

Tutorial 1

---

# COMP 2406 B - Fall 2022

## Tutorial #1

**Due Sunday, September 25, 23:59**

---

### Objectives

The goal of this tutorial is to make you comfortable with your coding environment and to continue practice HTML, CSS, and JavaScript. The agenda is:

- Practicing basic JavaScript programming using strings, arrays, and functions
- Practicing basic HTML/CSS

### Expectations

You need to submit solutions to all the problems in this tutorial. Our TAs will mark your submission. 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.

**Marking scheme.** For each tutorial, you will receive:

- 2/2 for submitting high-quality solutions to all problems. “High-quality” means that your code works (solves the problem) and is also neat and concise (no overengineering, please).
  - 1/2 for submitting solutions for all problems but some need improvement, or you are missing a problem.
  - 0/2 if you are missing several problems or your solutions are poorly done; or you do not make a submission, or your submission cannot be executed.
- 

### Problem 1 (Practice with JavaScript Loops)

Create a JS file named **insects.js** and write your solution to this problem there.

1. A given insect population doubles every week. If 2 insects exist on the first week, how many weeks will pass until the insect population exceeds 10,000 insects? Use a while loop to output the insect population each week until the population exceeds 10,000 insects.

Output:

2  
4

---

Tutorial 1

8  
16  
32  
64  
128  
256  
512  
1024  
2048  
4096  
8192  
16384

It will take 14 weeks for the insect population to exceed 10,000 insects

2. Researchers have discovered that every 4 weeks a disease kills 40% of the insect population after the population has reproduced. If 2 insects exist on the first week, how many weeks will pass until the insect population exceeds 10,000 insects? Write a do-while loop that outputs the insect population each week until the population exceeds 10,000 insects and stays above 10,000 despite the disease. Decimal places will appear in the number of insects after removing 40% of the population on week 4.

Output:

2  
4  
8  
16  
19.2  
38.4  
76.8  
153.6  
184.32  
368.64  
737.28  
1474.56  
1769.472  
3538.944  
7077.888  
14155.776  
16986.9312

It will take 17 weeks for the insect population to exceed 10,000 insects when 40% of the population is killed by disease every 4 weeks.

**Tutorial 1**

## Problem 2 (JavaScript Function)

Download the **drawBox.js** file from Brightspace. The code in the file produces a 5 x 10 box of question marks. Convert the code into a function called `drawBox()` that has three parameters:

`numRows` - The number of rows for the box.

`numCols` - The number of columns for the box.

`boxChar` - The character to use to create the box. If no argument is supplied, use "X".

For example: `drawBox(5, 4, "!")` and `drawBox(2, 6)` should display the boxes pictured here:

```

!!!!
!!!!
!!!!
!!!!
!!!!
XXXXXX
XXXXXX

```

If you are struggling, check lecture notes or consult w3School's JavaScript reference for a refresher:

[JavaScript Functions \(w3schools.com\)](https://www.w3schools.com/js/js_functions.asp)

## Problem 3 (Working with JavaScript Strings)

Download the **stringproblems.js** file from Brightspace. 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 this:

```

____/_____\_____/_____\_

```

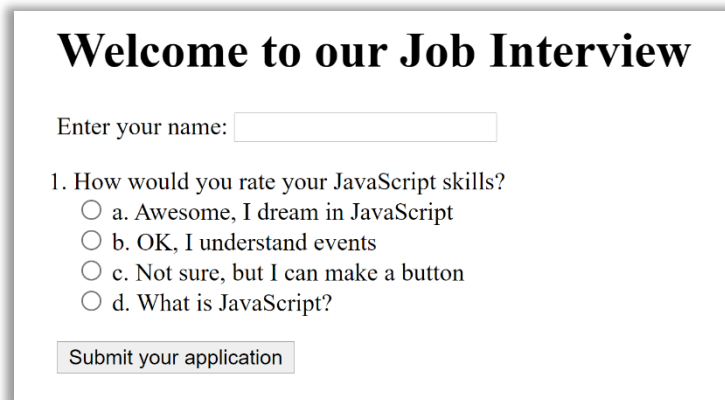
**Tutorial 1**

## Problem 4 (Static HTML Page)

Note: 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 questions. The page should also have a textbox for the user to enter their name and a “Submit” 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. Your page does not need to look fancy but should be well-organized.

An example of what your page may look like is below:



The example form is titled "Welcome to our Job Interview" in a large, bold, black serif font. Below the title is a text input field preceded by the label "Enter your name:". Underneath the name field is a question: "1. How would you rate your JavaScript skills?". This is followed by four radio button options: "a. Awesome, I dream in JavaScript", "b. OK, I understand events", "c. Not sure, but I can make a button", and "d. What is JavaScript?". At the bottom of the form is a button labeled "Submit your application". The entire form is enclosed in a light gray border with a subtle drop shadow.

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 5 (Save Your Work & Submit)

Keep your files organized. For example, place them into “Tutorial-01” folder, where you can easily find them later for your reference and for submission.

To submit your tutorial, you will need to **zip** (compress) all your files for the required tutorial into a single .zip file and submit it to the Tutorial submission on Brightspace. Name your file **T1-YourName.zip**. 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.