

ICS 128 - Web Scripting

# Instructions

* Use ES6 notation for all variable and function definitions
* Comment your code wherever possible for clarification
* If you get stuck, comment out your code and move on to the next part
* Remember, you can debug your code using the Web Browser Console
* 2% bonus points.

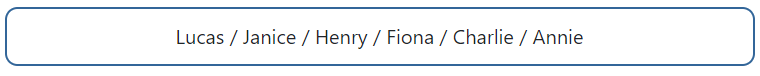
# Exercise #1: Arrays, Loops [8 Points]

1. Create an array that contains the following 12 elements:

Annie, Bob, Charlie, Danielle, Fiona, George, Henry, Isabelle, Janice, Kyle, Lucas

1. Using the appropriate type of loop, iterate over the loop. If the element’s array index is an even number, then add that array item to the beginning of a second different array. For example, the new array would contain he following names in the following order:

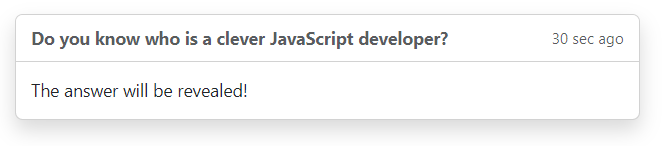
Lucas, Janice, Henry, Fiona, Charlie, Annie

1. Output the content of the array to the existing div with the id “exercise\_1”, so that this section of your page looks like the image below.

Do **not** apply a CSS class, instead, style the div using JavaScript.

# Exercise #2: Debugging [8 Points]

Fix the code in the <script>…</script> block within the HTML marked as “EXERCISE 2”. The page should, in the end, render something like following:

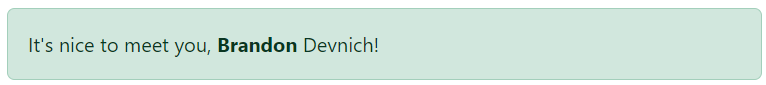
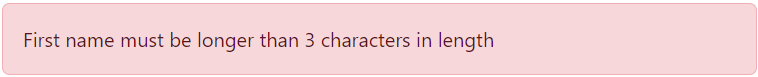
Note: you are not to write or change any HTML, only fix JavaScript syntax errors. Use the Browser Debug Console to fix the 8 errors.

# Part #3: Error Handling [14 Points]

1. Prompt the user for a first name and store it into a variable named “**firstName**”. Prompt the user for a last name and store it into a variable “**lastName**”.
2. Next, write a function named “**getGreeting**” that accepts two variables: a first name and last name.

* Within the function, if the first name has less than 3 characters **or** contains spaces, **throw an error** (not return an error) with the appropriate message describing the problem (see below in red).
* Else, if the last name has less than 3 letters **or** contains spaces, **throw an error** (not return an error) with the appropriate message describing the problem (see below in red).
* Otherwise, **return** a greeting.

1. Invoke the function **getGreeting** within a **try/catch block**, passing along the **firstName** and **lastName** that you prompted the user for.

* If there is no error, then display the message within the Bootstrap alert div using the “success” style.
* If you get an error, then output the error into the div with id “exercise\_3” and style it as a “danger (red)” bootstrap alert.

# Bonus [3 Points]

Add a randomized message (one of 4 messages) to your **getGreeting** function from exercise #3. In other words, the return string from **getGetting** will randomly return 1 of 4 messages. For example, your message randomly might be one of the following:

* It’s nice to meet you, Joseph Nelson
* My day, Joseph Nelson, is better having met you.
* Joseph Nelson, you bring darker clouds to a cloudy day.
* Dear Joseph Nelson,  
  I hope respond to this email…