**Practical:-1**

**Aim:-Create a program that declares and initializes variables of different data types (string, number, boolean) and displays their values. Write a function that takes two numbers as parameters and returns their sum.**

**Code:-**

const name = "Nishar khorajiya";

const num = 19;

const constant = 5.9;

const bool = true;

console.log("name:", name);

console.log("num:", num);

console.log("constant:", constant);

console.log("boolean:", bool);

//function for sum

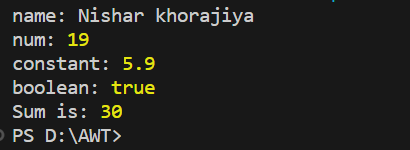
function sum(num1, num2) {

    return num1 + num2;

}

console.log("Sum is:", sum(10, 20));

**output:-**

****

**Practical:-2**

**Aim:-** **Create an array of numbers and perform the following operations: => Find the length of the array. => Access and display specific elements using indexing. =>Use array methods like push() ,pop(), shift(), unshift(), join(), delete(),concate(),flat(),splice() and slice() to modify the array. Create an object representing a person with properties like name, age, and gender. Implement a function that displays the person's details.**

**Code:-**

a = [1, 2, "3", 4, "5", 6];

console.log("ap1 = ", a);

console.log("size of ap1 is = ", a.length);

console.log("ap1[1] = ", a[1]);

console.log("ap1[4] = ", a[4]);

a.push(10);

console.log("ap1 after push() ", a);

a.pop();

console.log("ap1 after pop()", a);

// concate

const a1 = [1, 2, 3];

const a2 = [4, 5, 6];

const a3 = a1.concat(a2);

console.log("a1 = ", a1);

console.log("a2 = ", a2);

console.log("concate ap1ay a3 = ", a3);

// flat

const a4 = [0, 1, 2, [3, 4]];

console.log(a4.flat());

const a5 = [0, 1, 2, [[[3, 4]]]];

console.log(a5.flat(2));

// slice

console.log("slice");

const a6 = [1, 2, "3", 4.67, 5, 0.896, "one", 8, "tata"];

console.log(a);

console.log(a6.slice(1, 4));

// splice

console.log("splice");

const a9 = [0, 3, "darsh", 9.08];

a9.splice("splice => ", 0, "one");

console.log(a9);

// unshift

console.log("unshift")

const p1 = ["Banana", "Orange", "Apple", "Mango"];

p1.unshift("Lemon");

console.log("p1 = ", p1);

//shift

console.log("shift")

console.log(p1.shift())

// join

console.log("join");

const ele = ["1", "2", "3", "4", "5"];

console.log(ele.join(", "));

// delete

console.log("delete");

let a7=[1,2,3,4,5]

console.log("before:",a7)

delete a7[2]

console.log("after:",a7)

// object

console.log("object");

let data = {

  name: "nishar",

  age: 19,

  gender: "Male",

};

function detail(item) {

  console.log(item.name);

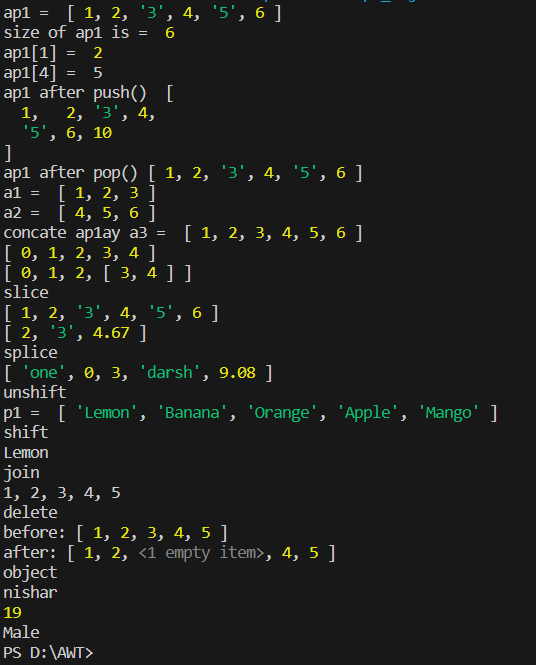
  console.log(item.age);

  console.log(item.gender);

