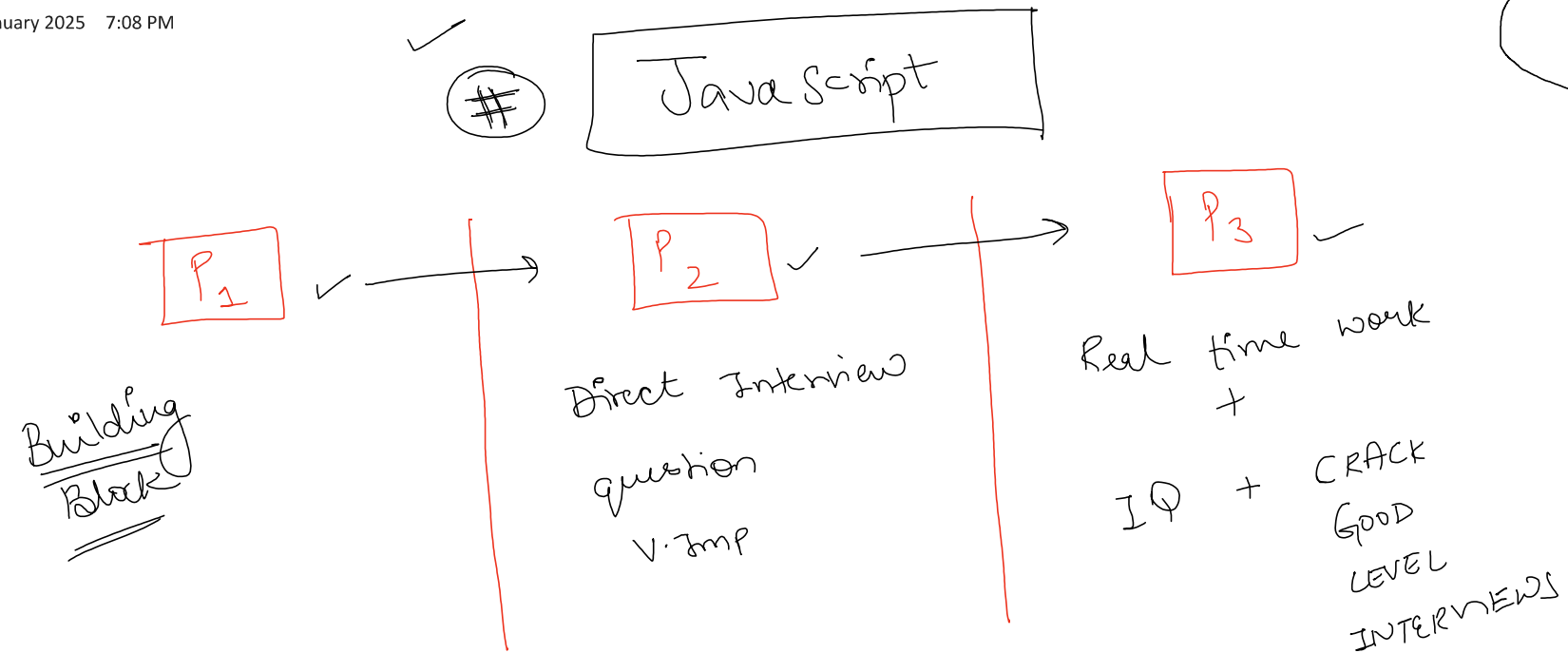


## Day-11()

Wednesday, 15 January 2025 7:08 PM



\_\_\_\_\_ X \_\_\_\_\_ X \_\_\_\_\_ X \_\_\_\_\_

# Phase - 1

V.Jmp. ~~task~~

★ Everything inside JS is an object.

★ JS is the only PL that is understood by the Browser

★ Browser ? → Chrome, Firefox, Safari, Brave etc ...

★ PL ? → set of instructions, understood by compiler / software (VS code)

★ Write your logics ★

★ Console ? → { R → Read  
E → Evaluate  
P → Print }

( { → min  
L → Loop )

How JS? Datatypes ⇒

8, 8 PD, (7, 1)

# Datatypes → (8)

{ PD → (7)  
D → (8) }

✓ PRIMITIVE (7)

✓ NON-PRIMITIVE (1)

Object } (1)

(7) {  
Number  
String  
Boolean  
undefined  
null  
Symbol  
BigInt

# Primitive → immutable  
→ single valued  
predefined ✓

Simple

⇒ something which further  
cannot be broken down.

X Milk

← Paneer

DHABA  
RAVI BHOJNALAY 100% Veg  
✓ Paneer 2pyaza ✓ PP  
✓ Paneer Tikka ✓ TPK  
paneer ...

✓ Kadhar  
✓ Butter Paneer

Primitive

DT

# # VARIABLES

→ container ~~which stores~~ data → value.

