**Variable:**

- Variable is nothing but container in programming language this is known as memory block where we can store the values.

- Syntax:

variable-keywords variable-name = value;

**Data types:**

- It is used to store the different types of data inside one variable name.

- These are known as datatypes in javascript.

- Types of data types

2 types are there

1. Primitive datatype:

- The datatype which we are using to store the value without passing any methods or functions.

- Directly we are able to call the types of data in JavaScript.

- Those types are actually known as Primitive datatype.

String:

- Inside the quotations what all the data we want to write it is going to be consider as String types of data.

- If we want to call numbers also inside quote the number also will be consider as String types of data.

Boolean:

- Boolean is nothing but "true" and "false".

- Based on condition we can call or get the Boolean value.

Undefined:

- if we just declare one variable-name based on specific var keyword and based on this variable-name if we don't want to initialize any value it will return undefined datatype as output.

Important-->Difference between Null and Undefined

**Null: Null means empty spaces.**

- If we don't want to initialize anything or if we want to make empty it will be known as null in JavaScript.

- But it is a bug or error in JavaScript.

- Because when we check the typeof of null it will return "Object" as our output.

- But why object still now the answer is not developed.

- So null is known as bug or error in javascript.

**What is NaN?**

- The full form of NaN is Not a Number.

- NaN is a property of global object.

- In other words, it is a variable of global scope.

- It is usually used to indicate an error condition for a function that should return a valid number.

**IsNaN:**

- IsNaN is a global object to check if a value is an NaN or what.

- Syntax:

console.log (isNaN (NaN)) -->true

console.log (isNaN (any perfect value) -->false

- Number.IsNaN does not convert the values to a number and will not return true for any value that is not of the type number.

- In javascript the value of NaN is considered a type of number.

**Expression**: The combination of operator and operand is known as expression.

**Assignment operator:**

- Assigning some values.

- We are having actually one type of assignment operator

- (=) is going to be consider as assignment operator.

**Arithmetic Operator:**

- +,-,\*,/,% ,++,--

- An arithmetic operator takes numeric values and their operands

and returns a single numeric value.

**Comparison Operator:**

- A comparison operator compares its operands and returns a logical value

based on whether the comparison is true.

- ==,===,!=,<,>,<=,>=

== - Inside multiple variable-name it will only check values will be same or not.

- It will never check any datatypes.

- If the values will be same, it will return true.

- Otherwise, it will return false as output.

=== - Inside multiple variable-name it will only check values and datatypes will be same or not.

- It will check values and datatypes both.

- If the values and datatypes will be same it will return true.

- Otherwise, it will return false as output.

! = - If both the value will be same and if we want to apply!=

it will return false.

- Otherwise, it will return true as output.

**Logical Operator:**

- 3 types we are having

**1. LOGICAL AND (&&):**

- Based on variable-name all the condition has to be satisfied.

- If all the condition will be satisfied, we will get true

- Else we will get false as output.

**2. LOGICAL OR (||)**

- Not mandatory that all the condition has to be satisfied.

- If any one condition will be true it will return true as output.

- If all the condition will be false it will return false as output.

**3. LOGICAL NOT (!):**

- It will return the vice versa results as output.

- If the output will return true and if we will use! operator

it will make the condition as false.

- If the output will return false and if we will use! operator

it will make the condition as true.

**String operator:**

- Inside the quotations either it will be single quote or double quote

what all the data we will write all the data will be consider as

string types of data.

String literals:

- Inside double quotations or single quotations what all the data

we call this is known as string literals.

**//Conditional or control statement**

If: Only true conditions can be evaluated

syntax:

if(condition){

statement;

}

**If-else:**

- The if-else statement executes a block of code if a specified condition is true.

- If the condition will be false, it will execute else block as output.

- The if/else statement is a part of javascript conditional statement which are cause to perform different actions based ondifferent condition.

- Syntax:

if(condition){

// printing statement

}

else{

// printing statement

}

**If else if or Nested If:**

Inside one if statement if we will call another if this is

known as nested if.

syntax:

if(condition) {

stmt;

}

else if(condition){

stmt;

}

.

.

