JAVASCRIPT DOCUMENTATION

As JavaScript is dynamically typed language, there is no need to specify datatype first.

3 methods to declare a variable;

1. var
2. let
3. const

Eg :

var a=10;

let b=13.5;

const c=15;

(If you are using const then further you won't be able to change the value of the variable)

To print anything in JS we use :

console.log("Hi"); OR

process.stdout.write("Hi");

STRINGS IN JS

let s1 = "CSIKJSCE";

FUNCTIONS IN JS

function name(parameters)

{

Statements

Return value

}

eg:

function test(data)

{

return data+1;

}

(You can also create anonymous functions in JS)

var a = function(data){

return data+1;

}

(If you function is one liner then you can also use arrow functions)

let b = (data) =>data+1; //(This is same as above anonymous function)

Taking input from User in JS

(We use process.stdin library in JS)

```
process.stdin.resume();
process.stdin.on(
        “data”,
        function(input){
                process.stdout.write(“Hi”)
        }
);
```

(So before the execution of function , by default the JS doesn’t allow you to use process.stdin, so to use it we write resume function in the first line.

stdin.on takes two arguments first being the way to get data, we want to read data from terminal so we use “data” and the second one is a function which reads data.

Inside the function we just prints the data using stdout.write function**.(IF YOU ARE STILL CONFUSED DON’T WORRY HACKKERANK HAS THIS SETUP BY DEFAULT FOR YOU)**

LOOPS IN JS

Major two loops in JS

1] While loop

2] For Loop

Syntax of while loop:

```
while(condition)
{
        statements;
        increment/decrement;
}
```

Eg:

Program to print numbers from 1 to 5

```
let i=1;
while(i<=5)
{
        console.log(i)
        i++;
}
```

FOR LOOP

Syntax:

```
for(initialization;condition;increment/decrement)
{
    Statements
}
```

Example:

```
for(let i=1;i<=5;i++)
{
    console.log(i);
}
```

ARRAYS IN JS

Syntax:

let cars = [“tata”,”MG”,”BMW”,5]

(In JS it is not necessary to have similar type elements in an array like we can have an array which contains numbers and strings)

To access array elements we use indexing:

Printing cars[0] will print tata, similarly cars[3] == 5;

To get length we can do console.log(cars.length);

To push an element in an array we use :

cars.push(“Audi”);