**JavaScript Functions**

function getAverage (a, b) {

    let average: number = (a+b)/2;

    console.log(average);

}

getAverage(5,7);

If we want to save to a variable:

function getAverage (a, b) {

    let average: number = (a+b)/2;

console.log(average);

return average;

}

let myResult: number = getAverage(7,5);

**Function expressions**

//normal function statement

function sayHi(){

console.log('hi');

}

sayHi();

//function expression

var sayBye = function(){

console.log('Bye');

}

sayBye();

//Function in function

function callFunction(fun){

fun();

}

var sayBye = function(){

console.log('Bye');

}

callFunction(sayBye);

**What is an array?**

An array is a list of values.

let myNumbers: number[]=[];

myNumbers=[1,2,3,4];

**Creating array using constructor**

We can also create array object using constructor of Array class as:

let myNumbers1: number[]=new Array(5);

console.log(myNumbers1);

//result: [ <5 empty items> ]

let myNumbers: number[]=new Array(1,2,3,4,5);

console.log(myNumbers);

// result: [ 1, 2, 3, 4, 5 ]

**Declareing elements**:

var myNumbers:Array<number>=[1,2,3,4,5];

console.log(myNumbers);

console.log(myNumbers[3]); //prints out the 4TH element (myNumbers[0] is the 1st and so on)

// result: [ 1, 2, 3, 4, 5 ]

4

**Adding elements**

var myNumbers:Array<number>=[1,2,3,4,5];

myNumbers.push(133); //adds another number to the array

myNumbers[6]=200; //adds another number to the array

myNumbers[0]=0; //updates value to the first tag of the array

console.log(myNumbers);

//result:

//[ 0, 2, 3, 4, 5, 133, 200 ]

**Removing elements**

var myNumbers:Array<number>=[1,2,3,4,5];

myNumbers.pop(); //removes the last element in the array

console.log(myNumbers);

//result:

//[ 1, 2, 3, 4 ]

myNumbers.splice(0, 1); //removes the first element

console.log(myNumbers);

//result:

//[ 2, 3, 4 ]

myNumbers.splice(1, 2); //removes 2 elements starting from the second element of the array

console.log(myNumbers);

//result:

//[ 2 ]

**Arrays and Tuples**

Tuples are for specify what type of value can be in an array.

let myStrings1: string[]=["Hello", "world"]; //just declaring the array

let myStrings2: Array<string> = ["Hello", "world"]; //just declaring the array

let myAnythings: any[] = ["Hello", 10, true]; //just declaring the array

let myTuple: [string, number] = ["Hi", 10]; //makes an array which can only contain strings and numbers

**Sort function**

let myAnythings: any[] = ["Hello", 10, true]; //just declaring the array

myAnythings.sort(); // sorts the array from lower to high

console.log(myAnythings);

//result:

//[ 10, 'Hello', true ]

**Reverse function**

let myAnythings: any[] = ["Hello", 10, true]; //just declaring the array

myAnythings.reverse(); // reverses the order of the elements

console.log(myAnythings);

//result:

//[ true, 10, 'Hello' ]

**Functions**

[**https://www.typescriptlang.org/docs/handbook/functions.html**](https://www.typescriptlang.org/docs/handbook/functions.html)

**Variable Scope (until const declarations):**

[**https://www.typescriptlang.org/docs/handbook/variable-declarations.html**](https://www.typescriptlang.org/docs/handbook/variable-declarations.html)

**Material Review**

* purpose of functions
  + reusability
  + reducing code duplication
  + abstraction, parameterization
* function expressions
  + function as variable
  + function as parameter
  + anonymus function
* global functions:
  + alert
  + console.log
  + parseInt
  + parseFloat
* return statement
* arguments
* variable scope
* spread syntax
* array methods:
  + push
  + pop
  + shift
  + unshift
  + indexOf -> egy elem helyét (indexét) adja meg egy tömbben, eredmény: szám.
  + slice
  + splice
  + concat
  + forEach ->
  + filter
  + map
  + every
  + some