

WEB222 - Assignment 2: JavaScript Objects

Grade value: 6% of your final course grade

Objective

Practise JavaScript String, Array and customized objects.

Specifications

Complete the two parts of the assignment as specified below.

Part A [3 marks]

Write a JavaScript program to perform the following tasks. No validation is required for user input – assume that the user will enter valid information.

Create comment line(s) for each of the Steps in assignment2a.js using block comments, indicating the start point of each Steps. e.g.

```

/*****
 * Step 1
 *****/

```

Use VSCode editor to write JS code and use node to run your code.

Here, we will introduce you to the Mockaroo service: <https://mockaroo.com>

Need some mock data to test your app?

Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.

Problem solved.

Use the image below for Field Name, Type and Options. Generate only 10 rows of JSON data.

Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats. Download data using your browser or sign in and create your own [Mock APIs](#).

[Need more data? Plans start at just \\$50/year.](#)

Field Name	Type	Options
<input type="text" value="id"/>	Row Number	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="firstName"/>	First Name	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="lastName"/>	Last Name	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="gender"/>	Gender	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="birthDate"/>	Date	1/8/1992 to 1/8/2000 in <input type="text" value="ISO 8601 (UTC)"/> blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="email"/>	Email Address	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="web"/>	URL	include: <input checked="" type="checkbox"/> protocol <input checked="" type="checkbox"/> host <input type="checkbox"/> path <input type="checkbox"/> query string blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="creditScore"/>	Number	min: <input type="text" value="200"/> max: <input type="text" value="800"/> decimals: <input type="text" value="0"/> blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x
<input type="text" value="rating"/>	Number	min: <input type="text" value="1"/> max: <input type="text" value="20"/> decimals: <input type="text" value="2"/> blank: <input type="text" value="0"/> % <input type="text" value="fx"/> x

Rows: Format: ☒ array ☒ include null values

Hint: Use "." in column names to generate nested json objects, brackets to generate arrays. [More information...](#)

Want to save this for later? [Sign up for free.](#)

Use the data for this part (A)

The data comes to your system as a standard download. Open it for editing. Copy it.

In your code, assign the JSON data as the value of a new variable as an array.

```
var people = // your pasted JSON goes here
```

Don't forget the statement's trailing semicolon.

Now you have a rich body of data, with which you can do the standard get all, get one, add new, edit existing, and delete item tasks.

Task 1

Create a function named **capNotFirstLetter** using the function declaration syntax. The function receives a single parameter of String type. Update / change the first letter of the string to lower case and other letters to upper case. The function returns the updated String. Hint: use the property and methods of String object – length, substr(from, length), substring(from, to) , toUpperCase() and/or toLowerCase().

Task 2

Copy the array to a new array, with something like this:

```
var data = people.map(p => p);
```

Using the function written in Task 1, Update / change the first letter of the string to lower case and other letters to upper case for all FirstName objects in the array. Log the array in the console.

Task 3

Using an appropriate method, filter out those objects with creditScore >400 and assign the array into a new variable. Log the new variable in the console. **Do not use loop for this step.**

Task 4

Sort the data array by date and log the sorted array in the console.

Save your file as assignment2a.js .

Part B [3 marks]

Download JavaScript file **assignment2b.js** from MySeneca/Blackboard under the **Assignments** section. The file contains some given code, including an array (named **courses**) of course objects and a prototype object (named **student**) for creating student objects. Do not change the given code. Write your code beneath the given code and complete the following tasks:

Task 1:

