# Unit 4 Lab - Dynamic Client-side Scripting - JavaScript

In this lab, you will practice using JavaScript.

* Create a new folder called “unit-4” in the “front-end-web-development” folder

**Exercise 1 - Number of Stars**

1. Create a new folder called “number-of-stars”
2. In the folder create a html file called “number-of-stars.html”
3. Create a “js” folder and in that folder create file called “main.js”
4. In the html file add the following code

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Exercise 1 - Stars</title>

  <script src="js/main.js"></script>

</head>

<body>

</body>

</html>

1. In the main.js file add the following code and save your changes

  // Declare variables

  let numberOfStars = 0;

  let outputStars = '';

  // inputs

  numberOfStars = prompt("Enter the number of stars?");

 // loop

  for (let i=0; i < numberOfStars; i++ ) {

    outputStars += "\*";

 }

 // output

 alert("Outputting " + numberOfStars + " stars: " + outputStars)

1. View the html page in Chrome. The prompt will be displayed. Enter a number and hit “ok”

A screenshot of a computer

Description automatically generated

1. The result will then be displayed

A white background with a blue and red object

Description automatically generated with medium confidence

1. This code will not work if the user enters any characters which as not a number. To fix this add the following code

  // check that user entered a number and that it is greater than zero

  // otherwise ask user to enter another number

  while (isNaN(numberOfStars) || numberOfStars <=0) {

    alert("You did not enter a number greater than zero.");

    numberOfStars = prompt("Enter a number greater than zero?");

  }

**Exercise 2 - Factorial**

1. Create a new folder, html and js file for this exercise as before
2. In main.js add the following code to create the factorial function and declare and initialise any variables

function factorial() {

  // declare variables

  let number = 0;

  let factorial = 1;

  let output = "";

}

factorial();

1. Add the input code

  // input

  number = parseInt(prompt("Enter a number between 1 and 20"));

  while ( isNaN(number) || number <= 1 || number >= 20) {

    number = parseInt(prompt("Please enter a number between 1 and 20"));

  }

1. Add the code to calculate the factorials. This uses two “for” loops. The inner loop calculates the factorial for a specific number while the outer loop generates the factorial for a range of numbers

  output = "Factorials up to " + number + ": ";

  // calculate factorials

  for (let j=number; j >= 1; j--) { // outer loop

    factorial = 1;

    for (let i=1; i <= j; i++) { // inner loop

      factorial \*= i;

    }

    output += j + '! = ' + factorial + ', ';

  }

1. Add the code to output the final result to the user using an alert()

  // output factorials

  alert(output);

1. For an input number of “5”, the output will look like this

A screenshot of a computer

Description automatically generated

1. An improvement to our code is to move the inner loop code to its own function called fac(). First create the arrow function

const fac = (num) => {

  let fac = 1;

  for (let i=1; i <= num; i++) {

    fac \*= i;

  }

  return fac;

}

1. Next replace the inner loop with this code to call the new fac function

    //factorial = 1;

    //for (let i=1; i <= j; i++) { // inner loop

    //  factorial \*= i;

    //}

    //output += j + '! = ' + factorial + ', ';

    output += j + '! = ' + fac(j) + ', ';

1. A variation to this solution is to replace the two “for” loops with “while” loops

**Challenges**

Create a sparate folder, html page and a linked JavaScript file for each challenge. Follow the instructions as outlined previously

For these challenges, print the output to the console using “console.log()“. Any input can be hardcoded in the code itself.

1. Write a function that takes a positive integer *N* and then calculates and displays the sum of the first *N* odd integers. For example, if *N* is 4, your function should display the value 16, which is 1 + 3 + 5 + 7.
2. Write a function drawConsolePyramid(height) that draws a pyramid of the specified height in which the width of each row increases by two as you move downward on the console. Each of the rows should be centered with respect to the others, and the bottom line should begin at the left margin. Thus, calling drawConsolePyramid(8) should produce the following figure:

Table

Description automatically generated

1. Write a function called "multiplyArray" that takes an array of numbers as a parameter and returns the product of all the numbers in the array.
2. Write a function randomAverage(n) that generates n random real numbers between 0 and 1 and then returns the average of those n values. Statistically, calling randomAverage(n) will produce results that become closer to 0.5 as the value of n increases. Write a main program that displays the result of calling randomAverage on 1, 10, 100, 1000, 10000, 100000, and 1000000. See [Math.random()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random) for details on how to generate a random number