

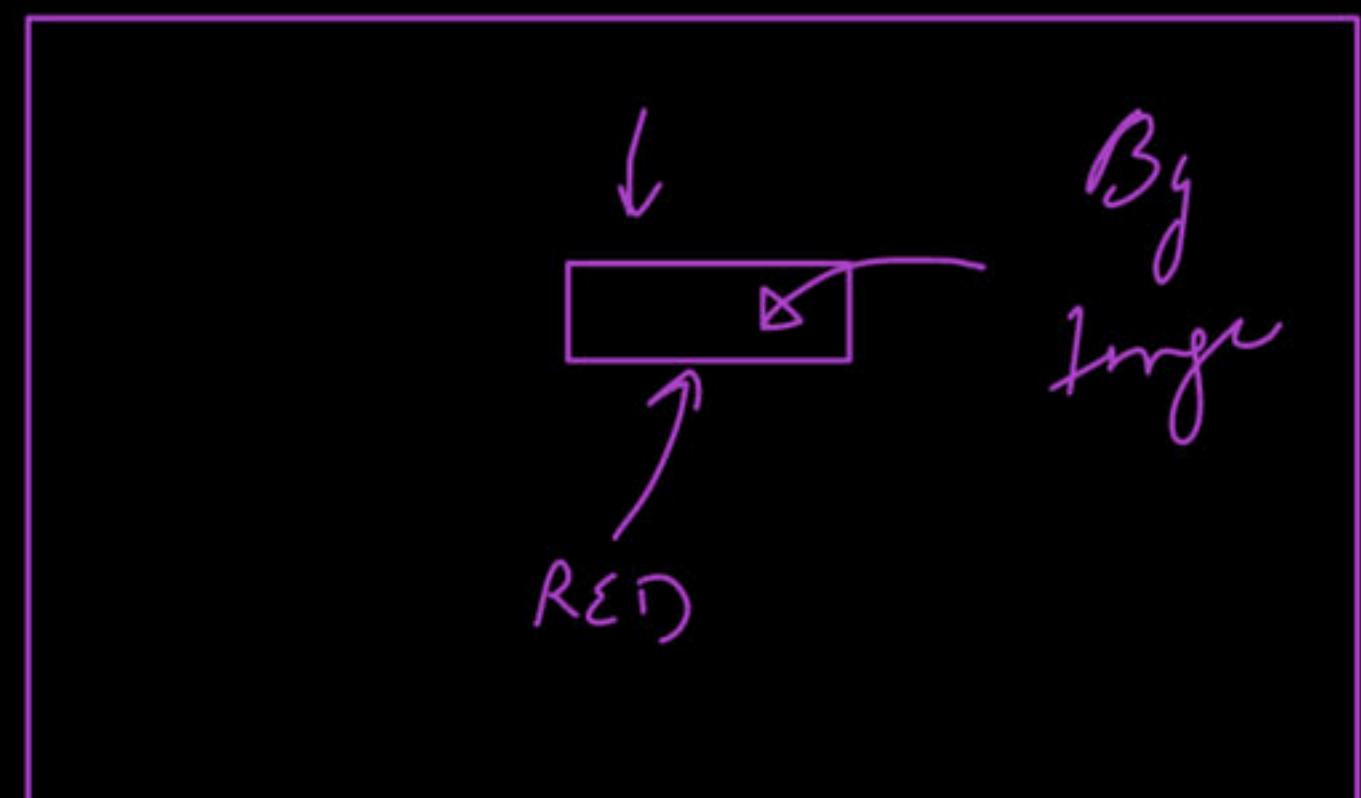
JS Basics - Class I

Special class

JavaScript Basics



Instructor: Love Babbar



What is JavaScript ?

