# 9. Arrays Lab

**Tasks**

Download the starting files from Moodle for this lab.

In this zipped file you will find **ALL** the HTML code.

You need to write the JavaScript code for each function.

In the JavaScript file you will see that I have:

* Created two **global** array variables. Therefore, these variables can be accessed from every function.

// Create global arrays

let names = [];

let titles = [];

* Added the function **addValuesToArrays().**

This function gets values from two textboxes and adds them to the arrays.

// Function to add values to the arrays

function addValuesToArrays() {

    names = document.getElementById("employNames").value.split(" ");

    titles = document.getElementById("employTitles").value.split(",");

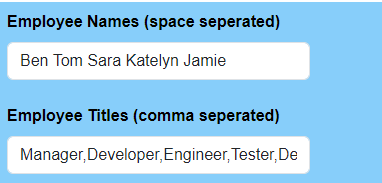
    writeToConsole( names );

    writeToConsole( titles );

    writeToConsole2( names, titles );

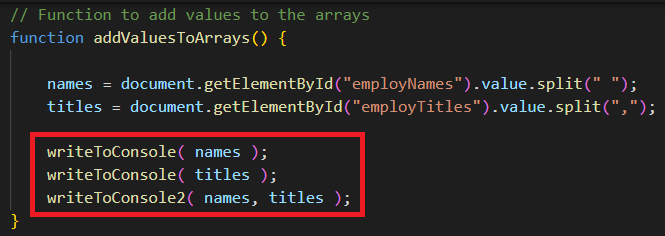
}

Notice below how both the names and titles textboxes have multiple values in them. In the Names textbox, each value is separated by a space. In the title’s textbox, each value is separated by a comma.   
Therefore, when I get ALL the values from both textboxes, I will need to find a way to get each value separately and then add that value to the array.



To help me do this, I will use the JavaScript String method **split**. Look at the two lines of code above where I use **split**. The difference between both lines of code is that I use a space or a comma to separate and get each value. The value is then put in the corresponding array, **names,** and **titles.**

The last part of this function is where I call new functions that I have written named, **writeToConsole**() and **writeToConsole2**().



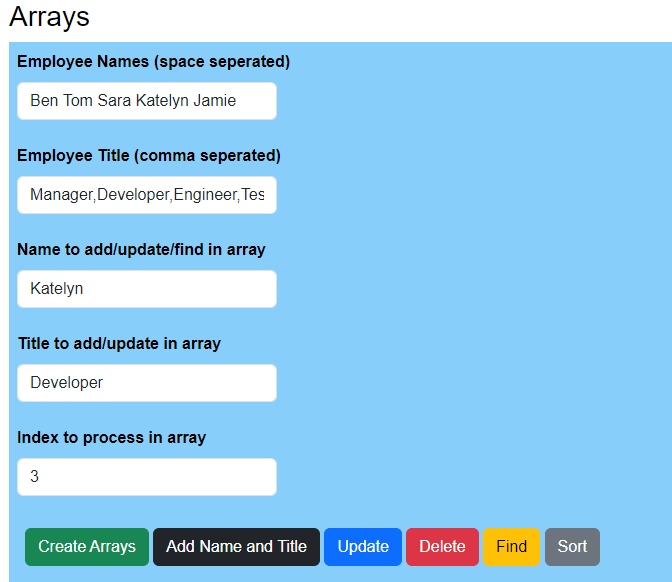
* Notice how I call each function and pass values in as arguments. I don’t necessarily have to pass the array(s) to these functions as they are global variables. However, creating a function that takes an array as a parameter, allows me to print any arrays that I pass into these functions.
* Review my **writeToConsole** and **writeToConsole2** functions in your scripts file.

**Note**

While I will demonstrate each task, it is important you ask any questions if you do not understand the code. Remember CA2 will be based on JavaScript.

**Ensure that you have the console open** when you are testing your code. This way if an error occurs, the console will help/show you.

The following screen is created using HTML. We will click each button and **call a JavaScript function**. The JavaScript functions will access values in the form and then perform calculations on the array.



1. **Create Arrays**

This function has been done for you. Review and ensure you understand it.

function addValuesToArrays() {

    names = document.getElementById("employNames").value.split(" ");

    titles = document.getElementById("employTitles").value.split(",");

    writeToConsole( names );

    writeToConsole( titles );

    writeToConsole2( names, titles );

}

1. **Add name and title.**

The purpose of this function is to add a new name and title to each array.

* Using this function below:

// Add name and title to array

function addNameandTitle() {

}

Add the code to achieve this:

* Get the name (third text box) from the form and add it to the names array (end of the array).
* Get the title (fourth text box) from the form and add it to the titles array (end of the array).
* Call the **writeToConsole2** function, passing in the two arrays as arguments.

1. **Update**

The purpose of this function is to **update** the array with new values (replace current values). We will update both the person’s name and title. Again, we will take the values from the name and title textboxes i.e., the third and fourth textboxes.

However, since we are updating a value in the array, we must know what **position** in the array we are updating. We will get this **position value** from the fifth textbox, named **index**.

To achieve the above, do the following:

* Create a new JavaScript function. To do this, look in your HTML file and see the name of the JavaScript you must create when the user clicks the update button.

Then inside this function

* Get the name, title, and index from the form.
* Remember to convert the index to a Number.

let index = Number( document.getElementById("indexInArray").value );

* Update each array (at this index) with the new name and title.
* Call the **writeToConsole2** function, passing in the two arrays as arguments.

1. **Delete**

The purpose of this function is to delete a value in the array.

* Create a new JavaScript function. To do this, look in your HTML file and see the name of the JavaScript function you must create when the user clicks the delete button.

Then inside this function

* Get the value from the index text box. This is the index/position in the array that we need to delete that value from.
* Call the **writeToConsole2** function
* Delete the value from the names and titles array using this index value.
* Call the **writeToConsole2** function
* Test to ensure the value was deleted.

1. **Find**

The purpose of this function is to find a value in the array. We then want to tell the user if we have found the value or not.

* Create a new JavaScript function. To do this, look in your HTML file and see the name of the JavaScript function you must create when the user clicks the find button.

Then inside this function

* Get the name from the form.
* Loop through the array to see if you can find this name.
* Print to the console if you found or do not find the name in the array.
* Print to the console the index/position you found this name at.

1. **Sort**

The purpose of this function is to sort **one** array.

* Create a new JavaScript function. To do this, look in your HTML file and see the name of the JavaScript you must create when the user clicks the sort button.

Then inside this function

* Sort the name array only.
* Call the **writeToConsole2** function

**Note**

We would need a more detailed way to sort the arrays as both arrays are linked. i.e. position 1 in the names array, relates to position 1 in the titles array. So, by just sorting the name array, we no longer have the correct title in the same position. You could try do this if you have time in the lab.

**Finally:**

Experiment with the other array functions that you have seen during the lecture.