Javascript Basic - I

Node is use for run you JS outside the browser.

When you can do code of C++ in Javascript we can call it as a node used for server-side

↳ Javascript use to add functionality or behaviour in our webpage

What is JS ?

ligh wight programming language / scripting language

client side scripting language.

How it comes?

Javascript was invented by Brendan Eich in 1995.

It was developed for Netscape

What can we do with JS ?

Web app

Mobile app

Networking app

CLI tools

Games

Do we need a compiler ?

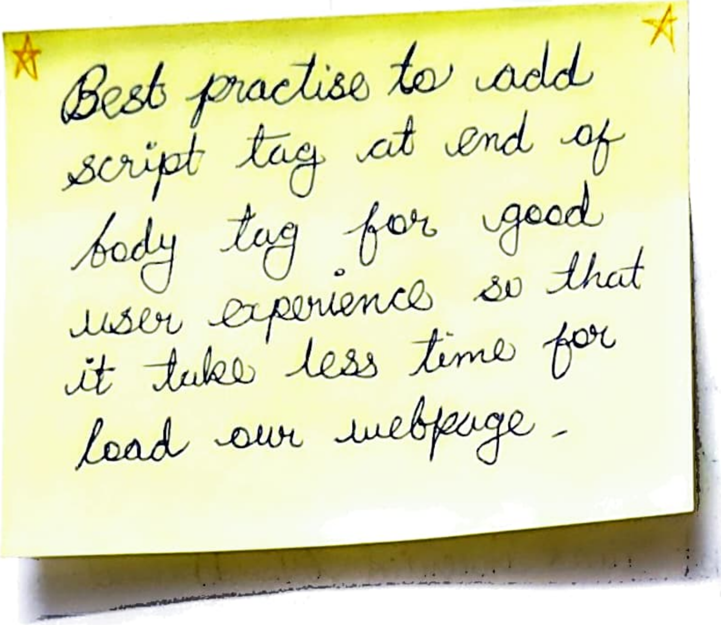
yes , JS Engine

for example, Chrome use V8 JS engine

Firefox use spidermonkey

Adding JS in code :-

you have to write your all code of JS
<script> ... </script>.
you can add <script> is body or head both place


Best practise to add script tag at end of body tag for good user experience so that it take less time for load our webpage.

Hello World in JS :-

```
console.log("Hello World");
```

Comment in JS :- use double slash [//]

```
//hello this is comment line.
```

Variable :- ~~an~~ named memory location
according to its scope there are 2 ~~categ~~ types

- * var - global mean you can access in all code
- * let - local mean scope is limited

you can create variable using

- * const - value is constand can't change.

here var, let & const is keywords.

let a = 5; → number

var name = 'Khushi'; → string

const PI = 3.14;

You can write mutiple variable in single line but best practice use different lines.

Primitive Types

String	var name = `Khushi';
Number	const PI = 3.14;
Boolean	let result = true;
Undefined	var k;
Null	var data = null;

Dynamic Typing :- same variable can be used to hold different data types.

