Week 8 Tasks for Submission

This document contains a number of tasks for you to attempt. There are three types of task:

- Tasks labelled as "no submission required, not part of portfolio"
  - They are for you to attempt and practice.
  - They provided assistance for you to develop solutions to other tasks that are part of your portfolio
- Tasks labelled as "PASS Submission Task"
  - These tasks are part of your portfolio geared towards all students
  - The solutions to these tasks must be uploaded for marking by your tutor
- Tasks labelled as "CREDIT Submission Task"
  - These tasks are part of your portfolio geared mainly towards students aiming for more than a pass grade. Of course students aiming for a pass grade may attempt and submit these tasks.

The solutions to these tasks must be uploaded for marking by your tutor

Week 08 references

Lecture 08 (available via Canvas).

Primitive and Reference data types
http://www.javascripttutorial.net/javascript-primitive-vs-reference-values/
http://docstore.mik.ua/orelly/webprog/jscript/ch04_04.htm
http://www.zsoltnagy.eu/understand-value-and-reference-types-in-javascript/

Arrays
http://www.w3schools.com/js/js_arrays.asp
https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/while

Objects
http://www.w3schools.com/js/js_objects.asp
Note: Avoid object methods, constructors and prototypes and the NEW keyword this week.

Task 1. (no submission required, not part of portfolio)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being Arrays and with the relevant meta data.
- The file should display a heading "Task 1"
- Link a JavaScript file to this HTML file.

Create a JavaScript file that:

- Has a function named start() that does the following:
  - Creates an array named **arrValues** which has the following values: **[1,2,3]**
  - Displays the text "List of Array Values:" on the web page.
  - Displays all the values in the array named arrValues on the web page

Task 1

List of Array Values:
1 2 3

Note: You can decide how to write data to the web page and how / when the start() function is called (e.g. as a result of onload event or click on the button).
You may add additional HTML elements.
You may add additional JavaScript functions.
Do not use any global variables in your solution.

Task 2. (no submission required, not part of portfolio)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being List Values and with the relevant meta data.
- The file should display a heading "Task 2"
- Link a JavaScript file to this HTML file.

Create a JavaScript file that:

- Has a function named start() that does the following:
  - Creates an array named **arrValues** which has the following values: **[5,2,0]**
  - Displays the text "List of Array Values:" on the web page.
  - Calls a function named **displayValues()** and pass the array named arrValues as an argument.

Create the function named **displayValues**().

- It must have a single **parameter** (that will accept an array of values).
- The function must **display each of the elements** stored within the array on the web page.
- The function must **not** return a value.

Task 2

List of Array Values:
5 2 0

Note: You can decide how to write data to the web page and how / when the start() function is called.
You may add additional HTML elements.
You may add additional JavaScript functions.
Do not use any global variables in your solution.

Task 3. (no submission required, not part of portfolio)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being More on Arrays and with the relevant meta data.
- The file should display a heading "Task 3"
- Link a JavaScript file to this HTML file.

Create a JavaScript file that:

- Has a function named start() that does the following:
  - Creates an array named **arrValues** which has the following values: **[6,2,4]**
  - Displays the text "List of Array Values:" on the web page.
  - Calls a function named **updateValues()** and pass the array named arrValues as an argument.
  - Calls a function named **displayValues()** and pass the array named arrValues as an argument.

Create the function named **displayValues**().

- It must have a single **parameter** (that will accept an array of values).
- The function must **display each element** stored within the array on the web page.
- The function must **not** return a value.

Create the function name **updateValues**().

- It must have a single **parameter** (that will accept an array of values).
- The function must **set each value** in the array to **zero**.
- The function must **not** return a value.

Task 3

List of Array Values:
0 0 0

Note: You can decide how to write data to the web page and how / when the start() function is called.
You may add additional HTML elements.
You may add additional JavaScript functions.
Do not use any global variables in your solution.

Task 4. (PASS Submission Task)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being Pass Task 1 and with the relevant meta data. **Make sure your name and student ID are part of metadata**.
- The file should display a heading "Pass Task - Arrays"
- Link a JavaScript file to this HTML file.