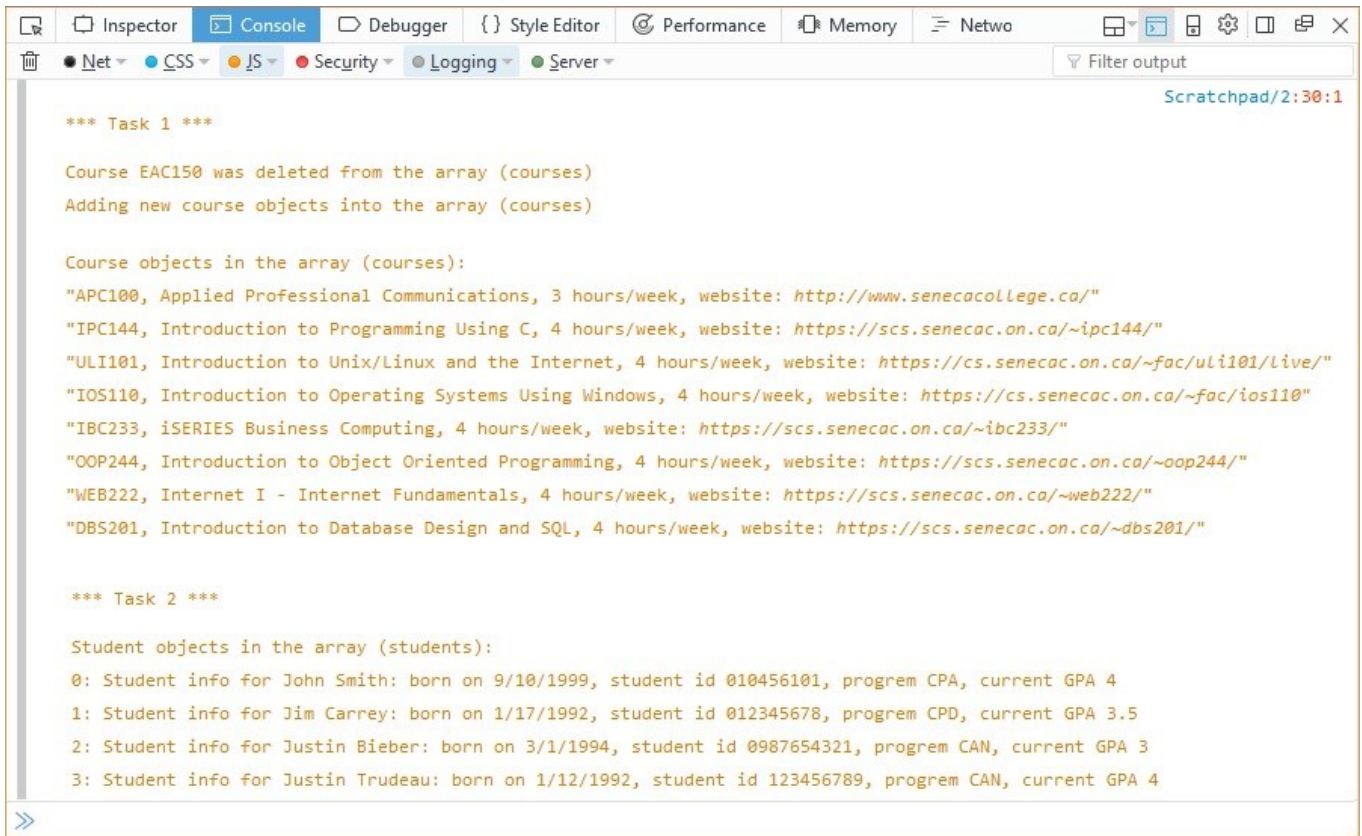
- Remove the last course object from the given array **courses** and store the removed object to a variable.
- Output a message to console to show the course which was removed from the array. Please see the screenshot of outputs on the browser console below. This screenshot was taken using Firefox. You are to log any output in the command prompt console using node.

- c. Create 4 course objects which should have the same properties as what the objects (in the course array) have. Store the 4 course objects in the variables **ibc233**, **oop244**, **web222** and **dbs201**, and the object properties should have appropriate values.
- d. Add these course objects in the array **courses**.
- e. Use for loop to loop through the course array and output the information of the course objects in the array to web console. Please refer the screenshot below.

Task 2:

- a. Create 4 student objects based on the given prototype **student**. Give appropriate property values for all student objects.
- b. Create an array named **students** and add all the student objects into the array.
- c. Use the **forEach** method to iterate the array students and output the information of the student objects to the web console. Please refer the screen below.
- d. Save your JavaScript code to file **assignment2b.js**.

Click the image to open it full-size in a new tab/window.



```

*** Task 1 ***

Course EAC150 was deleted from the array (courses)
Adding new course objects into the array (courses)

Course objects in the array (courses):
"APC100, Applied Professional Communications, 3 hours/week, website: http://www.senecacollege.ca/"
"IPC144, Introduction to Programming Using C, 4 hours/week, website: https://scs.senecac.on.ca/~ipc144/"
"ULI101, Introduction to Unix/Linux and the Internet, 4 hours/week, website: https://cs.senecac.on.ca/~fac/uli101/live/"
"IOS110, Introduction to Operating Systems Using Windows, 4 hours/week, website: https://cs.senecac.on.ca/~fac/ios110"
"IBC233, iSERIES Business Computing, 4 hours/week, website: https://scs.senecac.on.ca/~ibc233/"
"OOP244, Introduction to Object Oriented Programming, 4 hours/week, website: https://scs.senecac.on.ca/~oop244/"
"WEB222, Internet I - Internet Fundamentals, 4 hours/week, website: https://scs.senecac.on.ca/~web222/"
"DBS201, Introduction to Database Design and SQL, 4 hours/week, website: https://scs.senecac.on.ca/~dbs201/"

*** Task 2 ***

Student objects in the array (students):
0: Student info for John Smith: born on 9/10/1999, student id 010456101, program CPA, current GPA 4
1: Student info for Jim Carrey: born on 1/17/1992, student id 012345678, program CPD, current GPA 3.5
2: Student info for Justin Bieber: born on 3/1/1994, student id 0987654321, program CAN, current GPA 3
3: Student info for Justin Trudeau: born on 1/12/1992, student id 123456789, program CAN, current GPA 4
  
```

Assignment submission

- Add the following declaration at the top of your **assignment2a.js** and **assignment2b.js** files (failure to do this step will result in zero mark assigned to your assignment 2):

```

/*****
*
*           WEB222 - Assignment 2
* I declare that this assignment is my own work in accordance with Seneca
* Academic Policy. No part of this assignment has been copied manually or
* electronically from any other source (including web sites) or distributed to
* other students.
*
* Name: _____ Student ID: _____ Date: _____
*
*****/
  
```

- Compact your files **assignment2a.js**, **assignment2b.js** into a zip file named **assignment2-<your name>.zip**.
- Submit the zip file to the Blackboard (My.Seneca).

Important note

- **NO LATE SUBMISSIONS** for assignments. Late submissions will not be accepted and will receive a grade of zero (0).
- **After the end (11:30PM) of the due date/time**, the assignment submission link on the Blackboard will no longer be available.
- **Allow enough time to upload. Do not wait for the last moment** since there are transmission time/queuing delay/processing time etc. from your machine to the blackboard server.