.

else {

stmt;

}

**Ternary operator:**

The conditional(ternary) operator is the only one java script operator that takes three operands.

syntax: condition? expression1: expression2;

**What is switch case?**

- The switch statement evaluates an expression, matching the expression values based on a case and return the data as our output.

- Like else statement inside switch case we are having one more option which is known as default.

- Syntax:

switch(variable-name) {

case "value”:

printing statement;

break;

case "value";

printing statement;

break;

default:

printing statement;

}

**for loop:**

->If the initial and final conditions are clearly known then for loop is used.

syntax:

for (initialisation; condition; counter)

{

task;

}

**While loop:**

->The while loop creates a loop and executes the statement when the condition is true.

->It will check the condition first and if the condition is true then it will execute the statement.

Syntax:

while(condition){

// printing statement;

increment;

}

**Do-While loop in javascript:**

do {

printing statement;

number increment;

}

**while(condition)**

- It will check the condition later.

- Based on the first value it will always print the data.

- Later on, it will check the condition.

- If the condition will be satisfied it will go inside the block and it will print the data as output.

- If the condition will be false, it will execute rest of the cases then it will terminate the program

For Loop:

If the start and stop condition is clearly known we can use for loop.

syntax:

for (declaration or initialisation; condition; increment/decrement)

start condition stop condition counter

{

// loop body

}

**What is function?**

- Function is a block of code which we are using to perform some particular tasks.

- In javascript we are able to re-use those function details.

- So, it is known as reusable functional block also.

- Syntax:

function function-name(//parameter) {

// function body

}

call the function

**What is parameter?**

- After providing a name for function inside the () what all the variable names we called this is known as function parameter.

**What is argument?**

- While calling the function based on parameter what all the values, we will call this is known as

arguments.

**Function expression:**

- Function expression means if we want to store the function inside into any variable this is known as function expression.

**Anonymous Function:**

- Function without function name is known as anonymous function.

What is Fat Arrow or Arrow Function?

- We are using fat arrow function to minimize our code length.

- This is called single line function. Because we can call function within one line.

- This is known as fat arrow function because of the symbol.

- The symbol is (=>)

Fat arrow:

- Syntax:

variable-name = (//parameter) => {

// function body

}

**What is ECMAScript?**

- The full form of ECMA is European Computer Manufacturer's Association.

- ES6 or ECMAScript 2015.

**What is array?**

- If we want to store multiple values with multiple datatypes inside one variable name it is known as array in javascript.

- To store the value in an array there is no size limitations.

- Array value will be store based on index position.

- Array in java script is heterogeneous because we can store number type of value, string type, Boolean type in one array.

- Syntax:

var variable-name = ["value1","value2”, value3, value4, ........]

**Methods of array:**

1.Length:

- We are using length method to check the length of the data based on one specific array list.

2.Push ():

- Push method we are using if we want to add extra element inside our existing array.

- All the elements will be added at the end of the array list.

- No size limitations are here.

3.Pop ():

- If we want to delete the last element data from our array list we have to call pop method.

4. Shift ():

- If we want to delete the first element data from array list, we

have to call shift method.

5. Unshift () :

- Unshift method we are using if we want to add extra element inside our existing array.

- All the elements will be added at the starting of the array list.

- No size limitations are here.

6. Splice ():

- We are using splice method if we want to add or remove any

elements from array.

- Syntax:

splice (arg1, arg2, arg3)

- arg1 will consider the index position

- arg2 will consider the number of data we want to delete

- arg3 will consider the replaced elements.

7. Slice (arg1, arg2)

- arg1 will be consider as 1st index position.

- arg2 will be consider as last index position.

- Data will always print up to last index position.

- It will cut one part of the array

Searching and Filter in an Array

8. indexof ()

- We are using indexof method to search one element inside our array-list.

- If the value is present, it will return the index position as output.

- If the value is not present it will return -1 as output.

- It will always start to search the data from the 0th index position.

9. lastindexof ()

- We are using lastindexof method to search one element inside our array-list.

- If the value is present, it will return the index position as output.

- If the value is not present it will return -1 as output.

- It will always start to search the data from the last index position.

10. includes ():

- We are using includes method also to search a data inside one array.

- If the value will be present, it will return true as output.

- If the value is not present it will return false as output.

