

History of Java Script

Java Script was created by Brendan Eich in just 10 days in May of 1995 while he was working for Netscape Communication Corporation. Initially, it was called Mocha and was later changed to LiveScript. However in an effort of capitalize on the popularity of Java programming language, it was eventually named JavaScript.

What is Java Script

JavaScript is a high-level, interpreted programming language used to make dynamic and interactive web page. It is also commonly used on the server-side through Node.js.

Features

- * Supported for object-oriented, functional, and imperative programming ~~language~~ paradigms.
- * Dynamic typing & loose typing.
- * Support for Handling asynchronous events through callbacks, promises, and async/await.
- * Built-in support for handling data types such as arrays and objects.

What can we do with JS?

- We can make (i) web app / mobile app / network apps
- (ii) CLI tools
- (iii) games.

Do we need a compiler?

- for running JS we need

JS
engine

Browser	JS engine
In Firefox	Spider Monkey
In Chrome	V8

- We can run JS code in the browser without any code ID.

Script tag : → All js code will written in this tag.

```
<script>  
    console.log('Namaste Dunia');  
</script>
```

* this tag is used for client side scripting

* // this is comment

Linking of HTML file to JS file

```
<script src="index.js"> </script>
```

We use two ways for running the code

① On console

② On terminal → write :→ node index.js
file name

Variable (named memory location)

let a = 5; → variable name
let name = 'hozaifa';
let status = True;
let b = 2.5;

var a = 5
var name = 'hozaifa';
var status = True;
var b = 2.5;

var V/s let

{
 let a = 5
 ...
}

let perform
only here
(locally)

var → globally

- Redecclaration is not possible in case of let.
- Redecclaration is possible in case of var.

Constants (fixed value)

- Not possible to change it.

Rules for variable naming

- (1) Cannot be a reserved keyword.

[var let const if else for while do
switch case break continue return function
this new null undefined true false try
catch finally throw class extends
import export async await]

- (2) Meaningful name
- (3) Cannot start with a number
- (4) Cannot contain "space" or '_'
- (5) Camelcase [Ex → mohalHozaiFa, anujKumarGupta]

Defining multiple variable

let a, b; → ans → undefined
let a=1, b=2;

let a=1; } best practice
let b=2;

Primitive Types

- string → 'Mohd Hozaita' (sequence of character)
- Number → 1, 2, 3, 4, 1.23, 12.05
- Boolean → True or false
- Undefined → [let a;
console.log(a);] → output → undefined
- Null → defined value which is Null [let d = null;
console.log(d);]

Dynamic typing

```
let lastName = 'hozaita';  
console.log(lastName);  
lastName = 4  
console.log(lastName);
```

output
hozaita
4

Reference Types

- objects

```
let person = {
```

```
  firstName: 'hozaita',
```

```
  age: 24
```

```
};
```

for access

① dot notation

person.age

② bracket notation

person['age']

- Arrays

DS used to
contain a list
of items

```
let name = ['Ram', 'rahul']
```

names[0] → Ram

names[1] → rahul

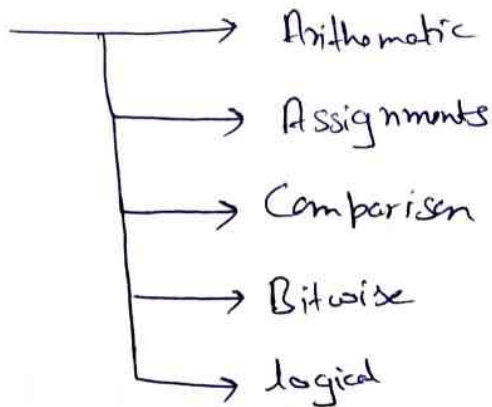
names[2] → undefined

```
names[0] = 'hozaita';
```

↑
override

- Function

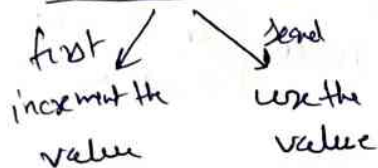
Operators



Arithmetic (+, -, *, /, %, **)

