LU01.A03 - myBubbleSort

Teilauftrag 1: Statisch

In zwei FOR-Schleifen wird ein unsortiertes Array eine feste Anzahl mal durchlaufen. Die Anzahl der Schleifendurchlaeufe ist abhaengig von der Anzahl unsortierter Array-Elemente.

scripts_m307 > lu01_script4fun > JS lu01_l03_myBubbleSort_v1.js > bubbleSortV1

```
1  var unsortedArrayV1 = [3, 55, 2, 8, 33, 199, 120, 100, 999, 53, 44, 54, 67, 22, 0]; // unsorted array of numbers
2  console.clear();
3
4  console.log("array bubbleSortV1 - Vorher: ", unsortedArrayV1);
5  console.log("function bubbleSortV1 - Nachher: " + bubbleSortV1(unsortedArrayV1));
6
7
8  /* ********************************************************************** */
9  /* Author: volkan.demir@bzz.ch, 02.03.23                                  */
10 /* Call: bubbleSortV1 (array)                                             */
11 /* Desc: Returns the array in a ascending or descending order.            */
12 /*       The algorithm is not smart, since it uses 2 for-loops with fixe rage.*/
13 /* ********************************************************************** */
14 function bubbleSortV1(myUnsArray) {
15     let arrLen = myUnsArray.length;
16     let myLocArray=myUnsArray;
17     let swapCnt = 0; // count of the necessary swap due to the wrong order of the numbers
18     let loopCnt = 0; // count of the necessary swap due to the wrong order of the numbers
19
20     for (var i=0; i<=arrLen; i++) {
21         for (var j=0; j<arrLen; j++) {
22             loopCnt++;
23             if (myLocArray[j]>myLocArray[j+1]) { // ascending = <; descending = >
24                 swapCnt++; // count swap
25               //  console.log("swap1:"+swapCnt);
26                 var tmp=myLocArray[j]; // change of two values of variables
27                 myLocArray[j]=myLocArray[j+1];
28                 myLocArray[j+1]=tmp;
29             }
30         }
31     }
32     console.log("loopCnt: "+loopCnt);
33     console.log("swapCnt: "+swapCnt);
34     return myLocArray;
35 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
    3,  55,   2,  8, 33, 199,
  120, 100, 999, 53, 44,  54,
   67,  22,   0
]
loopCnt: 240
swapCnt: 51
```

Teilauftrag 2: Flag

Variante 2 arbeitet intelligenter als die Variante 1. Mittels eines *Flag* (Ampel) wird bei jedem Durchlauf des Arrays ermittelt, ob die korrekte Sortierung vorliegt. Falls nicht wird ein weiterer Durchlauf gestartet.

```
var unsortedArrayV2 = [3, 55, 2, 8, 33, 199, 120, 100, 999, 53, 44, 54, 67, 22, 0]; // unsorted array of numbers
console.log("function bubbleSortV2: " + bubbleSortV2(unsortedArrayV2));
/* ********************************************************************** */
/* Author: volkan.demir@bzz.ch, 02.03.23                                  */
/* Call: bubbleSortV1 (array)                                             */
/* Desc: Returns the array in a ascending or descending order.            */
/*       The algorithm is smart, it knows, when sorting is finished.      */
/* ********************************************************************** */
function bubbleSortV2(myUnsArray) {
    var arrLen = myUnsArray.length;
    var myLocArray=myUnsArray;
    var finished=false; //flag decides whether another loop or not
    var swapCnt = 0; // count of the necessary swap due to the wrong order of the numbers
    var loopCnt = 0;

    while (!finished) {
        loopCnt++;
        finished = true; // reset of the finished-flag
        for (var i=0; i<arrLen; i++) {
            if (myLocArray[i]>myLocArray[i+1]) { // ascending = <; descending = >
                swapCnt++; // count swap
                finished=false; // have to loop another time
               // console.log("swap2:"+swapCnt);

                var tmp=myLocArray[i]; // change of two values of variables
                myLocArray[i]=myLocArray[i+1];
                myLocArray[i+1]=tmp;
            }
            //console.log(myLocArray);
        }
    }
    myLocArray.push(loopCnt) // Adding the swap at the end of the array
    myLocArray.push(swapCnt) // Adding the swap at the end of the array
    return myLocArray;
}
```

```
.EMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

tion bubbleSortV1: 0,2,3,8,22,33,44,53,54,55,67,100,120,199,999,225,51
tion bubbleSortV2: 0,2,3,8,22,33,44,53,54,55,67,100,120,199,999,15,51
```

Volkan Demir

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m307/learningunits/lu01/loesungen/02?rev=1773235269

Last update: **2026/03/11 14:21**