

# Data types

**DEADLINE:** 13/12/2021

## FOLDER STRUCTURE

```
FL18_HW4/ [*]
├── task [*]
├── homework/ [*]
│   ├── index.html [*]
│   └── app.js [*]
```

[\*] - required

[\*\*] - restricted (no need to submit)

## TASK

Write all tasks inside app.js file.

1. Write a JavaScript function that reverse an integer number.

```
reverseNumber(12345) // returns 54321
reverseNumber(-56789) // returns -98765
```

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

```
forEach([2,5,8], function(el) { console.log(el) }) // logs to
console: 2 5 8
```

3. Write function, which returns transformed array based on function, which is passed as a parameter. Reuse function from task 2.

```
map([2, 5, 8], function(el) { return el + 3; }) // returns [5, 8, 11]
map([1, 2, 3, 4, 5], function (el) { return el * 2; }) // returns
[2, 4, 6, 8, 10]
```

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

```
filter([2, 5, 1, 3, 8, 6], function(el) { return el > 3 }) //
returns [5, 8, 6]
filter([1, 4, 6, 7, 8, 10], function(el) { return el % 2 === 0 })
//returns [4, 6, 8, 10]
```



5. Write function, which returns array of names of people, who are over 18 and their favorite fruit is apple. Reuse functions from task 3 and 4.

```
//See input data example in CODE section  
getAdultAppleLovers(data) // returns ['Stein']
```

6. Write function, which returns array of keys of an object.

```
getKeys({keyOne: 1, keyTwo: 2, keyThree: 3}) // returns ["keyOne",  
"keyTwo", "keyThree"]
```

7. Write function, which returns array of values of an object.

```
getValues({keyOne: 1, keyTwo: 2, keyThree: 3}) // returns [1, 2, 3]
```

8. Write function, which returns formatted date.

```
showFormattedDate(new Date('2018-08-27T01:10:00')) // returns 'It is  
27 of Aug, 2018'  
// every month should be showed as 3 letters (e.g. Feb, Apr or Dec)
```

## RESTRICTIONS

- Using any built-in array or object methods(besides push)
- Using any external libraries

## BEFORE SUBMIT

- Code should be clean, readable, and tested

## SUBMIT

- The folder should be uploaded to gitlab repository “**FL-18**” into **main** branch

## CODE

Input data for task 5.

```
[  
  {  
    "_id": "5b5e3168c6bf40f2c1235cd6",  
    "index": 0,  
    "age": 39,  
  }  
]
```



```
    "eyeColor": "green",
    "name": "Stein",
    "favoriteFruit": "apple"
  },
  {
    "_id": "5b5e3168e328c0d72e4f27d8",
    "index": 1,
    "age": 38,
    "eyeColor": "blue",
    "name": "Cortez",
    "favoriteFruit": "strawberry"
  },
  {
    "_id": "5b5e3168cc79132b631c666a",
    "index": 2,
    "age": 2,
    "eyeColor": "blue",
    "name": "Suzette",
    "favoriteFruit": "apple"
  },
  {
    "_id": "5b5e31682093adcc6cd0dde5",
    "index": 3,
    "age": 17,
    "eyeColor": "green",
    "name": "Weiss",
    "favoriteFruit": "banana"
  }
]
```

## 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/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/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/Object/prototype)