

# COMP 2406 A

## Winter 2020 - Tutorial #1

---

### Objectives

- Logging in to the SCS undergrad computers
- Installing Node.js
- Practicing basic Javascript programming using strings, arrays, and functions
- Practicing basic HTML/CSS

### Expectations

- At minimum, you should complete problems 2 and 6 from this tutorial document. Problems 3-5 will provide you some good problem solving and Javascript programming practice. Remember to use the available resources (w3schools, Node.js documentation, Eloquent Javascript book, lecture materials, etc.) for more information if you are struggling to complete the problems.
- 

### Problem 1 (Lab Accounts)

Note: If you are using your own computer, you do not need to complete this problem. However, you may want to anyway, so that you have lab access for the future. Everyone who is registered in the course has access to the computers in the tutorial room (HP 4155) and some other SCS lab rooms. For information about all SCS computer labs see:

<https://www.scs.carleton.ca/technical-support/computer-laboratories>

In order to use these computers, however, you will need to set up your password for your lab account. Please note: this is *not* the same as your My Carleton One or your Carleton University email account. If this is your first time accessing SCS lab computers, you will need to set up this account. If you already have a Carleton University email account, you can set up your SCS (School of Computer Science) account from any computer by following the instructions found at:

[https://newacct.scs.carleton.ca/scs\\_authentication/newacct-policy-form.php](https://newacct.scs.carleton.ca/scs_authentication/newacct-policy-form.php)

You can also set up your account when you come to your first tutorial. The TAs will be around to help you with this process.

## Problem 2 (Installing Node.js)

If you are using the lab computers, you can skip the installation step, as Node.js is already installed on the lab computers. If you are using your own computer, go to <https://nodejs.org/en/> and download/install Node.js version 12.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 3 (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 4 (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 5 (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()`, look them up in w3School's JavaScript reference for a refresher:

[https://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](https://www.w3schools.com/jsref/jsref_obj_array.asp)

## Problem 6 (Static HTML Page)

Note: since the tutorial is not graded, the actual content of the HTML page you make for this problem can be anything you want. Come up with a design for a page that has some different HTML components and write the HTML/CSS to make it display properly when opened in a browser.

Create a basic HTML file that presents the user with a few multiple-choice trivia questions. The page should also have a textbox for the user to enter their name and a “Check Answers” button. For the time being, the button does not have to do anything. Open your HTML file in the browser to make sure it looks right. Try using some different CSS styling to change the look of the page. An example of what your page may look like is below:

### Welcome to the Trivia Page

Enter your name:

1. What is the capital of Ontario?

- ☐ a. Ontario City
- ☐ b. Alberta
- ☐ c. Toronto
- ☐ d. Ottawa

2. What is the capital of Canada?

- ☐ a. Ontario City
- ☐ b. Alberta
- ☐ c. Toronto
- ☐ d. Ottawa

If you're uncertain how to do something in HTML or CSS, you may wish to consult w3School's references for each language:

<https://www.w3schools.com/html/default.asp>

<https://www.w3schools.com/css/default.asp>

---

## Reminders

### Save Your Work:

Anything saved to the Desktop or C-drive of lab computers will not be preserved after you log off! Instead, each student is given space on a network drive, called the Z drive, that is accessible from any lab computer.

You can use Dropbox, Google Drive, your Z-drive, email it to yourself, or any other way to make a copy that is external to the PC in the tutorial room. Get used to backing up your work often as you do your assignments.