Java	Javascript
Server Sede programming language	Client & Server sede programming language
Compiled & interpreted language	Interpreted language
Compilation is mandatory	not mandatory
JVM is execution unit	browser is execution unit
Statically typed	dynamically typed
whole block of code will be executed at a time	line by line execution happens
not loosely typed	loosely typed

Types of js:

① Internal js: Can be inserted in body or head tag by using <Script> tag but it's good pratice to use in body tag. It should be the last tag of body tag

Syntax:

```
<Script language="javascript" type="text/javascript">
</Script>
```

Java	Javascript
Server Sede programming language	Client & Server Sede programming language
Compiled & interpreted language	Interpreted language
Compilation is mandatory	not mandatory
Jvm is execution unit	browser is execution unit
Staticaly typed	dynamically typed
whole block of code will be executed at a time	line by line execution happens
not loosely typed	loosel typed

Types of js:

① Internal js: Can be inserted in body or head tag by using <script> tag but it's good pratice to use in body tag, it should be the last tag of body tag

Syntax:

```
<Script language="javascript" type="text/javascript">
</Script>
```

② External js: js can be written Seperate file we have to store it in .js extension. we can link the file by Script tag to html

Syntax: <Script type="javascript" src="Script.js"></Script>

Advantages:

* easy readability, code reuseability
* decrease code length
* reduce page loading time

Disadvantage:

* Stealer can download file
* browser has to make additional http request to get js code
* large change in js code may cause unexcepted result in all of its dependent file

Comment:

① Single line Comment : //

② multi-line Comment : /* */

Advantage:

* Easy to understand
* avoid unnecessary code

js output / printing statements:

① Using innerHTML[DOM Element]: used to write into html Element.

Ex: document.getElementById("id").innerHTML = "Hello";

② Using document.write(): used to write into html o/p

Ex: document.write(5+6);

③ Using window.alert(): used to write into alert box

Ex: document.alert(5+6);

④ Using Console.log(): used to write into browser Console

ex: Console.log(5+6);

Statement:

are written to perform particular task it has 5 Separate Element

① values ② Keyword ③ Expression ④ Operator ⑤ Comments

Variables:

Variable Can be declared using 3 keyword

① var ② let ③ const

var: act as Container to Store data it canbe redeclared, Re-intialized

Ex: var x=5;

let: variable declared with let keyword cannot be Redeclared & have block-scope

Ex: let x=2;

Const: this type of variable can't be Redeclared, Reassigned & have block of scope

Ex: Const PI = 3.14159;

JS Datatype:

① primitive-data type: Number, String, boolean, undefined, Null,

② non-primitive-data type: Object, Array, Date(), ...

Number:

Only one number type we can write with & without decimal

Ex: let x1 = 31.50; let x2 = 3; let x3 = -82;

String:

Series of character with " ", ' '

Ex: var a = "Hi"; let b = 'true'; var c = '20';

Boolean:

It has 2 values true or false

var a = true; var b = false;

Undefined:

In js variable without value has undefined as value

Ex: var x;

Null:

Null is an object we can prove with help of typeof operator

Ex: var a = null;

Typeof Operator: find the type of js variable & syntax

String

```
var a=5;
var b=5.09;
var c=true;
var d="Hi";
var e='Hello';
var f=`welcome`;
var g=null;
var h;
Console.log (typeof a);
Console.log (typeof b);
Console.log (typeof c);
Console.log (typeof d);
Console.log (typeof e);
Console.log (typeof f);
Console.log (typeof g);
Console.log (typeof h);
```

O/p:

```
number
number
boolean
String
String
String
Object
Undefined
```

Operators:

1. Arithmatic : +, -, *, /, %, ++, --, **
2. logical : &&, ||, !
3. Assignment : =, +=, -=, *=, /=, %=
4. Comparision : ==, ===, !=, !==, >, <, >=, <=, ?
5. Bitwise : &, |, ~, ^, <<, >>, >>>
6. type : typeof, instance of, in, new, delete, void

Control Statement:

① if Statement:

```
if (condition){
}
```

② Else Statement:

```
if (condition){
}
else {
}
```

③ Else-if Statement:

```
if (condition){
}
else if (condition){
}
else {
}
```

④ Switch Statement:

```
Switch (condition){
case x:
    break;
case y:
    break;
default:
}
```

loop Statement:

① while loop:

```
while (expression){
    // code to executed
}
```

② do-while loop:

```
do{
    // code to executed
}
while (expression)
```

③ for loop:

```
for ( initialization; condition; increment){
}
```

Ex:
```
for (let i=0; i<car.length; i++){
}
```

④ for-in loop: loop is used to iterate object properties. for Every iteration for in loop store one of the keys in temp reference variable as String

```
for ( key in object){
}
```

