

JavaScript :-

- Javascript is a Client side as well as Server side Scripting language.
- Javascript is mainly used to enhance the interest of the user.
- With the help of javascript we can create more lively & interactive website or web application.
- In other words javascript is a prototype based object oriented, imperative, interpreted, case sensitive, lightweight, having functional programming & loosely typed Scripting language.
- With the help of javascript we can create dhtml (dynamic html). javascript makes your Webpage alive.
- Javascript was developed by Brenden Eich in 1995.
- Internally it was named as live Script. But simply for gaining popularity Brenden Eich renamed it to Javascript because Java is very popular at that time.
- Javascript mainly comes to overcome the limitations of VB script (visual basic script). Because VB script is limited to internet explorer & windows operating system. whereas javascript can run on any browser & any operating System.

- Javascript cannot run on its own. Browser is responsible for running javascript code. Every browser have javascript engine which is responsible for executing your javascript code.
 - Google chrome have 'v8 javascript' engine.
 - Mozilla Firefox have 'Spider monkey' JS engine.
 - Internet explorer have 'Chakra' JS engine.
 - Safari have 'nitro' JS engine.

Note :- There is no connection in betn Java and javascript. Because Java is a programming language & javascript is a scripting language.

- class based object oriented :-
Object creation is based on class
- Prototyped based object oriented :-
Object creation is based on function.
- Javascript is platform independent & program dependent.

Transpiler :-

Transpilers in javascript are source to source compilers that transform source code in non-javascript language. That means you've written a program in one language but wish to convert this to another language. Then you would invoke what's called a transpiler.

ECMA International (European Computer Manufacturer Association) :-

It simply created ECMA Script to standardized javascript.

✓ ECMA Script :-

ECMAScript is a standard for scripting languages such as javascript. It means ECMA creates rules for which followed by javascript.

Programming Language

1) A programming language is a type of computer language that consists of a set of instructions for communicating with computer.

2) Most programming languages are complicated. The translation overhead is incurred just once when the source is compiled. After that, it merely

3) Traditional programming languages are based on low-level languages.

4) Execution of a program takes more since they are compiled.

Scripting Language

1) A scripting language is a kind of programming language that is used to automate the execution of operations in a runtime environment.

2) Most scripting languages are interpreted languages. They must be parsed, interpreted & executed each time the program is run.

3) Scripting languages generally prefer high level language.

4) Execution of a script takes less time as scripts are generally short.

- 5) It is used to create new program or piece of software from the scratch.
- 5) It is used to enhance an existing program or automate a specific task. A Scripting language is used to control the behaviour of a program.
- 6) most of regular programming languages are stronger in data & control abstraction.
- 6) Scripting languages generally emphasize the extensibility & richness of expression.
- 7) a Programming language runs or executes independently.
- 7) A Scripting language is runs inside a parent program.
- 8) A programming language generally requires many lines of a code to accomplish a particular task.
- 8) It requires fewer lines than a programming language to accomplish a particular task.
- 9) programming language have high maintenance cost
- 9) It typically have a low maintenance cost
- 10) It generate .exe files.
- 10) do not create .exe files

- 11) UI design & graphic design are often partially supported by programming language
- 11) This are all greatly aided by Scripting language.
- 12) programming language 12) It requires host and self executable.
- 13) All programming languages are not Scripting languages.
- 13) but all Scripting languages are programming languages.
- 14) Ex:- Includes C, C++, Java, Python.
- 14) ex:- perl, PHP, Javascript.

① why javascript is Single threaded?

→ Javascript is a single threaded language that can be non-blocking.

• Single threaded means it has only one call stack. Whatever is on the top of the call stack is run first.

And non-blocking means in javascript we use Asynchronous method.

① Asynchronous means :-

The Async programming the program doesn't wait for the task to complete & can move on to the next task.

② Asynchronous implementation in javascript is done through call stack, call back queue, Web API, event loop.

③ NPM : - (Node Package Manager)

- 1) It known as the world's largest software registry.
- 2) developer can use npm to publish & share their code.
- 3) the command line interface that run from terminal allow you to interact with npm.
- 4) Large public data base of javascript code.
- 5) NPM allows you to install new packages from the registry.

① DOM :-

- The document object model is programming interface for HTML & XML (Extensible markup language).
- It defines Logical Structure. It is called as Logical Structure because DOM doesn't specify any Relationship between objects.
- DOM is a way to represent the webpage in a Structured Hierarchical way.

