

ES Next

DEADLINE: 05/02/2022

FOLDER STRUCTURE

```
FL18_HW6/ *
├── task
├── homework/ *
│   ├── js/ *
│   │   ├── tasks.js *
│   ├── .eslintrc.js *
│   └── index.html *
```

* - required

TASK

Note: Under the new operators we consider all operators from ECMA Script 6 and higher.

1. Create a function which returns max **even** element in array from array of strings
 - Use Math.max(), reduce and other new operators.

```
const arr = ['1', '3', '4', '2', '5'];
console.log(getMaxEvenElement(arr));
4
```

2. Write a code which swap variables values without a temporary variable
 - Please, use destructuring assignment.

```
15
16   let a = 3;
17   let b = 5;
18   // your code
19
20   console.log(a); //5
21   console.log(b); //3
22
```

3. Create a function which simply return value when there is some defined value passed and empty text string '-' when it is not defined .

- Check ES2020 operators, one of them can help here.

```
>
console.log(getValue(0));
console.log(getValue(4));
console.log(getValue("someText"));
console.log(getValue(null));
console.log(getValue(undefined));
```

0

4

someText

-

-

4. Create a function which return objects from array of arrays.

```
const arrayOfArrays = [
  ["name", "dan"],
  ["age", "21"],
  ["city", "lviv"],
];

console.log(getObjFromArray(arrayOfArrays));
▶ {name: "dan", age: "21", city: "lviv"}
```

5. Create function to enhance element with unique id. Here is how function should be invoked:

- Use Symbol() as a unique identifier

```
addUniqueId({name: 123})
```

- As a result, returned object should have extra property - unique identifier. Returned object should be copy of the passed parameter. Original object should not contain this id.

```
const obj1 = { name: "nick" };

console.log(addUniqueId(obj1));
console.log(addUniqueId({ name: "buffy" }));

console.log(Object.keys(obj1).includes('id'));
▶ {name: "nick", id: Symbol()}
▶ {name: "buffy", id: Symbol()}
false
```



6. Write a function which regroups object properties
- Destruct old object and construct new

```
const oldObj = {
  name: "willow",
  details: { id: 1, age: 47, university: "LNU" },
};

console.log(getRegroupedObject(oldObj));
```

▼ {university: "LNU", user: {...}} ⓘ

university: "LNU"

▼ user:

age: 47

firstName: "willow"

id: 1

▶ __proto__: Object

▶ __proto__: Object

undefined

7. Create a function which finds unique elements in array
- Use one of the new data types

```
const arr = [2, 3, 4, 2, 4, "a", "c", "a"];
console.log(getArrayWithUniqueElements(arr));
```

▶ (5) [2, 3, 4, "a", "c"]

undefined

8. Create a function which masks phone number, leaves only last 4 digits
- Use padStart

```
const phoneNumber = "0123456789";
console.log(hideNumber(phoneNumber));
```

*****6789

9. Create function which has all parameters always required. If they are not - throw error.
- Use default parameters feature and assign a function to it

```

    ,
    console.log(add(2, 3));
    console.log(add(2));
  }
  5
  ✖ ▶ Uncaught Error: b is required
    at required (<anonymous>:2:9)
    at add (<anonymous>:5:37)
    at <anonymous>:10:13
  > |

```

10. Use generator function to create an iterable sequence of values:

```

const generatorObject = generateIterableSequence();

for (let value of generatorObject) {
  console.log(value);
}
I
love
EPAM

```

RESTRICTIONS

- Adding **task/** folder is forbidden. Do not push it to repository. (Only **homework/** folder should be pushed)
- Do not use any external libraries

BEFORE SUBMIT

- Code should be clean, without comments, readable, and tested
- Make sure your GitLab folder structure meets folder structure from this document (without **task** folder)
- 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 **FL18_HW6** folder without **task** folder should be uploaded to GitLab repository "**FL-18**" into **main** branch.