**Minimumkiválasztás**

const intArr = [3254, 546, 23426, 3476, 234, 256365, 23, 4];

function getMinElementFromAnArray(arr) {

    let min = arr[0];

    for (let i = 1; i < arr.length; i += 1) {

        if (arr[i] < min) {

            min = arr[i];

        }

    }

    console.log(min);

}

getMinElementFromAnArray(intArr);

getMinElementFromAnArray([345, 546, 234, 47]);

**Buborék rendezés**

'use strict';

function bubbleSort(arr) {

    let swapped;

    do {

        swapped = false;

        for (let i = 0; i < arr.length - 1; i += 1) {

            if (arr[i] > arr[i + 1]) {

                let temp = arr[i];

                arr[i] = arr[i + 1];

                arr[i + 1] = temp;

                swapped = true;

            }

        }

    } while (swapped)

    return arr;

}

const arr = [435, 567, 8, 33, 346, 67, 678, 235, 225];

console.log(bubbleSort(arr));

// function defultParam({ firstName = 'John', lastName = 'Doe' } = {}) {

// }

**function binarySearch(arr, search) {**

    let start = 0;

    let end = arr.length - 1;

    // let counter = 0;

    while (start <= end) {

        // counter += 1;

        let mid = Math.floor((start + end) / 2);

        if (arr[mid] === search) {

            return true;

        }

        else if (arr[mid] < search) {

            start = mid + 1;

        }

        else {

            end = mid - 1;

        }

    }

    // console.log('futások száma:', counter);

    return false;

}

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

// const float = [1.23, 2.34, 3.23234234, 4.43535];

console.log(binarySearch(numbers, 5));

***function linearSearch(arr, search) {***

    for (let i = 0; i < arr.length; i += 1) {

        if (arr[i] === search) {

            return 'igen';

        }

    }

    return 'nem';

}

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

console.log(linearSearch(numbers, 5));

***function bubbleSort(arr) {***

    let swapped;

    do {

        swapped = false;

        for (let i = 0; i < arr.length - 1; i += 1) {

            if (arr[i] > arr[i + 1]) {

                let temp = arr[i];

                arr[i] = arr[i + 1];

                arr[i + 1] = temp;

                swapped = true;

            }

        }

    } while (swapped)

    return arr;

}

**function getNthMinElement(arr, nth) {**

    const sortedArr = bubbleSort(arr);

    if (nth > arr.length) {

        return arr[arr.length - 1];

    }

    return sortedArr[nth - 1];

}

const sampleArray = [235, 546, 23, 5245, 52, 6345, 82];

console.log(getNthMinElement(sampleArray, 10));

const sample = [435, 567, 8, true, NaN, 'hello', 33, 'szia', 346, 67.10, 678, false, 235, 225];

const sample2 = [435, 567, 8, 435, 345, 677];

// Hogyan lehetne a bubbleSortAsc és bubbleSortDesc függvényeket

// egybevonni? (Paraéterként megadni, hog milyen sorendbe rendezzük)

function bubbleSortAsc(arr) {

  let swapped;

  do {

    swapped = false;

    for (let i = 0; i < arr.length - 1; i += 1) {

      if (arr[i] > arr[i + 1]) {

        const temp = arr[i];

        arr[i] = arr[i + 1];

        arr[i + 1] = temp;

        swapped = true;

      }

    }

  } while (swapped);

  return arr;

}

function bubbleSortDesc(arr) {

  let swapped;

  do {

    swapped = false;

    for (let i = 0; i < arr.length - 1; i += 1) {

      if (arr[i] < arr[i + 1]) {

        const temp = arr[i];

        arr[i] = arr[i + 1];

        arr[i + 1] = temp;

        swapped = true;

      }

    }

  } while (swapped);

  return arr;

}

// A két filter függvényt hogyan vonhatnánk egybe?

function filterByNumbers(arr) {

  const numbers = [];

  for (let i = 0; i < arr.length; i += 1) {

    if (typeof arr[i] === 'number') {

      numbers.push(arr[i]);

    }

  }

  return numbers;

}

function filterByNotNumbers(arr) {

  const notNumbers = [];

  for (let i = 0; i < arr.length; i += 1) {

    if (typeof arr[i] !== 'number') {

      notNumbers.push(arr[i]);

    }

  }

  return notNumbers;

}

function customSorting(arr) {

  const numbers = filterByNumbers(arr);

  const notNumbers = filterByNotNumbers(arr);

  const sortedArr = bubbleSortDesc(numbers);

  return sortedArr.concat(notNumbers);

}

function customSorting2(arr) {

  const numbers = [];

  const notNumbers = [];

  for (let i = 0; i < arr.length; i += 1) {

    if (typeof arr[i] !== 'number' || Number.isNaN(arr[i])) {

      notNumbers.push(arr[i]);

    } else {

      numbers.push(arr[i]);

    }

  }

  const sortedArray = bubbleSortDesc(numbers);

  return sortedArray.concat(notNumbers);

}

console.log(customSorting(sample));

console.log(customSorting2(sample));

function custom13(arr) {

  const inputNumber = parseInt(prompt('Adj meg egy számot'), 10);

  arr.push(inputNumber);

  const sortedArray = bubbleSortAsc(arr);

  return sortedArray;

}

function getIndex(arr, value) {

  for (let i = 0; i < arr.length; i += 1) {

    if (arr[i] > value) {

      return i;

    }

  }

}

function custom13v2(arr) {

  const sortedArray = bubbleSortAsc(arr);

  const inputNumber = parseInt(prompt('Adj meg egy számot'), 10);

  if (!Number.isNaN(inputNumber)) {

    const index = getIndex(sortedArray, inputNumber);

    sortedArray.splice(index, 0, inputNumber);

  }

  return sortedArray;

}

console.log(custom13(sample2));

console.log(custom13v2(sample2));