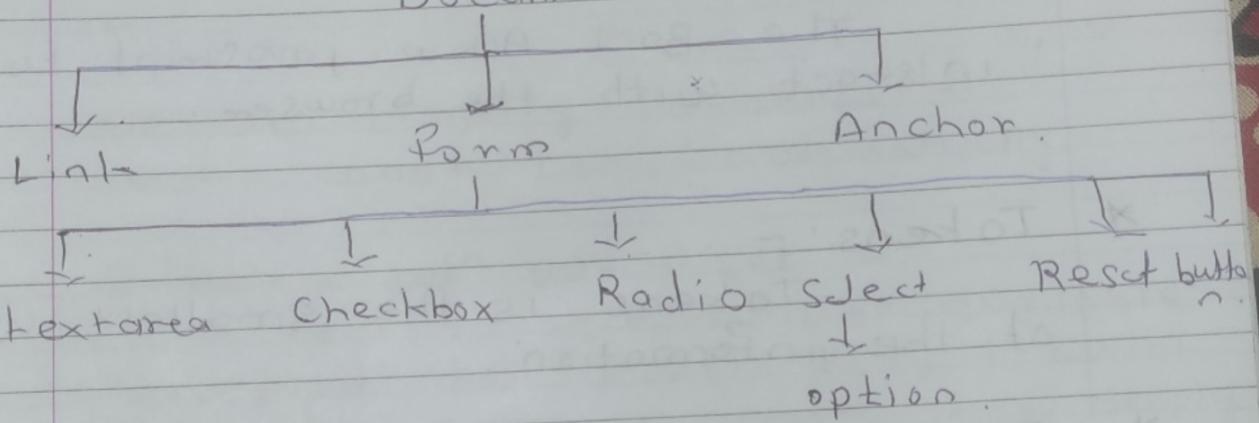
=> HTML is used to structure the webpage & javascript ~~can be~~ is used to add behavior. When an HTML file is loaded into a browser, the javascript cannot understand the HTML document directly. So a corresponding document is created (DOM).

- DOM is basically the representation of the same HTML document but in the different format with the use of objects.
- Structure of DOM :-

It is a tree like representation of a document.

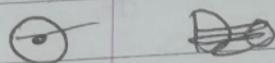
Each branch of the tree ends with a node, & each node contains object.

- Properties of DOM :- It like API that is used to set & get all the properties & methods of the browser.



- Methods of Document Objects :-

- 1) write
- 2) getElementById()
- 3) getElementByName()
- 4) getElementByTagName()
- 5) getElementByClassName()





① BOM :-

The Browser object Model is a browser specific convention referring to all the objects exposed by the web browser.

The BOM Allow javascript to interact with the browser.

* Tokens :-

Tokens is the smallest unit of the information.

* Constant / Literal :-

- Constant is an entity whose value cannot be changed during the execution of program.

Before ESG we define constant by

Var keyword itself by writing name of constant in upper case.

Var My-name^{value} = 100;

- Second way to define constant before ESG

Object define property {}.

* Variables:-

Variables is the name given to memory location whose value may change during the execution of program.

We can classify variables -

Var, const, Let, automatically -

Difference betⁿ Var & Let :-

A Var variable will be available throughout the Function body in which it is defined, no matter how deeply nested its definition.

A Let variable will ~~obt~~ only be available within the same block where it is defined.

Ex:-

Function abc()

{

if (true)

{

var Fvariable = 1;

Let bvariable = 2;

C.L (Fvariable)

Print 1

C.L (bvariable)

Print 2

if (true)

C.L (Fvariable)

→ Print 1

C.L (bvariable)

Print 2

}

}

C.L(CVariable) → Print 1
 C.L(bVariable) → throw error.

- ① on the global level, let does not create property on the global object whereas var does.

Ex:-
 var a = 1;
 Let b = 2;
 C.L("a") → Print 1
 C.L(this.b) → Error.
 undefined

* keywords :-

Keywords are the reserved words whose meaning is already defined we cannot use keywords as variable name.

* Identifiers :-

User defined name

- 1) Variable - lowercase
- 2) constant - uppercase
- 3) function - camel case notation.

* Separators:-

, ; () { } [].

* Operators:-

operators are the symbol which are used to perform some operation on which one or more than one operation.

1) Relational operator:-

i) == → Value, type

== → Value.

ii) != → value, type

!= → value.

2) Assignment operator.

3) Arithmetic operator.

4) Shorthand operator / compound operator /
Arithmetic ^{Assignment} Operator.

5) Conditional operator.

6) Logical operator

7) Increment / decrement operator.

8) Bitwise operator.

>>

>>> unsigned right shift operator.

9) Type operator:-

1) Typeof

2) Instanceof .

10) Miscellaneous operator : - New .

* Data type :-

Data type Simply represents the type & size of data.

There are 2 types of data type

- 1) Primitive data type
- 2) non-primitive data type

Primitive datatype :-

- number
- String
- boolean
- null [Object]
- undefined
- bigint
- Symbol

non-primitive datatype :-

- object
- Array
- Function

① "Use strict": -

Javascript code should be executed in "Strict mode".

means variable must be define using data type

"use strict";

Var a = 20 ;

a = 20 ; -- error.

- Type coercion :-

It is automatic or implicit conversion of values from one data type to another.

There are one key difference b/w type coercion or type conversion - type coercion is implicit whereas type conversion can be either implicit or explicit.

1) Number to string conversion

2) String to number conversion

3) Boolean to number conversion.

4) The equality operator :-

It is used to compare values irrespective of their type.

This is done by coercing a non number data type to a number.

- Let Variable declaration is recommended for modern javascript

- because Let provide block scope Scope, allowing for better control over variable visibility & reducing the risk of unintended side effects.