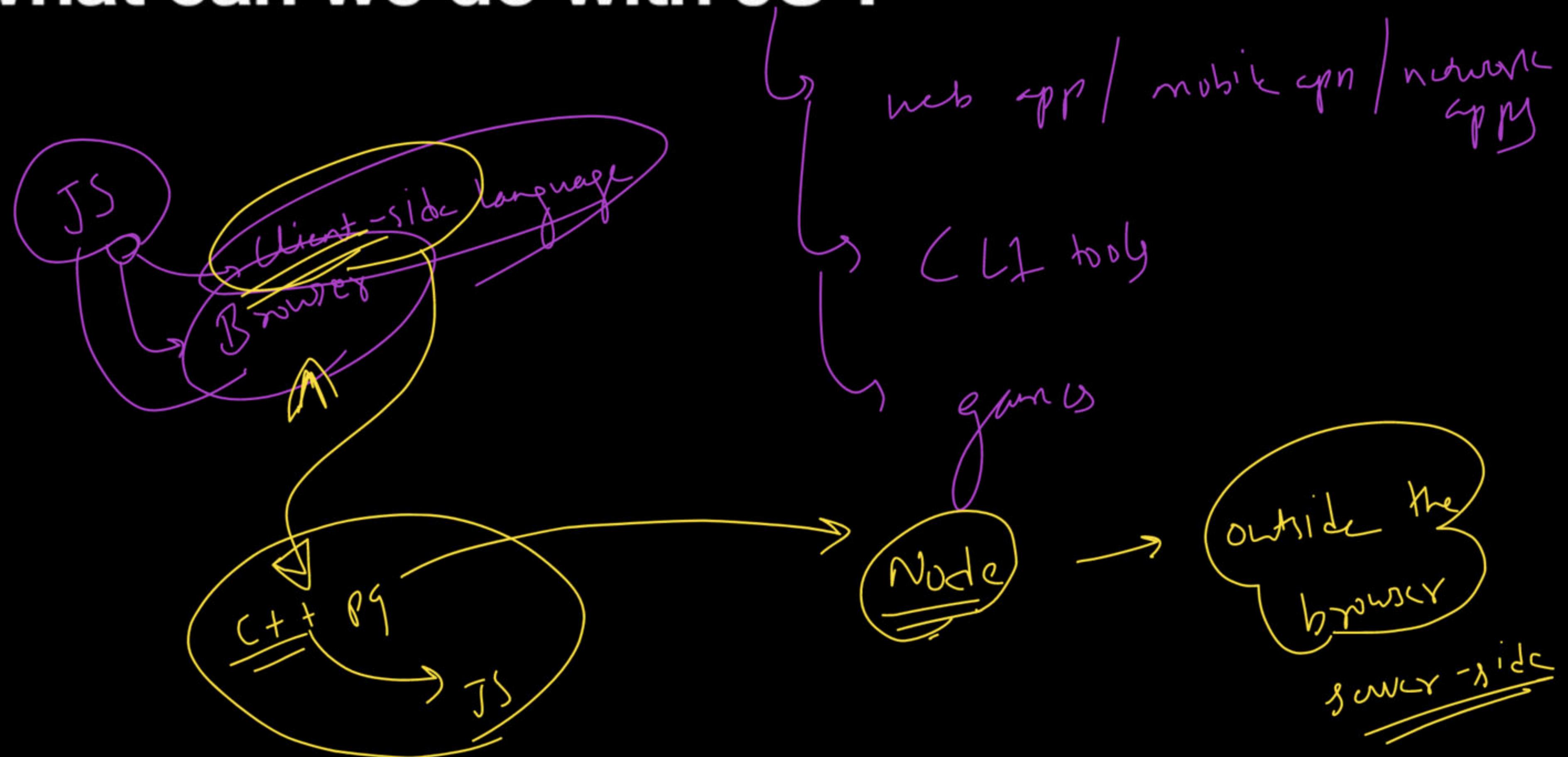
NetSlap

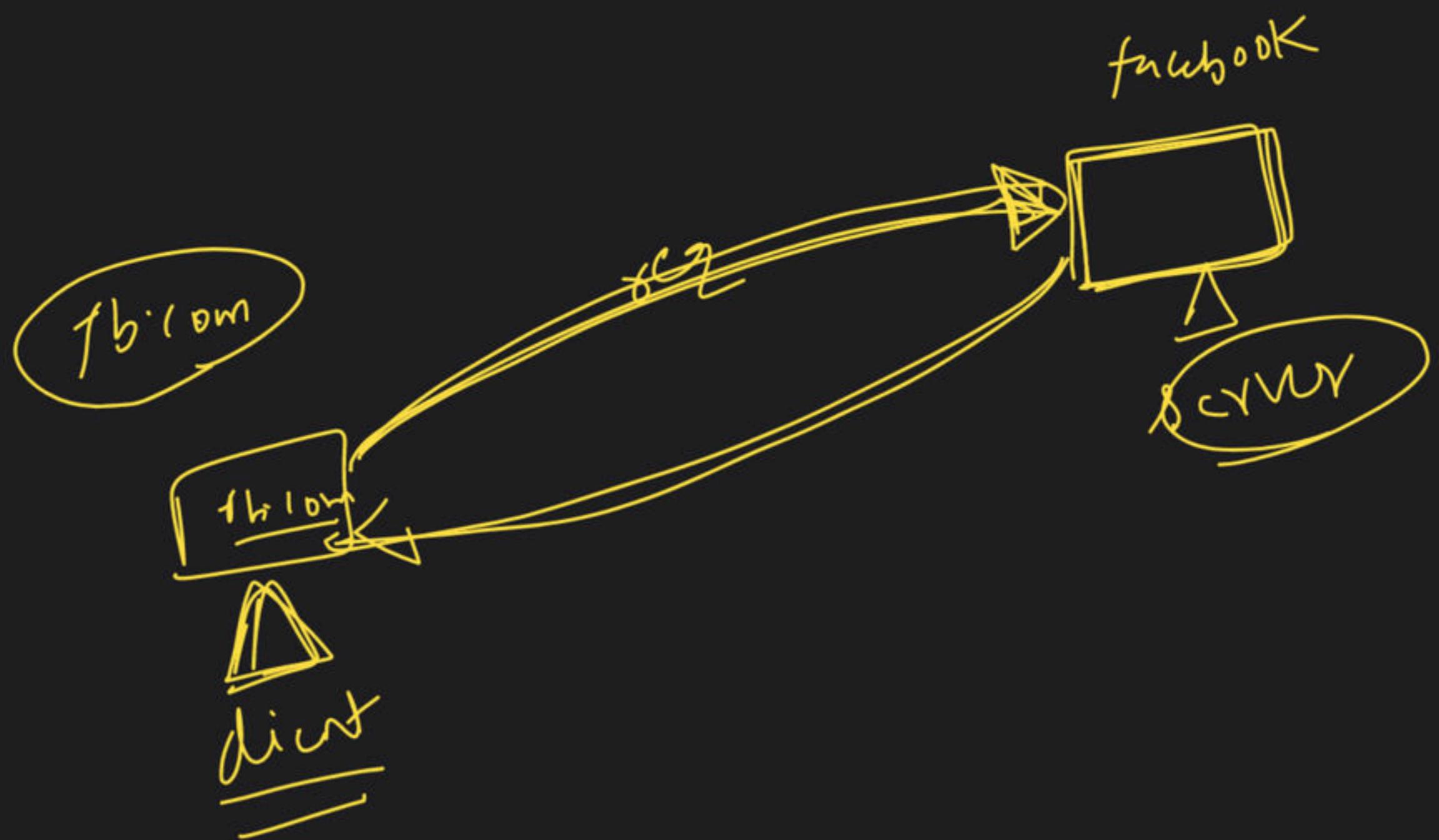
10 days

1995

