which changes an image (identified by its id) in the document. The function is called whenever the mouse is moved onto the image (the `onmouseover` event) and again when the mouse leaves the image (the `onmouseout` event). This causes the image source (the picture that is actually seen) to be swapped between `pic1.gif` and `pic2.gif`.

The third example, `CollapsibleText.htm`, shows how to make a web page containing text that can be expanded or collapsed on a mouse-click. This is achieved by using a function `ToggleVisibility` which alters the `display` attribute of a division in the page containing the text to be expanded/collapsed. The function is called when the mouse is clicked within an enclosing division which shows some summary text. This is a slightly more complex example than the others you have seen. To check your understanding, see if you can alter it so that the hidden text is expanded if the mouse is rolled over the summary text, and then disappears again when the mouse moves out of the summary text.

For your second checkpoint, use your imagination and create a page which uses all of the JavaScript you have met in this sheet. Your page must use variables and functions, must respond to mouse events of at least two different types and must contain some dynamic images. The content of the page is up to you, but be warned that if the demonstrators are not sufficiently impressed they may ask you to do some more work. Have fun!

CHECKPOINT [JAVASCRIPT 2]

Show the demonstrators the web page you created to show off your JavaScript skills. The page must use variables and functions, must respond to at least two different types of mouse events, and must contain some dynamic images.