11. sort ():

- If we want to sort any data either ascending or descending

order we have to call sort method.

- The default sort order is ascending.

12. reverse ():

- If we want to get the elements in reverse order, we can use reverse method.

a b

500,1,2.30,90,1000,67,4

a b

1,500,2.30,90,1000,67,4

a b

1,2.30,500,90,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,1000,67,4

a b

1,2.30,90,500,67,1000,4

a b

1,2.30,90,500,67,4,1000

a b

1,2.30,90,500,67,4,1000

a b

1,2.30,90,500,67,4,1000

a b

1,2.30,90,500,67,4,1000

a b

1,2.30,90,67,500,4,100

1,2.30,90,67,4,500,1000

**String:**

- In javascript what all the things we need to write inside the

quote ('' or "") all the data is known as String.

J s p i d e r s

0 1 2 3 4 5 6 7

**Finding a String in a String:**

1. IndexOf (search string, index position)

- The indexof method returns the index of(position) the

search data as output.

- It will always start to search the data from left to right.

- If duplicate value will be present, it will take only the first

character as output.

- If the value will not be present it will return -1 as output.

2. lastindexof(search string, index position)

- The lastindexof method returns the index of(position) the

search data as output.

- It will always start to search the data from right to left.

- If duplicate value will be present it will take only the first

character as output.

- If the value will not be present it will return -1 as output.

Replacing String Content

3. replace(searchfor, replecewith)

- The replace method replaces a specified value with the another value

of a string.

- The replace method() does not change the main string.

- It returns a new string.

- By default the replace method replace the first match if any

multiple character will be present.

4. repeat(arg1)

- It will take one argument which is number of times we wants to repeat the

string as our output.

11. toUpperCase()

- Convert all the lowercase data to uppercase.

- Argument passing is not required.

5. toLowerCase()

- Convert all the uppercase data to lowercase.

- Argument passing is not required.

6. concat()

- It will join 2 or more string together.

- No limitations to pass arguments.

**Math in javascript:**

- The math object we are using in javascript to perform some

mathematical operation on numbers.

- Types on Math

1. Math.Pi()

2. Math.Round()

3. Math.abs()

4. Math.floor()

5. Math.ceil()

6. Math.trunc()

7. Math.pow()

8. Math.sqrt()

9. Math.min()

10. Math.max()

11. Math.random()

**Object in javascript**

**What is object?**

- Object is a collection of properties and the properties will be combine with a key and a value.

- Javascript object is a non-primitive datatype that allows us to store multiple collections of data.

- 3 ways we are having to create objects in javascript

1. Using new keyword

- Syntax :

let variable-name = new Object();

2. Using literals

- object literals means inside the {} we have to store the keys and values .

- Syntax :

let variable-name = {key1:value1,key2:value2,.........}

3. Using constructor function

- Syntax :

let variable-name = function(){

// constructor

}

let variable-name = new Object-name classes in java script:

-->class is a blueprint of an object.

**Constructor**:

-->Every class must and should have constructor.

-->There are two types of constructors

1)Default constructor - If we wont specify the constructor the compiler will write its default constructor.

2)Parameterised constructor - If we specify the parameters it is called as a parameterised constructor then the compiler will not create default constructor.

**Inheritance**:

--> One class acquiring the properties of another class is called as Inheritance.

--> A class from where properties are inherited is called as Super class or parent class or base class.

--> A class which inherits the properties from another class is called as sub class or child class or derived class

--> Extends is a keyword used in inheritance.

**Encapsulation:**

To achieve an encapsulation in JavaScript: -

Use setter methods to set the data and getter methods to get that data.

The encapsulation allows us to handle an object using the following properties:

Read/Write - Here, we use setter methods to write the data and getter methods read that data.

Read Only - In this case, we use getter methods only.

Write Only - In this case, we use setter methods only.

**DOM- DOCUMENT OBJECT MODEL**

The DOM is the data representation of the objects that comprise the structure and content of document on web.

--> Use the DOM when we want to interact with the web pages.

--> DOM helps to add , delete, change, remove the contents from the HTML document.

getElementbyId:

---> To select the elements by its id, use document.getElementsbyId method

---> The getElementbyId returns an element which is specified by id.