⑤ for-of loop:

It allow us to iterate array Element easier for every iteration one of array Element will be stored inside it runs total no of times array Element we have when doesn't find array Element then it terminate

```
for ( variable of iterable){
}
```

JS functions:

function is a group of reuseable code which can be called anywhere in pgm

Syntax: function function_name(parameter_list){

}

Calling function:

<input type="button" onclick="SayHello()">

return Statement:

return the value of function (optional)

Anonymous function:

Anonymous function should be assigned to some variable. (function without name)

Ex: var res = function(){

Console.log("Hello");

}

Arrow function / Lamba function / Fat arrow:

Reduce the code & Increase the efficiency of function

- with arrow function:

hello = () => { return "Hello world"; }

- arrow function return value by default:

hello = () => "Hello world!";

- arrow function with parameter

hello = (val) => "Hello" + val;

Call back function:

function which is passed as an argument to another function

Ex: function callback (a, b) {
return a+b;
}

function add (a, callback) {
console.log (2, callback);
}

function add (2, function (c, d) {
return c+d;
});

Operator

= → Assignment operator used to assign value

== → Relational operator used to check value based on condition

=== → Relational operator used to check value based on condition & datatype

Interpolation:

Process of evaluating template literals which contain one or more place holders (ie exeepts variable name, object properties, function, array etc)

Syntax: document.write (`${var} ... ${var}`);

Anything Enclosed with: "/' are called as String literal.
`` is called as template literals

Note: Interpolation used to avoid multiple Concatination of String literals.

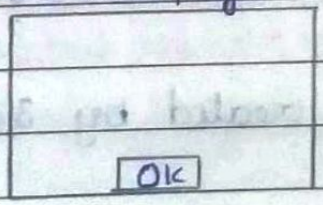
document.write(" The sum of "+a+" and "+b+" is "+c+"
");

ISNAN(): used to check wheather given data is number or not.

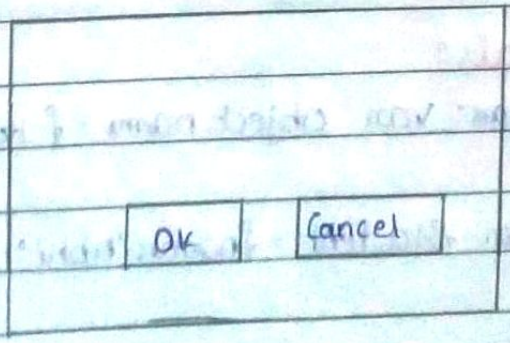
Number(): used to convert the String value to number, only it Contain number otherwise it return NAN

POP UP Boxes:

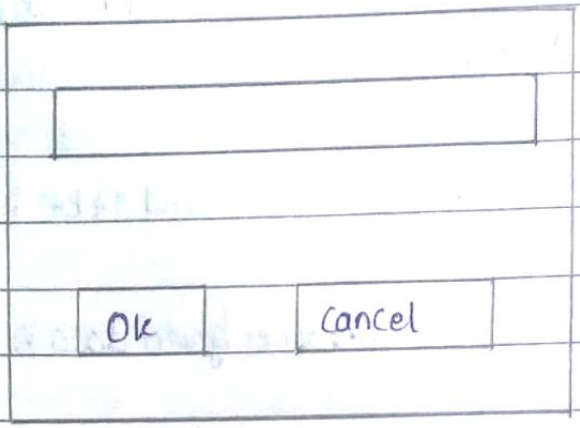
① alert(): used to alert the user or provide warning msg to user. until & unless we press Ok browser will not allow us to access the web page



② Confirm(): used to get conformation of user. when we press Ok true value will be assegned. if we press Cancel false value will be assigned



③ prompt(): Used to take User Input. when we press Ok Entered data will be stored & when we press 'cancel null value will be assigned



Object:

Object is a pair of Key & values. Key refers to properties of object.

if we want to access properties of object we can use dot notation or [] notation.

In [] notation Key must be String other wise we get error.

In js Object can be created by 3 way.

① Using new operator:

Syntax: var variablename = new object();

var student = new object();

② Using literals:

Syntax: var object_name = { key1: value1, key2 = value2 }

var student = {name: "Abhi", age = 34};

③ Using Constructor function:

```
Ex: function Student (name, age){
      this.name = name;
      this.age = age;
    }
```

Reading Object:

```
Console.log(Student.age);
Console.log(Student.["marks"]);
```

Update:

```
Student.age = 10;
Student["name"] = "pallavi";
```

Delete:

```
Delete Student.name;
Delete Student["age"];
```

Array:

* Array is a group of heterogenious element. we can store any type of data in an array

* Array is special type of object, it store multiple value in single variable

* Array size not fixed. we can access array element by index

* In array no continues memory location, we can skip the index. Skipped index is assigned with value called Undefined

* Array index start with 0 & End with length-1

* Manupulation of array Element is very easy with help of build-in method.