10

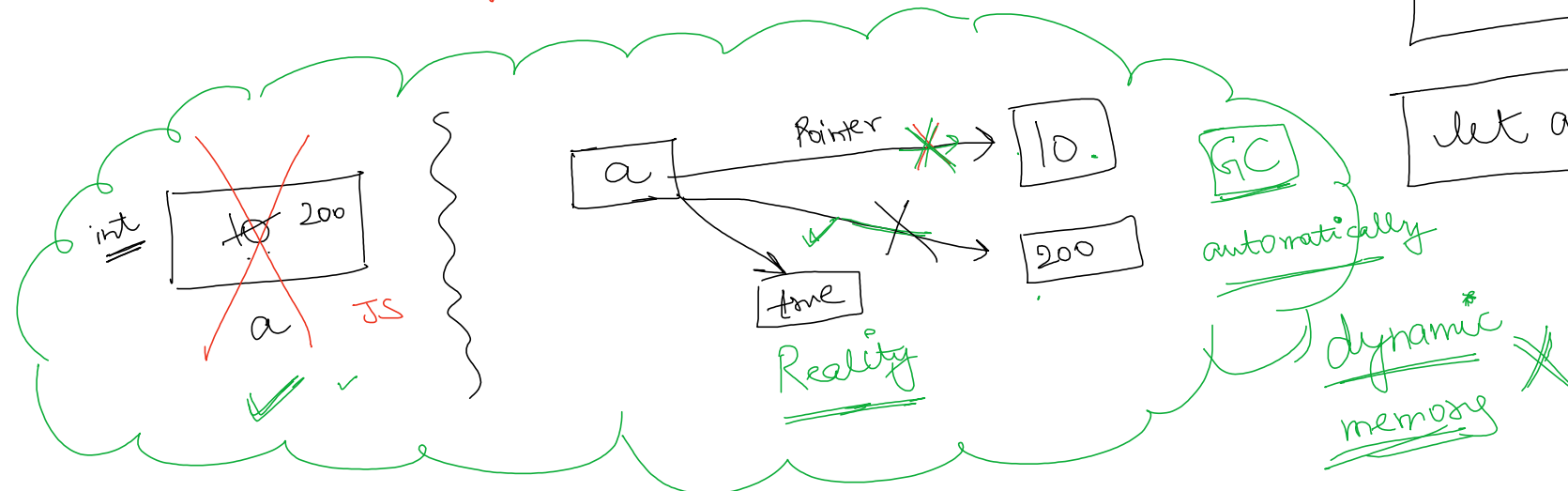
$a = \text{true} \quad \times$   
 $a = 200 \quad \checkmark$

```
int a = 10;
```

C++

let a = 10; JS

a = true ✓  
a = "sm" ✓



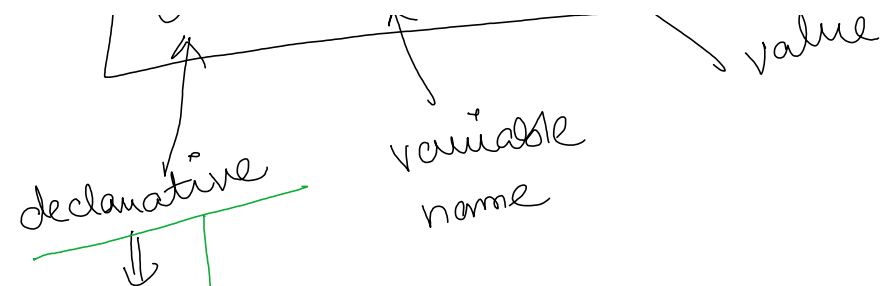
★ Everything<sup>p</sup> points in JS ★

✓ P1

# Variable write

let  $abc = 10_j$

eg:  $\frac{\text{let } a = 10}{\text{let } b = \text{true}}$



OneNote

```
let c = "Samarth"
```

var, let, const

# Nomenclature for a variables (rules):

Rules  
\*\*\*\*

- a-z, A-Z, 0-9
- we can't start with number
- only special charact → - \_ \$
- can we start - & \$
- reserve keywords ✗ (break, for, if, continue--)
- only use - & \$ ✓ (yes)

# let samarthvohra = "web"

let samarthvohra = web development = true (general)

