Data types

DEADLINE: 17/10/2020

FOLDER STRUCTURE

FL14_HW9/*
- homework/*
  - index.html*
  - index.js*
  - .eslintrc.js

* - required

TASK

Write all of the tasks inside `index.js` file

1. Write a function that accepts list of numbers either in string or number format and then converts the values. If it is 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)}) // 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 takes an array and a value as arguments and returns the value position(not index) in that array. If value is not present in passed array – return false.

```
getValuePosition([2, 5, 8], 8)  // returns 3
getValuePosition([12, 4, 6], 1)  // returns false
```

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

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

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

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

8. 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 fruits = [
  { name: 'apple', weight: 0.5 },
  { name: 'pineapple', weight: 2 }
];

getArrayOfKeys(fruits, 'name');
// returns ['apple', 'pineapple']
```

9. Write a function that accepts an array of groceries objects and returns total weight of all items. Reuse function from task 2.

```
const basket = [
  { name: 'Bread, weight: 0.3 },
  { name: 'Coca-Cola, weight: 0.5 },
  { name: 'Watermelon, weight: 8 }
];

getTotalWeight(basket)
// returns 8.8
```

10. 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(2020, 0, 2);
getPastDay(date, 1); // 1, (1 Jan 2020)
getPastDay(date, 2); // 31, (31 Dec 2019)
getPastDay(date, 365); // 2, (2 Jan 2019)
```

11. Write a function that formats a date in such format "YYYY/MM/DD HH:mm".

```
formatDate(new Date('6/15/2019 09:15:00')) // "2018/06/15 09:15"
formatDate(new Date()) // "2020/04/07 12:56" // gets current local time
```

RESTRICTIONS

- Using built–in array or object methods(besides *push*, *length* and date methods) is forbidden
- Using built–in string methods (except *parseInt*) is forbidden
- Using any external libraries 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 ./js/task1.js)

  Code should be without ‘errors’

SUBMIT

- The folder should be uploaded to gitlab repository '**FL-14**' 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