Create a **JavaScript** file that:

- Has a function named **start()** that does the following:
  - Creates an array named **arrValues** which has 5 to 10 different values (each of which is in the range 0 to 99).
  - Displays the text "Array Values before the update:" on the web page.
  - Calls a function named **displayValues**() and pass the array named arrValues.
  - Calls a function named **updateValues**() and pass the array named arrValues.
  - Displays the text "Array Values after the update:" on the web page.
  - Calls a function named **displayValues**() and pass the array named arrValues.

Create the function named **updateValues**().

- It must have a single **parameter** (that will accept an array of values).
- The function must **add 10 to each value** in the array.
- The function must **not** return a value.

Create the function named **displayValues**().

- It must have a single **parameter** (that will accept an array of values).
- The function must **display each of value** stored within the array on the web page.
- The function must **not** return a value.

You can decide how to write data to the web page and how / when the start() function is called.
You may add additional HTML elements. You may add additional JavaScript functions.

After the start function has been executed, your web page should look silimar to this:

Pass Task - Arrays

Array Values before the update:
1 4 8 2 0 3 7 6
Array Values after the update:
6 9 13 7 5 8 12 11

Produced by John Lee, ID 1234567

Note: You can decide how to write data to the web page and how / when the start() function is called.
You may add additional HTML elements.
You may add additional JavaScript functions.
Do not use any global variables in your solution.

Execute your code and take a screen capture of the web page.

Paste your HTML code JavaScript code and Screen Captures into the file named: **W08P.docx**

Task 5. (no submission required, not part of portfolio)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being About Objects and with the relevant meta data.
- The file should display a heading "Task 5"
- It should also contain a paragraph with id "doglist"
- Link a JavaScript file to this HTML file.

Create a JavaScript file, within which:

- Create a function named start() called as a response to the onload event.
- Place this code at the beginning of the start() function:
  var vDog;
  vDog = {dogBreed:"Poodle", colour:"Black", dogName:"Rollie"};

- Write additional code within the start() function so that the contents of vDog are extracted and displayed onto the Web Page in the following format:

> **Task 5**
>
> Doggy Details:
> Breed: Terrier Colour: Black Name: Rollie

Task 6. (no submission required, not part of portfolio)

- Replace the start() function with this code:

```
function start() {
        var vDog1 = {dogBreed:"Poodle", colour:"Black", dogName:"Rollie"};
        var vDog2 = {dogBreed:"Terrier", colour:"White", dogName:"Rex"};
        var vDog3 = {dogBreed:"Greyhound", colour:"Brown", dogName:"Wilks"};
        var arrDogs = [vDog1, vDog2, vDog3];
        var output = "List of Dogs<br />";
        output = output + arrDogs[0].dogName + "<br />";
        output = output + arrDogs[1].dogName + "<br />";
        output = output + arrDogs[2].dogName + "<br />";
        //Add a statement displaying output in the paragraph with id "doglist"
}
```

Task 7. (no submission required, not part of portfolio)

Modify the start() function.

- Replace the 3 statements adding values to the variable output by
  - Writing a **for** Loop that processes all the elements of the array.
  - The code within the for block must contain a single statement
    output = output + … + "
";

The output generated by your code must be identical to Task 6.

Task 8. (no submission required, not part of portfolio)

Modify the start() function.

- Add two additional dog objects with appropriate breeds, colours and names.
- Test your code again.
- Your code should display all dog names.
  You should **not** have to modify any code in the **for** Loop.

Task 9. (no submission required, not part of portfolio)

Modify the start() function so that it contains:

```
function start() {
var vDog1 = {dogBreed:"Poodle", colour:"Black", dogName:"Rollie"};
var vDog2 = {dogBreed:"Terrier", colour:"White", dogName:"Rex"};
var vDog3 = {dogBreed:"Greyhound", colour:"Brown", dogName:"Wilks"};
var arrDogs = [vDog1, vDog2, vDog3];
displayDogs(arrDogs);
}
```

Create a function displayDogs(arrDogList) { }

Write additional code in the displayDogs() function so that all dog names are displayed. Use a **for** Loop to achieve this. (Hint: you can just copy the relevant code from the start() function to the displayDogs() function.

Finally, modify the statement adding values to the variable **output** so that it displays dogName and dogBreed properties.

