datatypes:-
1.primitive
2.non-primitive

operators:-
**********
- operators are special symbols whic used to perform operation on operands.
ex :-  2(operand1) + 2(operand2)

1.unary
2.binary
3.ternary (operand1 ? operand2 :operand3)

1.1 arithmatic
1.2 assignment
1.3 relational
1.4 logical

1.1 arithmatic:- which performs operations such as,
	1.additional (concatenation :- joining of two or more string)
	2.substraction
	3.multiplication
	4.division
	5.modulus
	6.exponential
	7.increment(++)
		- pre-increment(++variable)
		- post-increment(variable++)
	8.decrement(--)

2.assignment operator:-
ex:- =
let a = 10;
additional assignment(+=)
substraction assignment(-=)
multiplication assignment(*=)
division assignment(/=)
mofulus assignment(%=)
exponential assignment(**=)

3.relational operator:-
- == , === , != , !== , > , >= , < , <=

- return type of relational operaots boolean values.

== - only checks value not a datatypes
=== - checks both data type and values.
!= -

4.logical operators
&& , || , !

```
&& - both LHS & RHS true
|| - anyone should be true
! - if true , it wil make false and vice-verse.

decision making statements:-
***************************
if
if-else
else-if ladder
switch
ternary condition

if:-
- when only ione condition is there then we can use if.
- if block will work only for true statement.
if()
{

}

if-else:-
- if is a true statement
- else is a flase statment

looping statments:-
- iterating the block of code multiple times.
- for loop:-
-minimum iteration count is "0".
    synatx: -  for(intitialization ; condition ; updation)
                    {//statements}
- while loop
-minimum iteration count is "0".
syntax:- while(condition){//statment}
- do-while loop
-minimum iteration count is "1".
-sytax :- do{//statments}
            while(condition)

which loops are entry controlled loops and exit controled loops?
entry controlled loops - for loop , while loop
exit controled loop - do-while loop

functions:-
************
- functions is a building block of javascript.
- function is a block of code which is used to execute specific task.
- function is a non-primitive data type.
- functins can be reusable.

synatx :-  function function_name(optional parameters,a ){ // function
delclaration
                                //statements
                                }
                        function_name(arguments);
```

types of function:-
**********
1.named function:- function is having a name and invoked(calling) with the same name.
2.anonymus function:- a function which doesnot have a name for it .
- it cannot be executed by itself.
3.function expression:- an anonymus function body is assigned as a value to the variable
and same function will be invoked with the same vairable.
4.IIFE(immidiate invoke function expression):-
- once function created immediatley it will called and executed.
- it can be invoked only ont time.
5.Arrow function :- it is introduced in es6 feature.
- it concise the normal function syntax.
- ()=>{};
- arrow function is similar to anonymus function.

6.Higher order function :-
-higher order function is a function which accepts another function as an arguments/values.

7.callback function:-
- callback function is a function which is passed as a argument to another function.

Note:- current version of javascript is es14.
es - ECMA script .
ECMA - european computer manufacturing association.
- es6 feature is stable.

explicit return :- explicitly we have to give a keyword called as "return".
explciit return is possible in both normal function as well as arrow function.
implicit return:- implicitly(by default) arrow function will take return.
note:- no printing statements will work after "return" statement.

closure:-closure is a memory created when functions are bundled together/ nested each other by giving accessibility to the innermost function to access the variables of outermost function.

Arrays :-
- it is a non-primitive data type.
- Array is used to store multiple values within a single variable.

syantx :- [];

what is the difference between map and forEach method.
map() :- it is used to iterate the values.
- it returns new copied array from the original array.

forEach() :- it is used to iterate the values.
          - it returns "undefined".

create array of objects
**********

what is object?
- Object is an entity which contains states and behaviors.
-states - properties , behaviour - functionality.
- Object is an entity to store the values in the form key and value pair.

synatax :-   let obj = {};
ex :-   let userdetails ={
        name:"Sam",
        age:23
        }

Note: - Javascript is mainly object -based language till 2015 (es6)
- after es6 feature they added oops concept.

JSON :-
*******
- JSON stands for Javascript object notation.
- JSON is a javscript technique used to transfer the data from client to server and server to client.

syntax :-

js object :-
**********
```
let a = {
        name :"value",
        age:25
}
```

json object:-
**********
```
{
        "name" :"Sam",
        "age" :25
}
```

- to create json file we have to create  a file with extension called
        "filename.json".

json methods:-
**********
1.JSON.stringify() :-it is used to convert JS obejct into json object.
2.JSON.parse() :- it is used to convert json object into javascript object.

Asynchronous in javascript:-

**********
- to perform multiple task at a time.

1.setTimeout() :- it is web api used to perform asyn operations with certain timeouts.
syntax :-  setTimeout(callbackfn , timeout);

2.setInterval() :- it is a web api used to perform async operation at every time interval specified.
synatx :-  setInterval(callbackfn , timeout);

3.promise :- Promise is an object which represents eventual completion or failures of async operations.
        states of promise:-
        1.pending state
        2.fulfilled state
        3.rejected state

syntax :-  let x = new Promise((resolve , reject)=>{});

        to execute the promise we have instance methods:-
                1.then() :- it is used to execute success or fulfilled state results.
                2.catch() :- it is used to execute rejected state results.
                3.finally() :- once promise is settled it will print for both success/failures

4.async & await

note:- javscript is by default "Synchronous(single threaded)".

Create - POST
Read - GET
update - PUT
Delete - Delete