* Pre/Post --- Inc/dec

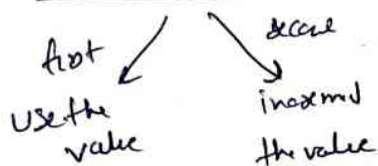
(i) Pre-Increment (++x)



let x = 10;
console.log(++x);

output
11

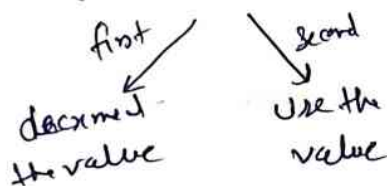
(ii) Post-Increment (x++)



let x = 6;
console.log(x++);
console.log(x);

output
6
7

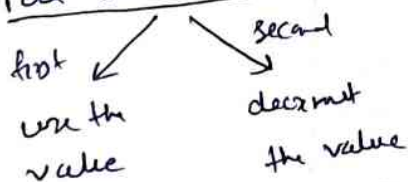
(iii) Pre-Decrement (--x)



let x = 10;
console.log(--x);

output
9

(iv) Post-Decrement (x--)



let x = 6;
console.log(x--);
console.log(x);

output
6
5

let a = 5; let b = 10

let ans1 = (++a) * (--b) ⇒ 54

let ans2 = (a++) * (b--) ⇒ 40

let ans3 = (a++) * (b--) ⇒ 56

let ans4 = (++a) * (b--) ⇒ 63

Assignment

$x = x + 5$
 $x += 5$ } same

Comparison ($>$, $<$, $>=$, $<=$, $===$, $!==$)

Equality Operation

Equality Operation

→ Loose equality ($==$)

→ Strict equality ($===$)

Ex

let num = 1; → int

let str = '1'; → string

num === str

out → False

↑

num == str (don't care about data type)

out → True

Ternary Operator

Ex

```
let age = 20;  
let status = (age >= 18) ? 'canvote' : 'cantvote';  
console.log(status);  
output ⇒ canvote
```

Logical Operator

- AND → $\&\&$ (all condition is true then true)
- OR → $\|\|$ (if only one condition is true then true)
- NOT → $!$ (True til False)
False til True)

With Non-Booleans (concept of false & Truthy)

falsey → undefined, null, 0, false, '', NaN

Truthy → anything that is not falsey

(True $\|\|$ false) → True

OR → short circuiting

(false $\|\|$ 5 $\|\|$ 1) → 5

Bitwise Operation

Bitwise AND $\rightarrow 8$
Bitwise OR $\rightarrow 1$

A	B	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Ex let $a=2$; $[000010]$
let $b=3$; $[000011]$
`console.log(a & b);`
output $\rightarrow 2$

Operator Precedence

(don't learn precedence table)
let $c = a + b * d / f$

Use this

`let c = (a + (b * d) / f);` \rightarrow Best Practice

Control Statements

if-else
switch

If-else-if-else

Ex

only one

multiple times
allowed

only one

```
let marks = 90;  
if (marks >= 90) {  
  console.log('A');  
}  
else if (marks >= 80) {  
  console.log('B');  
}  
else if (marks >= 70) {  
  console.log('C');  
}  
else if (marks >= 60) {  
  console.log('D');
```

```
else {  
  console.log('E');
```

output $\rightarrow A$

Switch case

Ex let num = 2

```
switch (num) {  
  case 1: console.log('A');  
    break;  
  case 2: console.log('B');  
    break;  
  case 3: console.log('C');  
    break;  
  default: console.log('D');  
}
```

output
B

Loops

(Repetition
of
tasks)

- For loop
- While loop
- Do-while loop
- What is an infinite loop?
- For-in-Loop
- For-of Loop

For loop

Ex

```
for (let i = 0; i < 5; i = i + 1)  
{  
  console.log(i);  
}
```

output
0
1
2
3
4

While Loop

Ex

```
let i = 0;  
while (i < 10)  
{  
  console.log(i);  
  i++;  
}
```

output
0
1
2
3
4
5
6
7
8
9

do-while-loop:-

condition is true or false does not matter.
do-while-loop atleast one time.

<u>Ex</u>	<pre>let y = 1; do { console.log(y); y++; } while (y < 10);</pre>	<u>output</u> 1 2 3 4 5 6 7 8 9
-----------	--	--

<u>Ex</u>	<pre>let x = 1 do { console.log(x); x++; } while (x > 10);</pre>	<u>output</u> 1
-----------	---	--------------------