You are not allowed to change the JS File names or modify the folder structure.

1. In D01/EX01, create a website that runs the JS attached file. In that file, fill in the function provided with one line of code that logs to the console “Hello World”. Call that function. You should open devtools (in your browser), and go to the “console” tab and see the string you just logged.

To test this exercise run: npm run testex1

1. In D01/EX02, create a website that runs the JS attached file. In that file, edit the function provided. That function, should accept one parameter (do not call it x, y.. Give it a real name). The code should print to the console “The number Y is bigger than 4” when the number is bigger than 4, or “the number Y is equal to 4” when the number is equal to 4, or “the number Y is less than 4” when the number is less than 4 (where Y is the value of your variable). Change the value to your variable, and check if you get the right output every time (do not forget to refresh the page after you change your code).

To test this exercise run: npm run testex2

1. In D01/EX03, create a website that runs the JS attached file. In that file, edit the function provided. Use as template your previous function. This function should accept two parameters instead of one, where the first one is the number to be compared and the second one the one to compare with. You need to compare the numbers, and according to the right situation print in the console:

The number X is bigger than Y

The number X is less than Y

The number X is equal to Y

E.g. you should have “The number Y is bigger than X” (where X is the value of your second variable).

To test this exercise run: npm run testex3

1. In D01/EX04, create a website that runs the JS attached file. In that file, edit the function provided. Use as template your previous function. Extend your code, to print “The modulo of X % Y is Z”, only if X is bigger than Y, and if the module to X % Y isn’t zero. It still should print the previous lines in case X is less than Y or X equals Y.

To test this exercise run: npm run testex4

1. In D01/EX05, create a website that runs the JS attached file. In that file, edit the function provided. Use as template your previous function. Change your code to have only one console.log after all of the if-else statements (hint: use a string variable, and change it accordingly to the logic of your code) .

To test this exercise run: npm run testex5

1. In D01/EX06, create a website that runs the JS attached file. In that file, edit the function provided. Use as template your previous function. Instead of logging a string to the console, create a h3 tag with an id of ‘output’, and change the text of the tag according to the logic you implemented.

To test this exercise run: npm run testd1ex6

1. In D01/EX07, create a website that runs the JS attached file. In that file, edit the function provided. Use as template your previous function. Add a new function that randomize the values of your two variables, using [Math.random()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random), with the range of 1 - 100. Call to that function inside the other one instead of providing two parameters.

To test this exercise run: npm run testex7

To run tests in the whole folder: npm run testd1