}

detail(data);

**output:-**

****

**Practical:-3**

**Aim:- Implement following features of ECMASCRIPT 6.**

**• The let keyword • The const keyword • Arrow Functions • The (Spread Of) ... Operator • For/of • Map Objects • Set Objects • Classes • Promises • Symbol • Default Parameters • Function Rest Parameter**

**Code:-**

//let keyword

let x = 1;

if (true) {

    let x = 2;

    console.log(x);

}

console.log(x);

//const keyword

const Pi = 3.14;

console.log("const keyword:", Pi);

//arrow function

const add = (a1, a2) => a1 + a2;

console.log("arrow function for sum:", add(5, 3))

//spread(...) operater

const arr = [1, 2, 3];

const copyArr = [...arr];

function sum(...numbers) {

    return numbers.reduce((acc, curr) => acc + curr, 0);

}

console.log("sum using spread operater:", sum(1, 2, 3));

//for-of loop

const arr2 = [1, 2, 3];

for (const num of arr2) {

    console.log(num);

}

//Map

const m = new Map();

m.set("name", "nishar");

m.set("age", 19);

console.log("Map:", m.get("name"));

//set

const s = new Set();

s.add(1);

s.add(2);

s.add(1);

console.log("set:", s);

//classes

class person {

    constructor(name, age) {

        this.name = name;

        this.age = age;

    }

    get detail() {

        console.log(`my name is ${this.name} and I am ${this.age} years old.`);

    }

}

const nishar = new person("nishar", 19);

nishar.detail;

//promises

function fetchData() {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

            const data = "my name is nishar";

            resolve(data);

        }, 2000);

    });

}

fetchData()

    .then((data) => console.log(data))

    .catch((err) => console.error(err));

//Symbol

const sym = Symbol("mySymbol");

const obj = {

    [sym]: "Symbol value",

};

console.log(obj[sym]);

//Default perameter

function defaultperam(collegename = "charusat university") {

    console.log(`My collegeName is(default peram):  ${collegename}`);

}

defaultperam();

//Function Rest Parameter

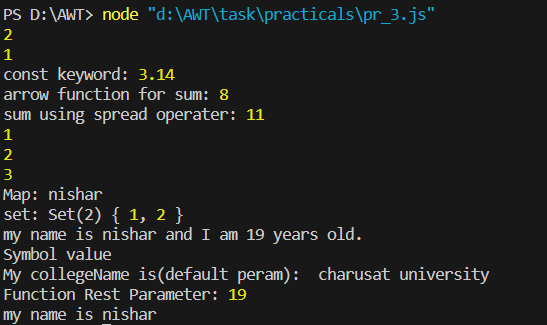
function sum(...numbers) {

    return numbers.reduce((accumulator, currentValue) => accumulator + currentValue, 5);

}

console.log("Function Rest Parameter:", sum(2,3,4,5));

**output:-**

****

**Practical:-4**

**Aim** **:-Write a function that calculates the factorial of a given number using recursion. Create a nested function that performs a specific task and invoke it within another function.**

**(NOTE: Implement the concept of variable scope in functions by declaring variables with different scopes (global, local) and accessing them).**

**Code:-**

function factorial(num) {

    function Fact(n) {

        if (n === 0 || n == 1) {

            return 1;

        } else {

            return n \* Fact(n - 1);

        }

    }

    return Fact(num);

}

let globalVar = "global var";

function main4() {

    let localVar = "local var";

    console.log("global variable:", globalVar);

    console.log("local variable:", localVar);

    let num = 5;

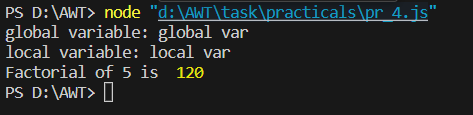
    let result = factorial(num);

    console.log(`Factorial of ${num} is `, result);

}

main4();

**output:-**

****