Help to complete the tasks of this exercise can be found on the chapter 1 ” Values, types and operators”, chapter 2 “Program structure” and chapter 4 “Data Structures: Objects and Arrays” of our course book “Eloquent JavaScript” (3rd edition) by Marijin Haverbeke. The aims of the exercise are to become familiar with JavaScript basics, especially values, types, arrays and operators. The student will also learn what kinds of structures a JavaScript program has, and how the flow of the program is controlled.

Embed your theory answers, drawings, codes, and screenshots directly into this document. Always immediately after the relevant question. Return the document into your return box in itsLearning by the deadline.

It’s also recommendable to use Internet sources to supplement the information provided by the course book. Especially the * marked task can require that.

The maximum number of points you can earn from this exercise is 10 + 1 = 11.

Tasks:

1. Answer the questions? (4 * 0,25 = 1 point):

a. What is the result and why?

`1 == '1'`

```
2    console.log(1 == "1");
```

True

Javascript converts the latter one into int due to double equals.

b. What is the result and why?

`1 === '1'`

```
console.log(1 === "1");
```

False

Javascript triple equals compares without converting type to same.

c. What is the result and why?

`typeof "Kissa";`

Returns string, since Kissa is within " " and it is string

```
5    console.log(typeof "Kissa");
```

d. What is the value of the variable `currentPort` and why?

`var port = 3001; var currentPort = port || 3000;`

It is 3001 since it is first one. || is or gate, and it needs only one true, so it continues.

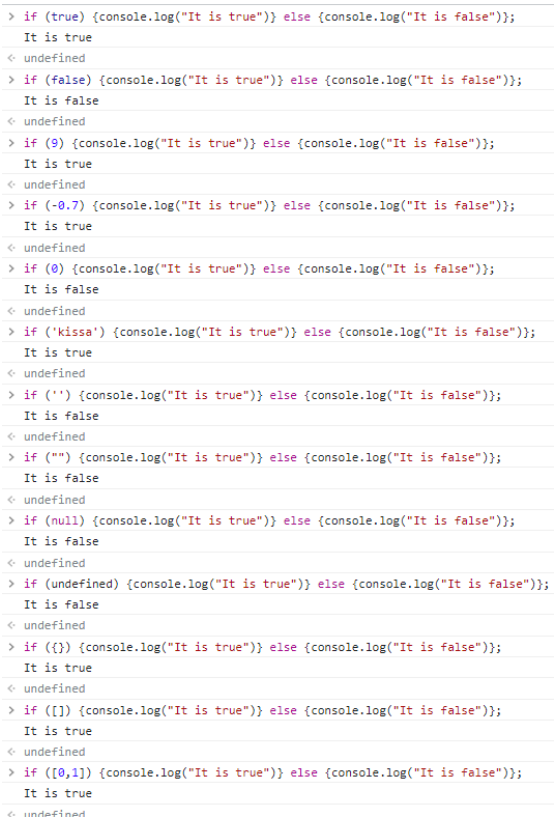
```
var port = 3001; var currentPort = port || 3000;
```

2. JavaScript boolean values. (2 * 0,5 = 1 point)

a. Are the following values true or false in JavaScript? Use a browser console and program an if-else statement to find the answers.

```
true, false, 9, -0.7, 0, 'kissa', '', "", null, undefined, {}, [], [0,1]
```

b. Why?



```
> if (true) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if (false) {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if (9) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if (-0.7) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if (0) {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if ('kissa') {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if ('') {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if ("") {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if (null) {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if (undefined) {console.log("It is true")} else {console.log("It is false")};
  It is false
< undefined
> if ({}) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if ([]) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
> if ([0,1]) {console.log("It is true")} else {console.log("It is false")};
  It is true
< undefined
```

If number that is not 0 it gives truth.

Null, undefined and “” gives false since it has no value.

Objects are considered true, whether there is object or not

3. Strings. (4 * 0,5 = 2 points)

In JavaScript, you can use single quotes, double quotes, or backticks to mark strings.

a. Is there any differences between these differently marked strings?

In Javascript backtick is little bit different, it is able to get “embedded” elements and for example calculate them

b. Give code clips highlighting the possible differences.

```
7   console.log(" it's half of 100 is ${100 / 2}");
8   console.log(' it's half of 100 is ${100 / 2}');
9   console.log(` it's half of 100 is ${100 / 2}`);
```

c. Give reasons to use the character \ inside strings.