- Conversion of string to number not possible in the case of '+'.

methods:-

`toFixed(4)` - dec including decimal -
4 digit

`toFixed(4)` - No including decimal -
4 digit

`ParseInt` - convert decimal to int.

These are only `ParseInt` & `ParseFloat` is present in javascript.

Q) Why `null>=0` gives true & `null<=0` gives false.

→ while using (`>=`, `<=`) operat operands are first converted to primitive, then to the same type.

&

(`==`) directly check the datatype.

document.write ("Hello").
 object method string.
 ↑
 object of
 Html document.

`Console.log()`

L object of Object.

Html document - is a prototype.

Javascript popups or Javascript boxes:-

1) Alert() :-

only have 1 option "ok" if
return undefined.

2) Confirm() :-

have 2 option "ok", "cancel",
it returns "True", "False".

3) Prompt() :-

provide input field with 2 option
if we click cancel it return null.

* Alert is a Function of BOM.

* Linebreaks :-

For Loop :-

for-in Loop :-

It works on the indexing.

for (var index in arr)

{

 d.write("arr :" + arr[index])

}

For-of-Loop :- It gives values of array

For (var value of arr)

{

 d.w(value);

}

For-each :-

Var arr = [1, 2, 3]

arr.forEach(element =>

{

 d.w("element "+ elements)

});

Function :-

Function is a self - content block of statement every function perform some specific tasks. There are two types of function.

1) predefined Function :- alert, prompt, confirm()

2) User-defined Function :-

Method is simply considered as a property.

The type of function is Function but functions can best be described as object.

There are 4 types to write a function.

1) Function Declaration & Function definition:-

```
function addition()
{
}
```

2) Function expression / Anonymous Function / Variable Function :-

```
var res = function()
{
}
res();
```

3) Lambda expression / Arrow Function :-

```
() => {
}
```

4) Immediately invoked Function expression

```
(function() {
})();
```

5) Immediately invoked arrow function :-

```
CC => {
    })();
}
```

Quiz on Function:-

1) Function greet() {
 console.log("Hello");
}

```
var sayHello = greet;  
sayHello();
```

→ Hello.

2) opt.fun(p,q) . other way
→ opt["fun"](p,q)

① function parameters

- default parameters : - function(x,y)
- default parameter values : - function(x,y=10)
- Rest parameter : - function(...x)
- The argument object : - (built-in object)


```
x = findMax(1, 2, 3)
for (let i = 0, i < argument.length, i++) {
  if (argument[i] > max)
```

① Use of this keyword : -

- In an object method, this refers to the object. also refers to global object.
- In Function, this refers to global object.
- In Function, in strict mode, this is undefined.
- In an event, this refers to element.
- methods like call(), apply(), & bind() can refer this to any object.

② Methods in function : -

i) call() : -

It is predefined method.
with call(), an object can use a method belonging to another method.

for ex : -

```
var person = {
    fname: function() {
        return this.name;
    }
}
```

```
Var person1 = {
    name: "Andrew",
}
```

```
Person.fname.call(Person1);
```

② call method can accept argument.

Ex: - Person::fname::call(person1, "indore",
"india");

2) Apply() :-

The apply is similar to the call() method.

Ex: - Person::fname::apply(person1);

Difference betn call() & apply() :-

call() method takes argument

separately.

apply() method takes argument as an array.

Ex: -

1) person::fname::call(person1, "india").

2) person::fname::apply(person1, ["indore",
"india"]);

Math::max::apply("A", [1, 2, 3]) ...

return 3.

3) Bind ()

an object can borrow a method from another object.

Ex:- Let `Fullname = Person.FullName`
`bind (member);`

① promise② Events :-

Every action which can change the state of an object is known as event.

```
<img Src="" height="" width="">
<p align="">
<input type="i" value="">
```

img } elements (HTML)
 p }
 input } object (Javascript)

Src	align	} Attributes (HTML)
height	type	
width	value	

} Properties (Javascript)

HTML - Element
 Button < JS → Object

↑
 Action
 click

↑
 event

click element .

- Click event / mouse event
 - onclick
 - ondblclick
 - onmouseevent
 - onmouseover
 - onmousemove
 - onmouseup
 - onmousedown

- Form event
 - onsubmit
 - onfocus
 - onblur
 - onchange

- keyevent
 - onkeyup
 - onkeydown
 - onkey press

- Load event / document event
 - onload
 - onunload

- click (Action) event

↓

on(click) → onclick → Function →

↑

event
Handler

event handle
function.

- Adding multiple listeners for a single event :-

```
myElement.addEventListener("click", functionA);
                           || -----> "click", functionB);
```

- * Object :-

A javascript object is a collection of named values

all objects in javascript are instances of Object.

only objects that doesn't inherit from object.prototype - null prototype.

It provides powerful mechanism to override & extend object behavior.
To make it more secure, object.prototype is the only object in core javascript language that has immutable prototype.

- instance methods of object :-

- 1) Valueof()
- 2) ToString()
- 3) ToLocaleString()