# Data types

## FOLDER STRUCTURE

FL11_HW9/*
    └─ homework/*
        └─ index.html*
        └─ index.js*
        └─ .eslintrc.js*

\* - required

## TASK

Write all tasks inside `index.js` file

0. Write function, which returns array of numbers from string parameter.

```
getNumbers('string'); // returns []
getNumbers('n1um3ber95'); // returns [1,3,9,5]
```

1. Write a function that could receive different amount of parameters (n1, n2, ...n) that have different data types (string, number, boolean, etc.) and returns an object where keys are data types of received parameters and value are their count. For example:

```
findTypes('number') // returns {"string":1}
findTypes(null, 5, "hello") // returns {"object":1, "number":1, "string":1}
```

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

```
executeforEach([1,2,3], function(el) { console.log(el) }) // logs 1 2 3
```

3. Write function, which returns transformed array based on function, which passed as a parameter. 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 > 3 }) // returns [5, 8]
```

5. Write function, which returns formatted date.

```
showFormattedDate(new Date('2019-01-27T01:10:00'))
// returns 'Date: Jan 27 2019'
// every month should be showed as 3 letters (e.g. Feb, Apr or Dec)
```

6. Write function, which returns Boolean value, is received string parameter can be converted to valid date.

```
canConvertToDate('2016-13-18T00:00:00') // false
canConvertToDate('2016-03-18T00:00:00') // true
```

7. Write function, which returns difference between two dates in days
```
daysBetween(new Date('2016-03-18T00:00:00'), new Date('2016-04-
19T00:00:00'))  // 32
```

8. Write function, which returns amount of people, who are over 18. Reuse function from task 4,7

```
//See input data example in CODE section
getAmountOfAdultPeople(data) // returns 3;
```

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

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

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

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

## PLEASE NOTE

Allowed to use  Math.round();

## RESTRICTIONS

- Using built–in array or object methods(besides push, hasOwnProperty and date methods)
- Using any external libraries

## CODE

Input data for task 8 and 9.
```
[
  {
    "_id": "5b5e3168c6bf40f2c1235cd6",
    "index": 0,
    "birthday ": '2016-03-18T00:00:00',
    "eyeColor": "green",
    "name": "Stein",
    "favoriteFruit": "apple"
  },
```

```json
{
  "_id": "5b5e3168e328c0d72e4f27d8",
  "index": 1,
  "birthday ": '1991-02-11T00:00:00',
  "eyeColor": "blue",
  "name": "Cortez",
  "favoriteFruit": "strawberry"
},
{
  "_id": "5b5e3168cc79132b631c666a",
  "index": 2,
  "birthday ": '1984-04-17T00:00:00',
  "eyeColor": "blue",
  "name": "Suzette",
  "favoriteFruit": "apple"
},
{
  "_id": "5b5e31682093adcc6cd0dde5",
  "index": 3,
  "birthday ": '1994-04-17T00:00:00',
  "eyeColor": "green",
  "name": "George",
  "favoriteFruit": "banana"
}
]
```

## 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 '**FL11**' 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