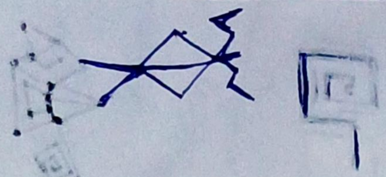

J

3/1 + 2 ** 2

3/1 + 4

(7)

- console window → (clear → ctrl + l) (for window)

→ ctrl + → size inc.

→ In console →

- REPL → Read - Evaluate - Print - loop

- ↑ → पुरानी चीजे दिखेगा
  ↓ → नई ~~चीजे~~ चीजे

- Variable → Name of ^storage^ location

- type of a → To find type of a.

- अगर ज्यादा no. like dia point के बाद तो automatically nearest ~~value~~ value में convert like 0.99999999999999999 = 1

- NAN → Not a no. like { 0/0 ~~also~~ will give type NAN }
  (vo khud तो no. But vo is to represent no no. hai.)

→ NAN + 1 = NAN (A NAN के साथ को भी operation give invalid no.)

- Operator precedence →

  ( )
  ↑
  * *
  ↑
  *, /, % (Left to right)
  ↑
  +, -

  - (5 + 2) / 7 + 1 * 2
    7 / 7 + 1 * 2 (L to R)
    1 + 2
  - 4 + 1 * 6 / 2
    4 + 3 = 7

- Let keyword
  let age = 29;   or   let age ;
  (If ask age → undefined)
  age = 25 ;

- Const →
  Can't change { • const num = 3 ;
  num = 10 ;
  error }

* let ispass = 'true';
अगर ये string type वाला है
if bool change to
' ' ये हटाओ

- var (old syntax) {आज कल इसे let से replace कर दिया}

practice var age = 23;
- let age = 23 ;

age + 2;
age ;
it will give 23

- Assignment : += , -= , *=
- Identifier :
(i) a-z , 0-9, _ , $
(ii) begin with letter, $ or _ .
(iii) case sensitive.
(iv) Reserve words can't used.

- How to write
(i) camelCase [mostly used in JS] (first letter small baki sbhi words ke letter (first) capital)
(ii) snake_case (underscore used)
(iii) PascalCase (All words start with capital letter)

type change is allowed here

let a = 5;
type of a;
no.
a = true;
type of a
'boolean'

- **JS is dynamic typed.** (data type change) | Typescript is static typed (No data type change)

*: String

let role = 'iron'
or
let role = "iron"

{ Double या single किसी एक को use करना है - good appearance (consistent) }

- Let sentence = 'this is "apple"' { Bahar single ही andar double use karo }

= ('this is' apple") → yaha the string consider हो जाएगा & baki ka error

String indices

→ Indexing → If length se zyada print karna chahta to undefined

→ let name = 'tony stark';
→ name[name.length];
undefined.
→ name[name.length - 1];
'k'
→ name.length
10

OR

- "shradha".length
7
- "shradha"[6]
'a'

Concatenation

adding strings together

- "tony" + " " + "stark" = "tony stark"
- "tony" + 1 = "tony1"
-

NULL and undefined in JS

- Undefined → variable not assigned a value is of type undefined.
let a ; → undefined , type is also undefined.
a;
- Null → The null value represent the intentional absence of any object value.

let a = null;
undefined. [isko upper vala likhenge to ye but value of a is null]
a
null [value of a]

null is a keyword is JS.

- length of empty string → 0 ("")
,, ,, single space string → 1 (" ")