Task 10. (PASS Submission Task)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being Pass Task 2 and with the relevant meta data. **Make sure your name and student ID are part of metadata**.
- The file should display a heading "Pass Task - Objects"
- It needs some empty paragraphs with ids to be referred to from JavaScript to display outputs as listed in the following tasks.
- Link a JavaScript file to this HTML file.

Create a **JavaScript** file that:

- Has a function named **start()** containing the following lines of code:
  var vCar1 = {make:" Toyota", colour:"Silver", price:24000};
  var vCar2 = {make:" Mazda", colour:"Red", price:30000};
  var vCar3 = {make:" Audi", colour:"Black", price:90000};
  var arrCars = [vCar1, vCar2, vCar3];

- Add code to the start() function that:
  - Calls a function named **listCars**() and passes the argument **arrCars** to it
  - Calls a function named **displayStats**() and pass the argument **arrCars** to it

- Write the function named **listCars**() which expects a **single** parameter.
  - The parameter is an array of car objects.
  - The function must **display all car details** on the web page in the following format:
  - Make: YYYYYYYY Colour: XXXXXXXX Price: $99999

- Write the function named **displayStats**() which takes a **single** parameter.
  - The parameter is an array of car objects.
  - The function must calculate and display:
    - The number of cars in the array.
    - The total of all car prices (**use a loop** to calculate that).
    - The average price (formatted to 2 decimal places) :
      E.g.
      Total Cars: 3
      Total of all Prices: 144000
      Average Price: 48000

Test and if necessary troubleshoot your code.

Paste your HTML code, JavaScript code and Screen Captures into the file named: **W08P.docx**

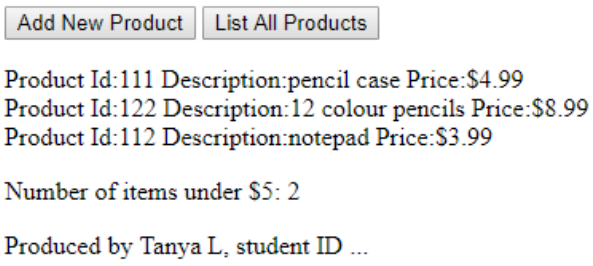
Task 11. (CREDIT Submission Task)

Create a HTML file based on template_upd.html:

- Update the <head> section with the title being Credit Task and with the relevant meta data. **Make sure your name and student ID are part of metadata**.
- The page should look similar to the shown screenshot.
- The HTML page must also have a paragraph with the id value "output".
- Link a JavaScript file to this HTML file.
- When the user clicks the Add New Product button, call the **addProduct**() function in the JavaScript file.
- When the user clicks the List All Products button, call the **listProducts**() function in the JavaScript file.

In the **JavaScript** file:

- Create a **global array** named **garrProducts**.
- Create a function **addProduct().** This function must:
  - Get the id, description and price data from the web page
  - Create a product object using that data
  - Add that object to the array named garrProducts
  - Display the text in the output element of the web page "Product Added. Total number of products now 99" (where 99 is the current length of the garrProduct array).
- Create a function **listProducts()**. This function must:
  Clear the value of the HTML "output" element.
  Display every product in the garrProduct array in "output" using the format:
  Product Id:XXX Description:XXXXXXXXXXX Price:$9999
  After all product details have been displayed, call **countCheap**() passing garrProduct as an argument and display the returned value in the format "Number of items under $5:" 99
- Create a function **countCheap() that expects one parameter – an array.** This function must:
  Examine the price of every product in the garrProduct array, if the price is less than $5 then local variable **count** needs to be incremented.
  The function must return the value of count.



Imagine that the List All Products button was clicked before any product was added to the array. In this case the user should get a message “No products on this list”.

Hint: within the function **listProducts()** use an if statement to check if the length of the array is 0, in this case display the error message, otherwise execute the code listing products and displaying the number of products under $5.

First screen capture the screen after the List All Products button has been clicked before any products have been added. Paste the screenshot into the file named **W08C.docx**

Then test your code by adding at least 6 different products with differing prices (above and under $5). Screen capture the screen after the List All Products button has been clicked.

Paste your HTML code JavaScript code and Screen captures into the file named **W08C.docx**