what is JavaScript?

Javascript : It is a client side script used to apply dynamic changes on webpage like validation ,

event programming , animations etc

or

Javascript : It is event based script used to perform dynamic changes on web page , where events are

activity which when when performed lead to access specific functionality.

async function

An async function is a function declared with the async keyword, and the await keyword is permitted within it. The async and await keywords enable asynchronous, promise-based behavior to be written in a cleaner style, avoiding the need to explicitly configure promise chains.

**JavaScript types**

Boolean type.

Null type.

Undefined type.

Number type.

BigInt type.

String type.

Symbol type.

Datatypes

Datatypes : Datatype refers to type of data stored in variable ,

as JS is fully object oreinted hence variables are refrences & datatype indicate typeof data refered by refrence variable.

Var let const are data types

Var is used to global access

Let is scopic in particular block of code

Const means it never change . it is constant always

Note :-

-> JS is dynamically typed

-> To handle each data prototypes are used (number , string......)

Types of datatypes :-

-> **Primitive or Builtin datatypes** : Which have default support by language

number , string , Boolean

-> **Non primitive or User defined datatypes** : Which will be implemented as per users requirementarray , object , function ,array

----------------------------------------------------------------------------------------------------------------------------

Operators

Operators : Those programming entity which perform operation on operands are operator

a+b -> (a+b):expression

(a,b):operands ,

(+):operator

**Classification of operators** :- operators are classified on the basis of number of operands

-> Unary operator : 1 operand : ++ , --

-> Binary operator : 2 operand : + , = , > , && , |

-> Ternary or Conditional operator : 3 operand (condition)?true statement:false statement;

Types of operators :-

-> Arithmatic operator : + , - , \* , / , %

-> Assignment operator : = , += , -= , \*= , /= , %=

-> Comparison operator : == , === , > , < , >= , <= , !=

-> Logical operator : && , || , !

-> Bitwise operator : & , | , ^ , ~ , << , >>

-> Increment/Decrement operator : ++ , --

-> Ternary or Conditional operator

-> Special operator : typeof

Controll structure : It is used to deploy decision making , iterations & switching at application level

-> **Conditional statements** : Used to deploy decision making

\* if statement

\* if-else statement

\* Ladder if-else statement

(if-elseif-else statement)

\* switch statement

-> **Controll statements** : Used to deploy iterations at application level (loops)

Number data : do-while , for , while loop

Collections : for-in loop , for-of loop

-> **Jumping statements** : Used to apply switching at application level

break , continue

**Functions**

Functions : It is block of code which will be invoked when it is called , used to implement code reusability write once & run any number of time.

As Javascript is object oreinted each entity is considered as

object & functions are also considered as reference.

Classification of functions :-

-> **Pre-defined or Builtin function** : Those functions which are available by default in env.

eg : parseInt() , parseFloat().........

-> **User-defined or Custom function** : Those functions which are created as per users requirement.

**-> Types of function on the basis of its implementation:-**

**\* Default function**

syntax :

function function\_name(args) -> if any

{

statements

return -> if any

}

calling syntax : function\_name()

--------------------------------------------------------

\* Anonymous or Variable function

syntax :

var function\_name=function (args){ -> if any

statements

return -> if any

}

calling syntax : function\_name()

ex :-

var demo = function(args){

statement

return

}

Demo();

--------------------------------------------------------

\* Expression or Arrow function

syntax :

var function\_name=(args)=>{ -> if any

statements

return -> if any

}

calling syntax : function\_name()

ex:

var demo=()=>{

statements retrun

}

Demo();

------------------------------------------

\* Lamda Expression or Fat arrow function

\* Callback function

\* Prototype

--------------------------------------------------------

-> Classification onthe basis of argument & return statement :-

\* Function without argument & without return statement

\* Function with argument & without return statement

\* Function without argument & with return statement

\* Function with argument & with return statement

JS Collections

**JS Collections** :- It is collection of multiple hetrogeneous data elements at particular location , apart from array this collections are capable to store multiple hetrogeneous data instead of homogeneous data elements

Depending on storage technique this storages are classified into two parts :-

-> Array

-> Object

-> **Object :** It is collection of hetrogeneous data elements , where each data element will be given by a key:value pair to handle & manupulate data inside object

syntax:

var object\_name={key1:value1,key2:value2,.....,keyn:valuen}

eg:

var empDetails={"eno":1001,"enm":"Jarvis","esal":1000.56}

key : string index

value : any hetrogeneous data

-> **Array** : It is collection of hetrogeneous data elements , where each data element will be given by a numeral index value to handle & manupulate data inside array

i) Manual array

var array\_name;

array\_name=[ele1,ele2,ele3.....elen];

or

var array\_name=[ele1,ele2,ele3.....elen];

ii) Array by using array function

var array\_name=new Array(ele1,ele2,ele3.....elen);

or

var array\_name=new Array(no. of element);

JSON

JSON stands for Javascript Object Notation. JSON is a text-based data format that is used to store and transfer data. For example,

What is the DOM?

DOM stands for Document Object Model. It is a programming interface that allows us to create, change, or remove elements from the document. We can also add events to these elements to make our page more dynamic.

-----------------------------------------------------------------------------------------------------------

**JavaScript Hoisting**

Hoisting is a mechanism in JavaScript that moves the declaration of variables and functions at the top. So, in JavaScript we can use variables and functions before declaring them.