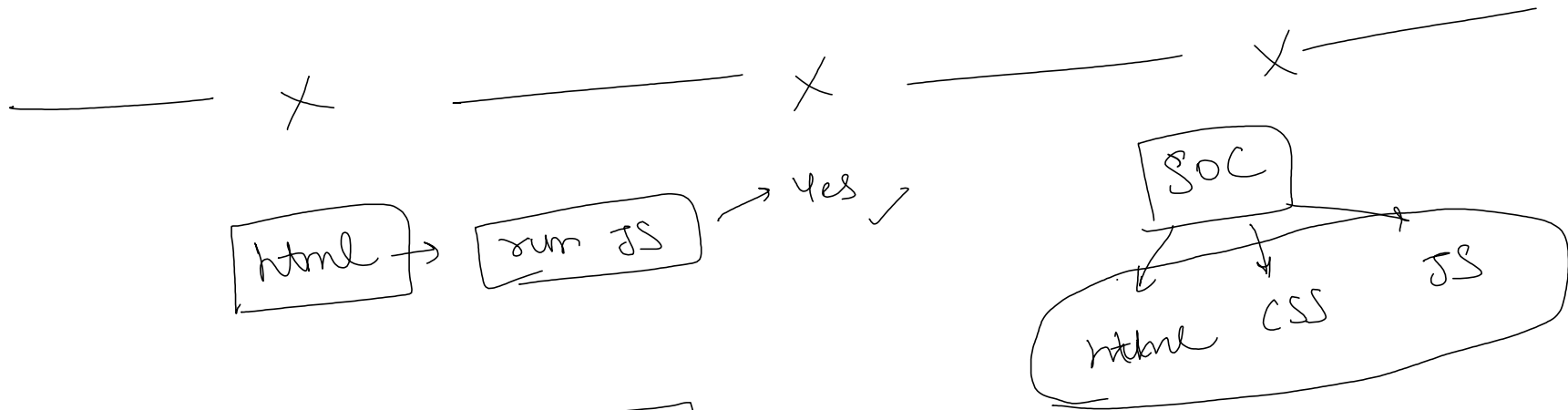
let samarthvohra = 1

let gomykylbmeury — — — — — 1 — — — — —

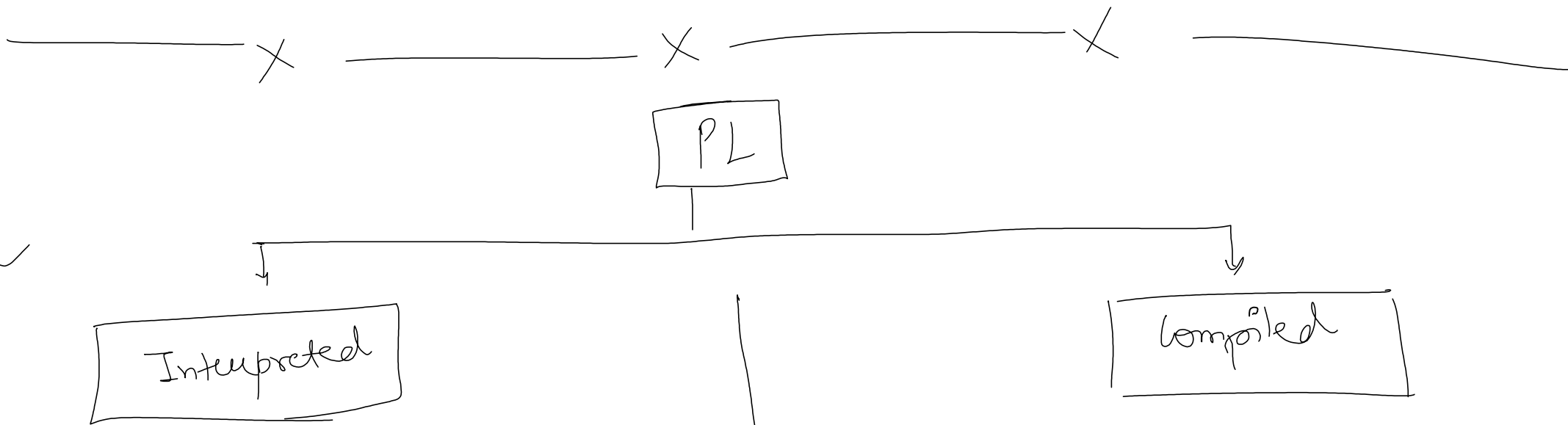
# ways to write variables:

- ✓ ① Camel Case → samarthVohraWebDevelopment ★
- ✗ ② Kabab Case → samarth-vohra-web-development
- ✓ ③ Snake Case → samarth\_vohra\_wels\_development (No)

\$, -



# Numbers



1. overrules your

2. the entire code get build &

- line by line execution  
code.
  - if there is an error, the code will stop there.
- eg: JS, python

$\% \Rightarrow$  remainder

$$\begin{array}{r} 2 \overline{) 21} 10 \\ \underline{- 20} \\ 1 \end{array}$$

$$\begin{array}{r} 6 \overline{) 27} 4 \\ \underline{- 24} \\ 3 \end{array}$$

$$\begin{array}{r} 20 \overline{) 10} 0 \\ \underline{- 0} \\ 10 \end{array} \leftarrow \text{remainder}$$

- OneNote
- executed at once.
  - all the errors appears together.
- eg: C++ / java

$$21 \% 2 = 1$$

$$27 \% 6 = 3$$