var demo; undefined

demo = 47; undefined → number

demo = true; number → boolean

demo = 'Khushi'; boolean → string

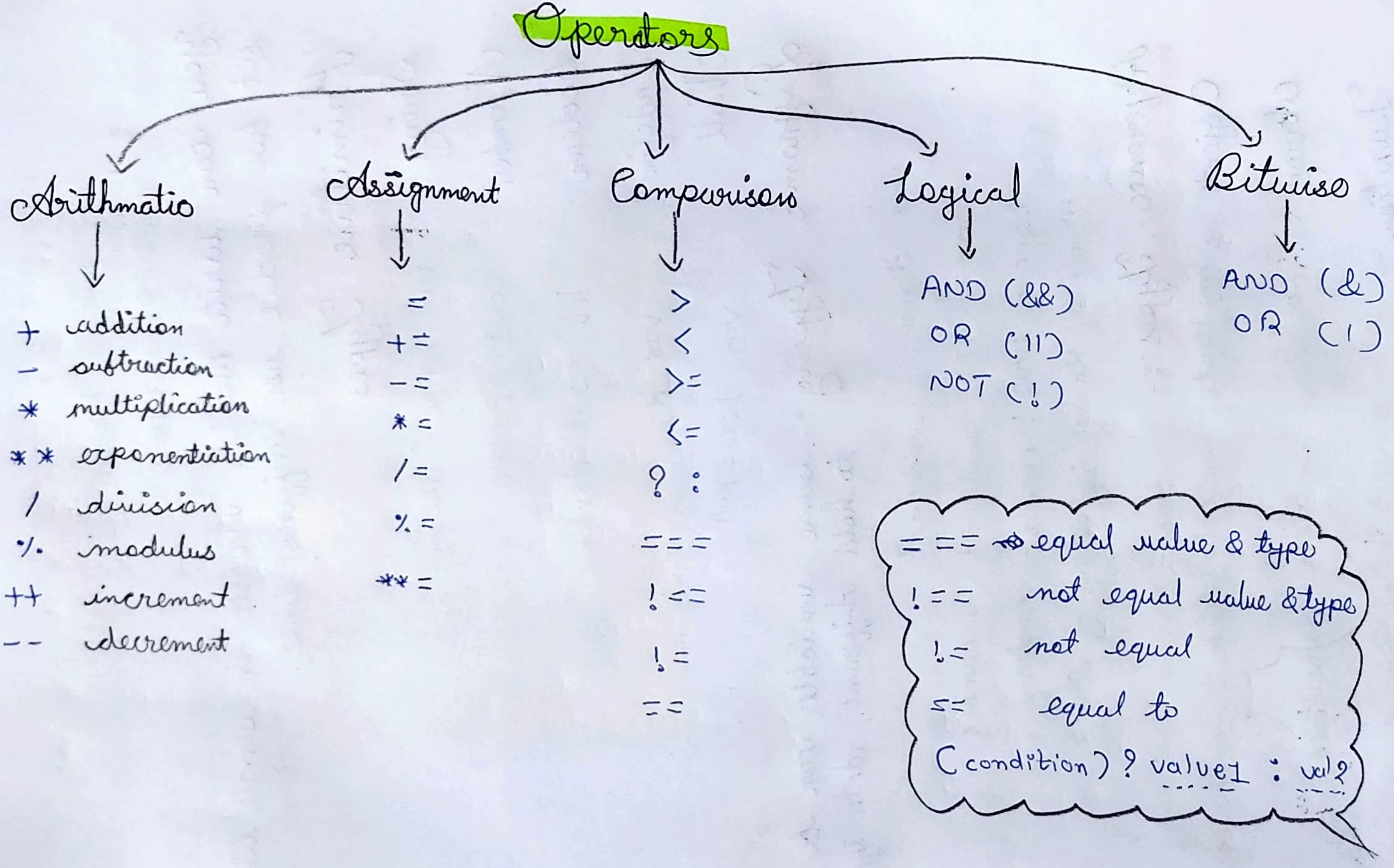
Reference Types :-

Object → multiple variable linked with each other.

Array → collection of items (variables)

Function → block of code that fulfil a specific task.

— We will learn it deep further ☺



Operators

Arithmatic

+ addition
- subtraction
* multiplication
** exponentiation
/ division
% modulus
++ increment
-- decrement

Assignment

=
+=
-=
*=
/=
%=
**=

Comparison

>
<
>=
<=
? :
===
!==
!=
==

Logical

AND (&&)
OR (||)
NOT (!)

Bitwise

AND (&)
OR (|)

=== equal value & type
!== not equal value & type
!= not equal
== equal to
(condition) ? value1 : val2

* With Non-booleans value in OR (||)

```
false || true      => true
false || 'khushi'  => khushi
```

2 types of value

falsy
- undefined
- null
- 0
- false
- NAN

truthy
- Anything that is not falsy

OR(||) will stop checking condition onec it get first true value.

(null || 5 || 1) => 5

For operator precedence use brackets :)

Control Satetement :

if-else

```
if (condition){
    ...code
}
elseif(condition){
    ...code
}
: add as much as
: you want elseif :)
else {
    ...code
}
```

switch

```
switch (expresion)
{
   case... : //code.
             break;
   case ... : // code.
             break;
   : you can add
   : as much case
   : as you want. :)
   default : //code;
}
```

Loops

for loop

```
for(initialisation; condition; updation)
 {
   .... //code

 }
```

While loop

```
 initialisation;
 while ( condition)
 {
   ..., //code
   updation;
 }
```

Do while

```
initialisation;
do
{
   .... //code
   updation;
}
while(condition);
```

For in loop

```
for( key in object)
{
   .... //code
}
```

For of loop

```
for(variable ofiterable)
{
  ..//code
}
```

Break : to jumps out of loop

Continue: to skip one iteration of loop