Array method

Concat()	Join array
Every()	Return true Every Element of array satisfy the Condition
filter()	Create new array with all Elements of array based on condition.
join()	Join all Elements of array into String (accept 1 String arg & join all Elemt into String)

IndexOf()	It accept 2 arg 1st array Element, 2nd is Index value. It is used to search the Index value of perticular Element. 1st It allways check 1st accurence Index value if we aren't passed 2nd arg. if Element present return the Index value of perticular Element else It return -1
lastIndexOf()	It accept 1 arg & It always searches the last accurence Index value of perticular Elment else return -1
POP()	Return removed Element from array allways element removed at End of array
Push()	always insert the Element at the End. It return the updated array length, It accep 1 String arg.
Reverse()	Reverse the array Element
Slice()	accept 2 arg where 1st arg is Starting Index value & 2nd arg is last Index value. Specefied Index Elements are extracted if last Index is not specifed then It extract upto length-1
Splice()	It accept 'n' no of arg where 1st arg is Index value & 2nd arg is no of Element to be removed & 3rd arg is the new Elements to be added to an existing array Ex: Spilce (2, 0, "Raja", "Rani");
Unshift()	It always Insert the Element at the Starting of an array. It always return update array length
Shift()	It used to remove Element from Starting of an array & It return removed Elements from array.

sort()	Sort array
toString()	Return String representation of an array

String:

String is collection of character we can represent it in Single quotes, Double quotes, backticks

String method:

toUpperCase()
toLowerCase()
Slice (arg1, arg2)
Substring (arg1, arg2)
Substr(arg1, arg2)
replace(arg1, arg2)
include(arg1)
EndsWith(arg1)
StratsWith(arg1)
IndexOf(arg1 arg2)
lastIndexOf(arg1)
concat(...)
charAt(arg1)
charCodeAt(arg1)
repeat(arg1)
split(arg1)
String.fromCharCode(arg1)

Event:

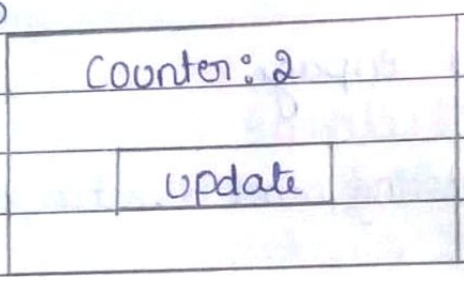
Event are special kind of function which will be executed when we perform any action on web page

Event can be called with help of Event handler all Event handler are prefixed with "on"

Ex: <button onclick="counter()">update counter</button>

```
function counter(){
    let my = document.getElementById("counter");
    my.innerHTML = count++;
}
```

o/p



Attribute:

onclick
onfocus
onbdl click
ondrag
onload
onloadstart
onmousedown
onpput
onunload
onmouseout
onerror
onmessage
onformchange
onoffline
etc...

DOM (Document Object Model):

