

# Functions

**DEADLINE:** 20/06/2022, 18:00

## FOLDER STRUCTURE

```
COE1_HW5/*  
└─ homework/*  
    └─ index.html*  
    └─ index.js*  
    └─ .eslintrc.js
```

\* - required

## TASK

### Task #1

Write a function - *isEqual*

It should accept two arguments and returns **true** if first one value equals second one or **false** otherwise.

**Tip:** no need for if/else clause nor ternary operator

**For example:**

```
isEqual(3, 3) // => true
```

### Task #2

Write a function - *isBigger*

It should accept two arguments and returns **true** if first one has **greater** value than second one or false otherwise.

**Tip:** no need for if/else clause nor ternary operator

**For example:**

```
isBigger(5, -1) // => true
```

### Task #3

Write a function - *storeNames*

It should accept an **arbitrary** number of strings and return an array of that strings

**For example:**

```
storeNames('Tommy Shelby', 'Ragnar Lodbrok', 'Tom Hardy')  
// => ['Tommy Shelby', 'Ragnar Lodbrok', 'Tom Hardy']
```

### Task #4

Write a function - *getDifference*

It should accept two arguments as numbers and return their difference. But the function *never returns a negative value*. If second parameter is greater than first one, function will change their order.

For example:

```
getDifference(5, 3) // => 2  
getDifference(5, 8) // => 3
```

### Task #5

Write a function - *negativeCount*

It should accept an array of numbers and return the count of negative values from the array.

**For example:**

```
negativeCount([4, 3, 2, 9]) // => 0
negativeCount([0, -3, 5, 7]) // => 1
```

### Task #6

Write a function – *letterCount*

It accepts two string arguments and returns an integer of the count of occurrences the 2nd argument is found in the first one.

If no occurrences can be found, a count of 0 should be returned.

**For example:**

```
letterCount("Marry", "r") // => 2
letterCount("Barney", "y") // => 1
letterCount("", "z") // => 0
```

### Task #7

Our basketball team (**x** – our team) completed the championship. The result of each match look like "**x:y**".

Results of all matches are recorded in the collection like this: ["95:74", "107:107", "99:110", ...]

Write a function – *countPoints*

It should accept a collection of football games scores and count the points of our team in the championship.

Rules for counting points for each match:

- if  $x > y$  - 3 points
- if  $x < y$  - 0 point
- if  $x = y$  - 1 point

**For example:**

```
countPoints(['100:90', '110:98', '100:100', '95:46', '54:90', '99:44', '90:90', '111:100']) // => 17
```

## RESTRICTIONS

- Usage of **Math object** is forbidden;

## BEFORE SUBMIT

- Remove all unnecessary files that you might have included by mistake
- Verify that all functionality is implemented according to requirements
- Make sure you code is well-formatted, and validated via validator (w3org Markup Validation Service)
- Add comments if the code is difficult to understand
- Fix warnings/errors in the browser console

- Verify that the name of the folders and files meet the requirements
- Make sure there are no errors/warnings in the browser console
- Run the linter and fix all warnings and errors.

## HOW TO

Use linter :

- In order to use npm package manager you should install nodejs (<https://nodejs.org/> )
  - Install eslint to check your code (npm install -g eslint)
    - open a terminal(or cmd)
    - run eslint (i.e. eslint ./ index.js)
- Code should be without 'errors'

## SUBMIT

The folder should be uploaded to gitlab repository '**COE-1**' into **main** branch