Data types

**DEADLINE:** 14/01/2020

## FOLDER STRUCTURE

|  |  |
| --- | --- |
| FL12\_HW9/\*  └─ homework/\*  └─ index.html\*  └─ index.js\*  └─ .eslintrc.js\* | \* ­­­- required |

## TASK

Write all tasks inside index.js file

1. Write a function that converts the argument values. If it will be a string, convert it to number and wise versa. It should return an array of converted values.

convert('1', 2, 3, '4') // [1, '2', '3', 4]

2. Write function, which iterates over array and executes function on each element.

executeforEach([1,2,3], function(el) {console.log(el \* 2)}) // logs 2 4 6

3. Write function, which returns transformed array based on function, which passed as a second parameter (callback). If array contains a number as string, it should convert it and return as number. You’re allowed to change a body of that callback function if you need. Reuse function from task 2.

mapArray([2, '5', 8], function(el) {return el + 3}) // returns [5, 8, 11]

4. Write function, which returns filtered array based on function, which passed as a parameter. Reuse function from task 2.

filterArray([2, 5, 8], function(el) { return el % 2 === 0 })

// returns [2, 8]

5. Write a function that reverses the string value passed into it.

flipOver('hey world') // 'dlrow yeh'

6. Write a function which creates an array from the given range of numbers

makeListFromRange([2, 7]) // [2, 3, 4, 5, 6, 7]

7. Write a function that accepts an array of object and returns new array of values by passed key name.

That function should not change the original array. Reuse function from task 2.

const actors = [

{ name: ‘tommy’, age: 36 },

{ name: ‘lee’, age: 28 }

];

getArrayOfKeys(actors, ‘name’); // [‘tommy’, ‘lee’]

8. Write function substitute() that accepts an array of numbers and manages to replace all numbers lower than 30 with '\*'. It should return a new array with numbers and '\*' instead of numbers lowest from 30. Reuse function from task 3.

substitute([58, 14, 48, 2, 31, 29]); // [58, '\*', 48, '\*', 31, '\*']

9. Write a function which returns a day number that was some amount of days ago from the passed date.

It should not change the given source date.

const date = new Date(2019, 0, 2);

getPastDay(date, 1); // 1, (1 Jan 2019)

getPastDay(date, 2); // 31, (31 Dec 2018)

getPastDay(date, 365); // 2, (2 Jan 2018)

10. Write a function that formats a date in such format "YYYY/M/d HH:mm".

formatDate(new Date('6/15/2018 09:15:00')) // "2018/6/15 09:15"

formatDate(new Date()) // "2020/1/7 12:56" // gets current local time

## RESTRICTIONS

* Using built–in array or object methods(besides push, length and date methods)
* Using built–in string methods (except parseInt)
* Using any external libraries

## BEFORE SUBMIT

* 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 ./js/task1.js)

Code should be without ‘errors’

* Make sure the code is clean, readable, and tested
* Verify that all functionality is implemented according to requirements
* Delete all unnecessary comments
* Validate code via eslint

## SUBMITs

* The folder should be uploaded to github repository '**FL12**' into **master** branch

## USEFUL LINKS

* <https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Date>
* <https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Array/prototype>
* <https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Object/prototype>
* <https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/String>