You use it to format the text

d. Give examples of different String methods. Where did you find that information?

toLowerCase(), toUpperCase(), toString(). I found this information from internet.

4. Variables and constants. (1 point)

You can use the keywords `var` and `let` to define variables in JavaScript.

a. What is the difference between the keywords `var` and `let` when defining a variable? (0,5 points)

Var is global where let is “limited to the scope of a block statement”

b. What happens if you don’t remember use either of these keywords when defining a variable? (0,25 points)

Then the variable will be global

c. How do you define a constant in JavaScript? (0,25 points)

Using const.

5. Looping in JavaScript. (2 * 0,5 = 1 point):

Let's use the array `distances = [ 164, 526, 248, 12, 81, 181, 34].`

a. Use for loop to calculate the sum of the distances. (0,5 points)

```
let distanceSum=0;

for (let i=1; i <= distances.length-1; i++){
    distanceSum += distances[i];
}

console.log(distanceSum)
```

b. Sum the distances from the beginning of the array until the first distance with a value less than 100 is reached. Use while loop. (0,5 points)

```
let i=0

while (distances[i]>100){
    distanceSum+=distances[i];
    i++;
}
```

6. Considering JavaScript arrays. (4 * 0,25 = 1 point)

a. Can an array contain both numbers and objects at the same time in JavaScript?

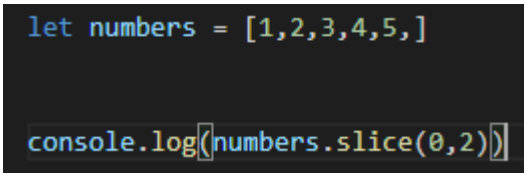
Yes.

b. Explain what it means to 1) modify an array in place or to 2) return a modified copy of an array. Give one example of a JavaScript method that modifies an array in place, and one example of a method that returns a modified copy.

array = [1,2,3,4,5]

array[0]=2

array = [2,2,3,4,5] for example.



c. What does it mean that a JavaScript array is mutable? *

It can be modified, changed values etc. immutable would be opposite.

d. You have a following code clip. How many arrays do you have in the memory at the end?

```
let array1 = [1,3,5];
let array2 = array1;
```

Two, array2 is just copy of array1

But both are own arrays

7. Working with JavaScript arrays (2 points)

Let's use the array `distances = [ 164, 526, 248, 12, 81, 181, 34 ]`.

a. Write a code clip that returns the length of the array. (0,5 points)

```
Front End Development > Lesson 1 > JS EX01T7.js > ...
1   let distances = [164, 526, 248, 12, 81, 181, 34];
2
3   console.log(distances.length)
```

b. Write a code clip that adds the distances `8`, `533` and `76` at the end of the array and in this order. Use one of the array methods. (0,5 points)

```
let distances = [164, 526, 248, 12, 81, 181, 34];
distances.push(8, 533, 76)
console.log(distances)
console.log(distances.length)
```

c. Write a code clip that removes the number `248` from the array in place. Use an array method. A tip: One of the methods to consider could be `splice`. (0,5 points)

```
> (10) [164, 526, 248, 12, 81, 181, 34, 8, 533, 76]
  10
> (9) [164, 526, 12, 81, 181, 34, 8, 533, 76]
```

```
5   distances.splice(2,1)
```

d. Clone the array `distances` to the variable `distances_duplicate`. Use ES6 way: The spread operator. * (0,5 points)

```
Front End Development > Lesson 1 > JS EX01T7.js > ...
1   let distances = [164, 526, 248, 12, 81, 181, 34];
2   distances.push(8, 533, 76)
3
4   distances.splice(2,1)
5
6   distances_duplicate=[...distances]
```

Please, search the Net to help.

8. Working with JavaScript Array methods filter, map and reduce (2 points) *

These JavaScript methods are heavily used in modern JavaScript applications.

Let's use the array `points = [ 64, 56, 48, 12, 81, 91, 34, 19, 95, 55 ]`.

a. Return a new array into a variable called `enough_points`. The new array contains all numbers of the original array `points` that are at least 40. Use the method `filter`. (0,5 points)

b. Return a new array into a variable called `grades`. The new array contains the grades that are calculated from the numbers of the original array `points`. Use the method `map`. The evaluation scale is the following: at least 40 points -> 1; 50 -> 2; 60 -> 3; 70 -> 4; 85 -> 5. Otherwise, the grade is 0. (0,5 points)

c. Calculate the average grade by chaining the methods map and reduce. Use the original array `points`. (1 point)

Please, search the Net to help.