// console.log(a);

// const a = 5;.

// let a = 5

// a = 10

// console.log(a)

// const name = {

// names : ["abi","rahul"],

// hello : function(){

// console.log("hello");

// }

// }

// console.log(name.names[0]);

// let student1 = new Object();

// student1.name = function (){

// console.log("ram");

// }

// student1.name();

// console.log(student1.name)

// let a = 15;

// if(a%3 == 0 && a%5 == 0){

// console.log("hi and hello")

// }

// else if(a%3 == 0){

// console.log("hi")

// }

// else{

// console.log("hello")

// }

// const input = [1, 2, 3];

// input.forEach(value => {

// switch (value) {

// case 1:

// console.log("Matched case for 1");

// break;

// case 2:

// console.log("Matched case for 2");

// break;

// case 3:

// console.log("Matched case for 3");

// break;

// default:

// console.log("No match");

// }

// });

// const input = [1, 2, 3];

// switch (input) {

// case [1, 2, 3]: // Different array instance in memory

// console.log("Matched!");

// break;

// default:

// console.log("No match");

// }

// if(false)

// console.log('welcome')

// console.log('hello')

// let a = 1;

// for (console.log('a'); a <= 3; console.log('b')) {

// console.log('c');

// a++;

// }

// initialization -> { condition -> statement -> incr/dec }->loop

// let a = 1

// for( let a = 1;a<100;a++)

// console.log("hi")

// console.log(a);

// while(){

// console.log("hi")

// }

//without return type and non paramaterized

// function js(){

// console.log("hello")

// }

// js()

// function name(a,b){

// return a+b

// }

// console.log(name(7,6))

// const a = (x,y) => {

// console.log("hello")

// return x+y

// }

// console.log(a(7,5))

// let b = () => "hi"

// console.log(b())

//spread operator

// let a = [1,2,3]

// let b = [...a]

// b[2] = 10

// console.log(b)

// console.log(a)

// let a = [1,2,3]

// let b = a

// b[2] = 10

// console.log(b)

// console.log(a)

// let a = 10

// let b = a

// b = 20

// console.log(b)

// console.log(a)

// let a = 10

// function x(){

// a = 20

// console.log(a)

// }

// x()

// console.log(a)

// let a = [1,2,3,4]

// let b = [5,6,7,8]

// let c = [...a,...b]

// console.log(c)

//spread

// function spreadexample(values){

// console.log(values)

// }

// spreadexample(...b)

// function spreadexample(...values){

// console.log(values)

// }

// spreadexample(...b)

//rest

// function spreadexample(x,...values){

// console.log(values)

// console.log(x)

// }

// spreadexample(1,2,3,4,5)

// function spreadexample(...values,x){

// console.log(values)

// console.log(x)

// }

// spreadexample(1,2,3,4,5)

// spread operator -> expand(spread) the values

// rest operator -> combine the values in an array or object

//in array, there was an contiguous memory location. so if we reassingn the value, it stores only address

// a = [1000,1004,1008,1012]

// a = [1,2,3,4]

// b = [1000,1004,1008,1012]

// const obj = {

// name : "ram",

// age : 18,

// dob : "28-02-2005"

// }

// const b = {...obj}

// console.log(b)

//destructuring

// let a = [12,3,4,5,6]

// let [b,c,d,e,f] = a

// console.log(a);

// let a = [12,3,4,5,6]

// let [b,c,d,...y] = a

// console.log(b);

// const obj = {

// name : "ram",

// age : 18,

// dob : "28-02-2005"

// }

// let {...d} = obj;

// console.log(obj)

// console.log(d)

// let {name,age,dob} = obj

// console.log(name)

// console.log(age)

// console.log(dob)

//callback

//passing parameter as function

// function no1(callback) {

// setTimeout(() => {

// console.log("dog");

// callback()

// },1000)

// }

// function no2(){

// console.log("cat")

// }

// // no1()

// // no2()

// no1(no2) //used to maintain the order {dog , cat}

// function attendance(callback){

// setTimeout(() => {

// console.log("present");

// callback()

// },1000)

// }