COMP 2406A
Winter 2021 - Tutorial #1

Objectives

- Installing Node.js
- Practicing basic Javascript programming using strings, arrays, and functions
- Practicing basic HTML/CSS

Expectations

To receive full grades for this tutorial, all problems must be completed. For each tutorial, you will receive:

- 4/4 for submitting high quality solutions to all problems
- 3/4 for submitting solutions for all problems but some need improvement
- 2/4 if you are missing problems or your solutions need significant improvement
- 1/4 if you are missing several problems or your solutions are poorly done
- 0/4 you do not make a submission or your submission cannot be executed

---

Problem 1 (Installing Node.js)

Go to https://nodejs.org/en/ and download/install Node.js version 14.x. You can change the installation location if you want but be sure not to deselect any of the features on the 'custom setup' page of the installation wizard; the default options will install everything we need.

Once installed, you should be able to run the command `node -v` from the command line and see the version number printed out. If you did this successfully, Node.js is now installed on your computer. Node comes packaged with a second application called NPM. We will not need NPM for this tutorial, but you should check that it installed correctly by running npm -v in the command line. If it prints out a version number, you're good to go.

You can run any Javascript files you create for this tutorial by navigating to the file's directory from the command line and issuing the command:

```
node YourFileName.js
```

Problem 2 (A Basic Javascript Program)

Write a program that prints out the following pattern to the console (7 rows):

```
#
##
###
####
#####
######
#######
```

Your program should contain no more than one console.log() statement (looped over multiple times) and you should be able to adjust the number of rows printed by changing one number within your code. Save your code in a file with the .js file extension and execute it using Node.js.

Problem 3 (Working with Javascript Strings)

Download the stringproblems.js file from cuLearn. Run the code with Node.js. It will write output to the console window. It should produce a simple string output meant to look like a landscape profile like this:

```
___/'''''\_______/'\_
```

Modify this code so that it uses the '_' underscore character for both the flat portions and the tops of the hills. Before you start, think about how you need to modify the existing code to accomplish this. Think about what you would need to do to put an underscore ‘above’ a line of text. You may need more variables and/or more lines of text in the result. The output should now look like:

```
    ____          _
___/    \_______/  \_
```

Problem 4 (Higher Order Array Functions)

Download the students.js file from cuLearn. This file contains an array called *students*, which contains several Javascript objects representing student information. Add code to this file which uses the higher order array functions (`filter()`, `map()`, `reduce()`) to generate and output the information required below:

1. Generate an array containing the first names of all students. Print it to the console to verify.
2. Generate and print an array containing the full names of students (first and last name separated by " ") of all students who received an exam grade of 80 or higher. The students should be ['James Johnson', 'Stephanie Ottesen', 'Leonard Arvan', 'Beverly Mott', 'Beatrice Jaco']
3. Generate and print the total average final grade of all students. Each student's final grade should be calculated with the weighting: assignment=40%, tutorial=10%, exam=50%. The average of all grades should be ~ 71.36%.

If you are struggling, remember to break the problems down into smaller steps (e.g., get the filter to work first before implementing the map portion). If you're not sure how to use filter(), map(), or reduce(), read about them in the Eloquent Javascript book (see assigned readings) or look them up in w3School's JavaScript reference:

https://www.w3schools.com/jsref/jsref_obj_array.asp

Problem 5 (Static HTML Page)

Create a basic HTML file that presents the user with some information about a movie. The page should show the movie's name, release date, and duration. The page should also contain a list of some of the actors in the movie and a section with reviews of the movie. Try using some different CSS styling to change the look of the page. Your page does not need to look fancy but should be well-organized. If you're uncertain how to do something in HTML or CSS, you may wish to consult w3School's references for each:

https://www.w3schools.com/html/default.asp
https://www.w3schools.com/css/default.asp

Problem 6 (Submission)

Zip all of your files for this tutorial and submit them to the Tutorial #1 submission on cuLearn. Make sure you download your zip file and check its contents after submitting. If your zip file is missing files or corrupt, you will lose marks.