Goal6: Assignment: MidTerm Practical (4hr)

Mid Term Practical

You will create an array of student objects. Demonstrate your ability to add a new student object to the array of students, then use the DOM to output the student information using innerHTML.

Objectives & Outcomes

Successful completion of this activity will show that you can:

- Assess your skills and comprehension of the materials taught up to this point. (Objects, Arrays, Functions, Conditionals, Methods, Scoping, Debugging, DOM)
- Assess your ability to use all your new programming skills and apply it to a very practical assignment.
- Assess your ability to logically separate code into functions.
- Assess your ability to utilize JavaScript DOM manipulation functions effectively.
- Employ solid craftsmanship.

Level of Effort

This activity should take approximately 240m to complete. It will require:

- 0m Research
- 15m Prep & Delivery
- 225m Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

Reading & Resources

Mid-Term - Rubric *(necessary)*
This rubric outlines the points for the assignment. Make sure you check off each one as done before submitting your assignment.

Assignment Files *(necessary)*
This link has all the files you need to get started with this programming assignment.

Instructions

Getting Started:

- This mid term is OPEN... to a point.
- There will only be 2 restrictions.
    1. You will work on your own to accomplish the programming tasks in this mid term.
    2. You can use the web as a reference only. You cannot use any technology to ask for help. (i.e. NO iChat, screen shares, texting, document sharing, etc...)
- You may ask for help, guidance and direction from others (other students, Lab Instructor(s), or Course Director), but they may not help with coding, nor share code.
- You will need to create a script tag for js/main.js in your .html file.

- You will need to wrap your code (in the js/main.js) in a self-executing function.
- Make sure you adhere to proper folder constructs, if applicable (css, images, js, etc).

Criteria:

To obtain full credit on the assignment your submission should match the functionality of the demonstration. The following criteria must be adhered to and you must satisfy all items on this assignment's rubric.

- Open the js/main.js file.
- There will be no more then 4 global variables. 1 for the student array of objects documented above, 1 for the button (the click event), and 2 other global variables you can use as you wish.
- Create an **array of student objects** containing 3 main properties (keys) and 3 sub properties. The 6 total properties are:
    - name
    - address (this should be an object with 3 sub properties, see below)
        1. street
        2. city
        3. state
    - GPA (this will be an array of GPA scores (grade point averages) for the student in this object)

  Example of 1 object:

  -- {name: 'jbond', address:{address: '3300 University Blvd', city: 'Winter Park', state: 'Fl'}, gpa: [2.5, 3.5, 4.0]};
- In the **array of student objects** above you will need to populate it with at least 2 objects of information (fully populate all key values). The GPA field is an array so you will need to populate the array with a minimum of 3 GPA scores (see the example above).
- Console.log ALL the information in ALL objects on 3 lines.
    1. name
    2. address
    3. GPA:

       Example:
        1. Name: jbond
        2. Address: 3300 University Winter Park Florida
        3. GPA: [2.5, 3.5, 4.0]
- Add a new object to the array of objects above:
    - Create a function call that passes arguments for a new object. The arguments being passed would be the same data items in the already existing objects (name, address, GPA).

      Example of a function call with arguments we are passing to the function:
        - addData('super man', '123 Test Dr', 'Orlando', 'Florida', [3.2, 4.0, 2.2]);
    - The addData function will add this new student information to the existing array of student objects, documented above.
    - The new object format will be similar to the already existing objects.
    - Console.log ALL the information in ALL objects on 3 lines. This should display with the new object that was just added directly above.
        1. name
        2. address
        3. GPA:

           Example:
            1. Name: jbond
            2. Address: 3300 University Winter Park Florida
            3. GPA: [2.5, 3.5, 4.0]

- Create an onclick event on the 'Next' button which, when clicked calls a function that does the following:
    - Use JavaScript's innerHTML property to display ALL object data (one student at a time) in the HTML (like the demonstration).
    - The student's information stored in the object, should display on 4 lines.
        1. name
        2. address
        3. GPA
        4. Average GPA
    - The "Average GPA" directly above should be calculated in a separate function. The function should calculate the student's Average GPA and return the results.
    - When the last student object has been detected AND has already displayed, you will need to do the following:
        1. disable the click event
        2. change the "Next" button text to display "DONE!!!" in the button
- There should not be any duplicate code in the main.js file. If there is, you will need to refractor the code by creating a function for the duplicate code and then calling the function when it is needed.
- Using the **date method**, add a date to the end of each student object. When you refresh your browser , the browser should now display the date as well, for the student that is being displayed. Make sure the date displays in the console.logs data and the innerHTML.
- Make sure to put comments throughout your code.

  Extra Credit:

  Create a function that will validate the "basic" GPA format '#.##'. This function will need to run before the first console.log displays the information. If a GPA does not meet the format, just stop the program and display a console.log error message. You may NOT use regular expression for this extra credit.

GOOD LUCK!!!

Deliverables

You will submit your assignment into FSO using the file format name lastName_firstName_midterm.zip.