Lab – Additional JavaScript Exercises

Useful resource - W3schools: http://www.w3schools.com

Exercise 1- Rollover Effects

1. Create a new html file called rollover.html
2. Create two external links within the <body> element
    a. Add an external link to w3schools.com website
    b. Add another link to html5test.com website
3. Add <style> elements in the <head> element
4. Add a rollover style definition for a visited link in the <style> element
5. Add a rollover style definition for a hovered link
6. Preview the file in a browser and test the rollover effects work
7. Change the rollover styles and test the changes are reflected in the browser window
8. Add a few images to the html file (sample ones within the images folder on Moodle can be downloaded if required)
9. Add an 'onmouseover' event to change the source of the image
10. Add an 'onmouseout' event to change the source back to the original file
11. Preview the file in a browser and test these events work
12. Save the changes to the file

Exercise 2 - For Loop

1. Create a new html file called for_loop.html
2. Place <script> elements in the <head> element
3. Use a 'For loop' inside the <script> elements which displays a number from 1 to 10 (inclusive) and its cube (cube of 2 is calculated by 2*2*2=8)
   Notes:
   Declare a variable
   Add the for loop syntax ie for (stmt1;stmt2;stmt3){code}
4. Use document.write to display the outcome in the browser window similar to see Figure 1.
5. Preview the file in a browser and check the outcome
6. Try some of your own calculations
7. Save the changes to the file

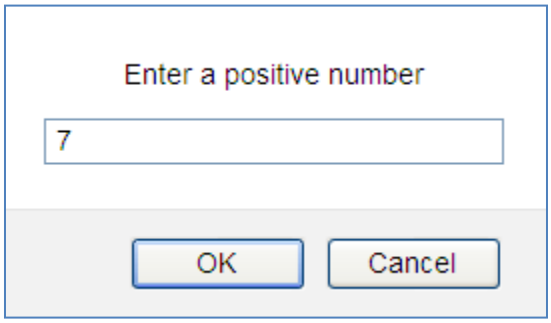
Figure 1

Number 1 cubed equals 1
Number 2 cubed equals 8
Number 3 cubed equals 27
Number 4 cubed equals 64
Number 5 cubed equals 125
Number 6 cubed equals 216
Number 7 cubed equals 343
Number 8 cubed equals 512
Number 9 cubed equals 729
Number 10 cubed equals 1000

Exercise 3 – Function

1. Open for_loop.html (from Exercise 2)
2. Save the file as function.html
3. Create a function called cube() in the <script> elements
4. Move the JS code inside this function
5. Add a <button> element to the <body> section of the page
6. Add an onclick event handler to this button which will call the function cube() whenever it is clicked
7. Preview the file in a browser and check the outcome. A button should be displayed, which when clicked, displays the similar information to Figure 1

Exercise 4 – While Loop



1. Create a new html file called while_loop.html
2. Place <script> elements in the <head> element.
3. Use a 'While loop' inside the <script> elements which displays a positive number entered by the user and decreases the number by one until it is equal to zero
4. Declare a variable 'input'
5. Prompt the user to enter a positive number below 10
6. Assign the user's input to the variable 'input'
7. Add the while loop syntax ie while(condition){code}
   Notes:
   The condition is the variable is a positive number
   The code should display appropriate text, including the positive number and then reduce it by 1 and continue to display it until the condition is not true. Alternative text should be displayed when the condition is not true.
8. Save the changes to the file.
9. Figure 2 shows the output when the user enters number 7 after the prompt
10. What would be the effect of incrementing the number?
11. Why is this not recommended?

Figure 2

Number 7 is a positive number
Number 6 is a positive number
Number 5 is a positive number
Number 4 is a positive number
Number 3 is a positive number
Number 2 is a positive number
Number 1 is a positive number
Sorry, the number is not positive.

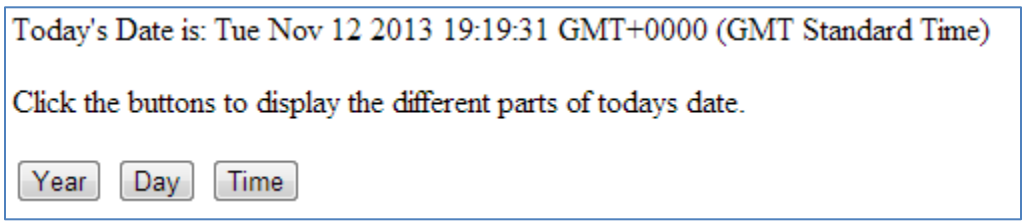
Exercise 5 – Arrays

1. Create a new html file called array.html
2. Place <script> elements in the <head> element.
3. Create and populate an array in one statement
4. Name the array 'bands' and add the names of four bands to it
5. Display the names of all bands
6. Display the names of all bands in reverse order
7. Display the number of bands stored in the array
8. Display the name of the first band
9. Display the name of the last band
10. Save the changes to the file

Exercise 6 – Date

1. Create a new html file called date.html
2. Place <p> elements in the <body> section
3. Assign an id="demo" for the <p> element
4. Place <script> elements in the head of the HTML page.
5. Create a function called year
6. Assign today's date to a variable d
7. Assign the HTML element with an id="demo" to a variable y
   Assign the current year to be the content of variable y
8. Create a button after the <p> element
9. Use the button to call the function year()
10. Add an additional function for the weekday
11. Create an array called weekday and add the weekdays in the array, starting with Sunday
12. Add an additional function for the time
13. Look back over the slides and the examples on Moodle to try and finish off this exercise so that the page looks similar to below when it is displayed in the browser window:

```
function year()
{
var d = new Date();
var y =
document.getElementById("demo");
y.innerHTML=d.getFullYear();
}
```



Note:

- The current date is displayed automatically when the page is loaded
- The next line contains sample text in a <p> element with an id=”demo”
- 3 buttons have been added below the <p> element
- The Year button will call a function to replace the sample text with the current year
- The Day button will call a function to replace the sample text with the current weekday as a name eg Sunday instead of 0
- The Time button will call a function to replace the sample text with the current time which will concatenate the hours and mins with a “:” as a separator eg 15:20