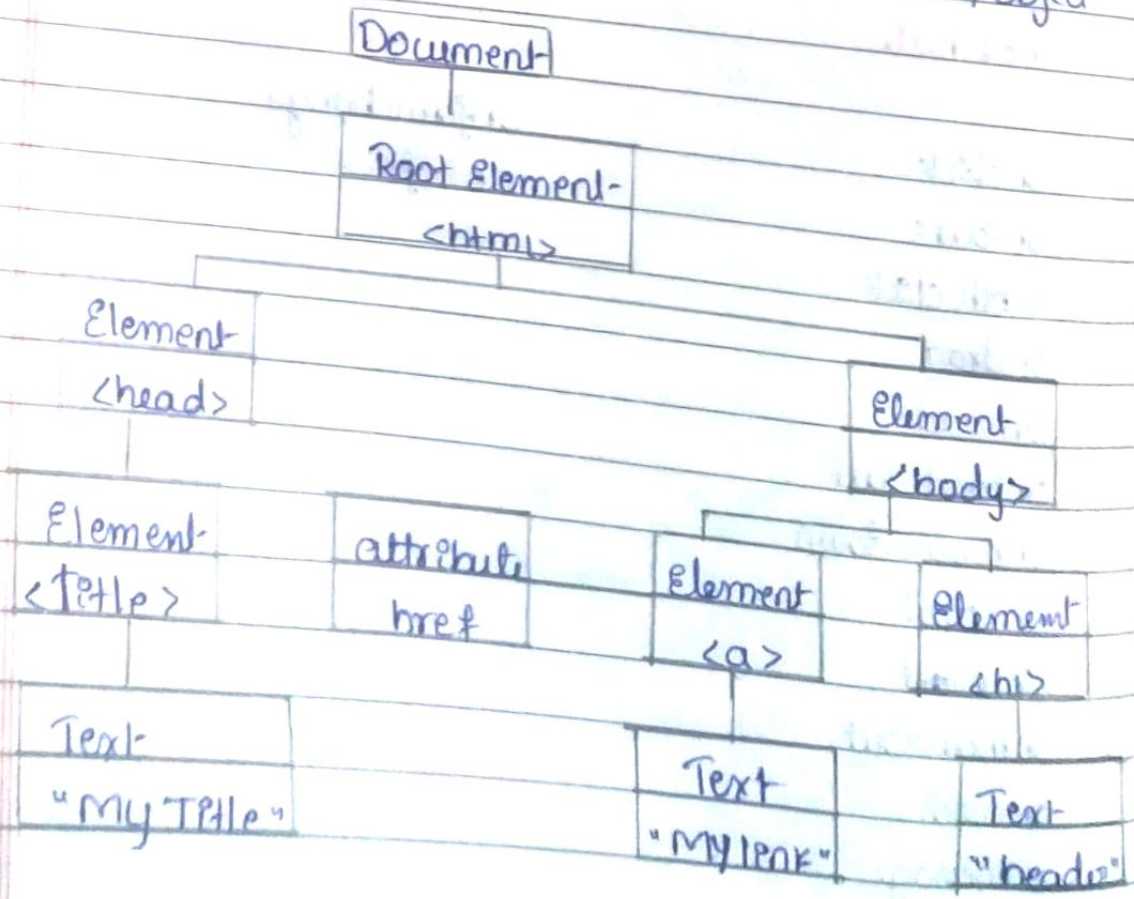
Every web page resides inside a browser window which can be represent as object

-the way a document content is accessed & modified is called DOM

With the DOM, js get power to

* Change all html element, attribute, css in page
* Add / remove html element
* Create new html event in page
* Js can react to all existing html event in page

HTML DOM model is constructed as a tree of object



DOM is w3c Standard (world wide web Construction)

It defines the Standard for accessing document

w3cdom is platform & language-neutral interface that allow pgm & script- to dynamically access & update the Content, Structure, style of document

It has 3 different part-

* Core DOM - for all Doc
* XML DOM - for xml Doc
* html DOM - for html Doc

html DOM is Standard object model & programming interface for html It defines html Elements as object- & properties, method & event of all object-

DOM Document:

document.getElementById("id"): find Element by id

document.getElementByTagname("name"): find Element by tag name

document.getElementByClassname("classname"): find Element by classname

document.queryselectorAll(id, classname, type, attribute, value) group Elements as specified

Changing html Element:

Element.innerHTML = "new content" → Change innerHTML

Element.attribute = newvalue → Change attribute value of html Element

Element.style.property = "new style" → Change style of Element

Element.SetAttribute(Attribute.value) → Change attribute value of Element

Adding Event handlers:

document.getElementById('id'), onclick = function() {

JS-Hoisting:

Hoisting is js default behaviour of moving declarations to top of current scope (to top of current Script or current function)

Js only hoists declaration not initialization

Synchronous pgm	Asynchronous pgm
task are performed one at time & only when one is Completed the following is unblocked.	we can move another task before previous one finish we are able to deal with multiple request simultaneously
wait for task to finish to move to next one	• Complete more task in shorter period.

Promises:

A promise in js objects that links producing code & Consuming code

Producing code that can take some time

Consuming Code is that must wait for the result

JS object can be

• Pending • fulfilled • Rejected

promises allows to handel with asynchronous methods & return value like Synchronous method

pending:

if Asynchronous method in ential State (working) then result is undefined

fulfilled:

Operations was completed Successfull & return value

Rejected:

Operation is failed - it return error object.

we must use promise method to handle promise

How to use promise

Promise.then() take 2 arguments, a call back for success & another for failure

```
mypromise.then(
    function(value){ /* code to successful */},
    function(value){ /* code to some error */}
);
```

JS object	JSON object
name don't require double quote	name require double quote
Key can be String, number, identifier	Key must be String with double quote
String with double quote & Single quote	String value must be double quote

```
{
  type: 'Phone'
  Price: 120
  "Instock": true
}
```

JSON.Stringify() →

← JSON.parse()

```
{
  "type": 'phone'
  "price": 120
  "Instock": true
}
```