

Date. Regular Expression. JS Sorting Methods

DEADLINE: 12/05/2022

FOLDER STRUCTURE

```
FL19_HW11/*
├─ homework/*
│   └─ index.html*
│   └─ index.js*
└─ .eslintrc.js
```

* - required

TASK

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

1. Write a function that accepts a date/timestamp and returns a textual representation of the corresponding weekday (i.e. 'Monday', 'Tuesday', etc.). For the sake of the task, let's assume that the input is always a valid date object or a timestamp.

```
getWeekDay(Date.now()); // "Thursday" (if today is the 22nd October)
getWeekDay(new Date(2020, 9, 22)); // "Thursday"
```

2. Write a function that will return the number of days until the New Year.

```
getAmountDaysToNewYear(); // 124 (if today is the 30th August)
getAmountDaysToNewYear(); // 365 (if today is the 1st January)
```

3. Write a function that accepts a person's date of birth as an input and if person is more than 18 years old, than return the string whether person is allowed to pass or not. For the sake of the task, let's assume that the input is always a valid date object.

If persons age is equal or more than 18 return: `// Hello adventurer, you may pass!`
If persons age is less than 18 return: `// Hello adventurer, you are to yang for this quest wait for (if less than one year left than "few more months" if more than year than `${years left} years more`)`!

(assuming today is the 01st January 2022)

```
const birthday17 = new Date(2004, 12, 29);
const birthday15 = new Date(2004, 12, 29);
const birthday22 = new Date(2000, 9, 22);
```

```
getApproveToPass(birthday17); // Hello adventurer, you are to yang for
this quest wait for few more months!
```

```
getApproveToPass(birthday17); // Hello adventurer, you are to yang for
this quest wait for 3 years more!
```

```
getApprovedToPass(birthday22); // Hello adventurer, you may pass!
```

4. Given string (`'tag="div" class="main" style={width: 50%;} value="Hello World!"'`) write a function that from this string returns a string with a div element (`<div class="main" style="width: 50%;">Hello World!</div>`):

```
const elementP = 'tag="p" class="text" style={color: #aeaeae;} value="Aloha!'"

transformStringToHtml(elementP);
// '<p class="text" style="color: #aeaeae;">Hello World!</p>'
```

5. Write a function that accepts a string as an input and returns a boolean that defines if the input is a valid JavaScript variable. Use a regular expression to validate the input. Here is the syntax for valid identifiers:

- each identifier must have at least one character.
- valid identifier characters are the following: alpha, digit, underscore, or dollar sign.
- the first character cannot be a digit.

```
isValidIdentifier('myVar!'); // false
isValidIdentifier('myVar$'); // true
isValidIdentifier('myVar_1'); // true
isValidIdentifier('1_myVar'); // false
```

6. Write a function that accepts a string as an input, capitalizes the first letters of each word and returns the capitalized string. Use a regular expression to achieve the desired result.

```
const testStr = "My name is John Smith. I am 27.";
capitalize(testStr); // "My Name Is John Smith. I Am 27."
```

7. Write a simple password validation function that accepts a string as an input and returns either true (valid) or false (invalid). The password is considered to be valid if it satisfies all of the following requirements:

- there is at least 1 uppercase letter.
 - there is at least 1 lowercase letter.
 - there is at least 1 number.
 - needs to be at least 8 characters long.
- It is invalid otherwise. Use a regular expression to validate the password.

```
isValidPassword('agent007'); // false (no uppercase letter)
isValidPassword('AGENT007'); // false (no lowercase letter)
isValidPassword('Agent000'); // false (no numbers)
isValidPassword('Age_007'); // false (too short)
isValidPassword('Agent007'); // true
```

8. Write a function `bubbleSort` (using bubbleSort algorithm) which takes an array of integers as input and returns an array of these integers in sorted order from least to greatest.

NOTE: ***bubbleSort*** should not use the built-in `.sort()` method.

```
console.log(bubbleSort([7,5,2,4,3,9])); //[2, 3, 4, 5, 7, 9]
```

9. Write a function `sortByItem` which accepts object with two keys `{item, array}` and return given array of objects sorted by provided item name

```
const inventory = [
  { name: 'milk', brand: 'happyCow', price: 2.1, },
  { name: 'chocolate', brand: 'milka', price: 3, },
  { name: 'beer', brand: 'hineken', price: 2.2, },
  { name: 'soda', brand: 'coca-cola', price: 1, },
];

console.log(sortByItem({item: 'name', array: inventory})); // will return
[
  { "name": "beer", "brand": "hineken", "price": 2.2 },
  { "name": "chocolate", "brand": "milka", "price": 3 },
  { "name": "milk", "brand": "happyCow", "price": 2.1 },
  { "name": "soda", "brand": "coca-cola", "price": 1 }
]
```

RESTRICTIONS

- Using any external libraries is forbidden.

BEFORE SUBMIT

- Remove all unnecessary files that you might have included by mistake.
- Verify that the names of the folders and files meet the requirements.
- Verify that all functionality is implemented according to the requirements.
- Make sure you code is well-formatted and validated via validator (w3org Markup Validation Service).
- Add comments only if the code is difficult to understand.
- Make sure there are no errors/warnings in the browser console; fix them if any.
- Run the linter and fix all the 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 "FL-19" into **main** 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/RegExp
- <https://javascript.info/date>
- <https://javascript.info/regular-expressions>
- <https://regex101.com/>
- https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/sort
- https://en.wikipedia.org/wiki/Bubble_sort