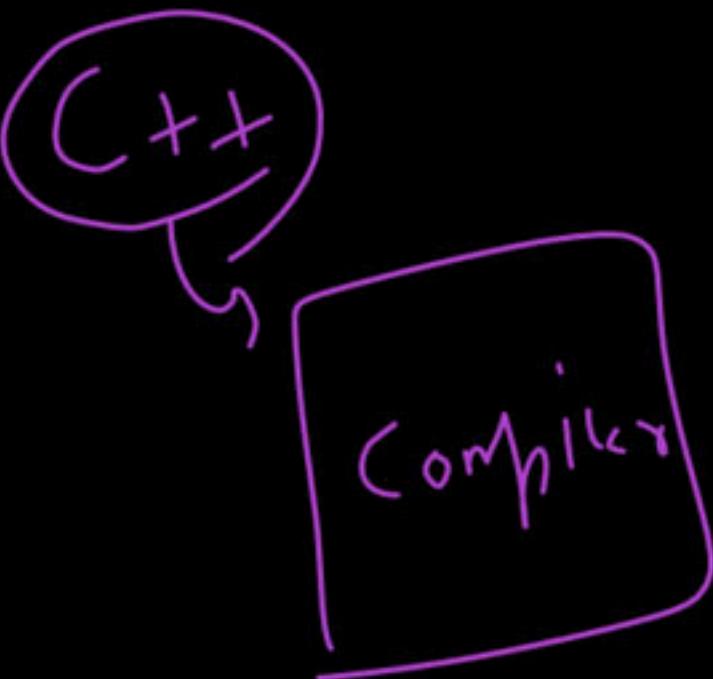
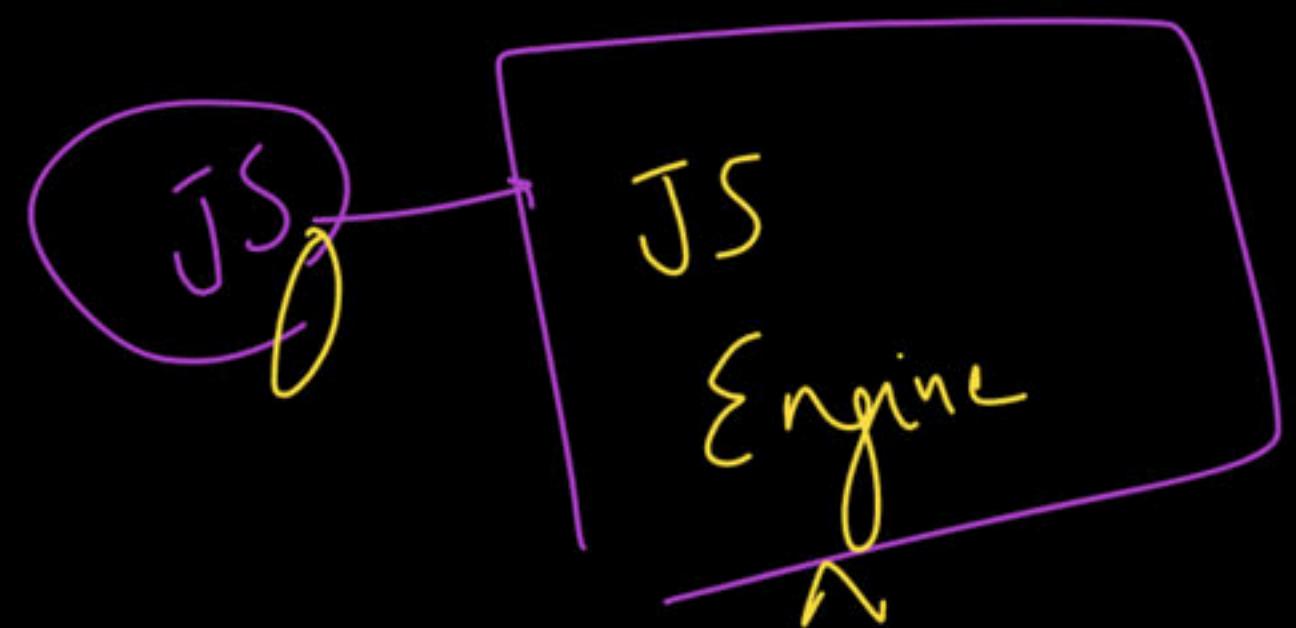
Mohua

What can we do with JS ?





Do we need a Compiler ?



SpiderMonkey

V8

JS Setup ?

(Code Editor) / IDE

VS Code

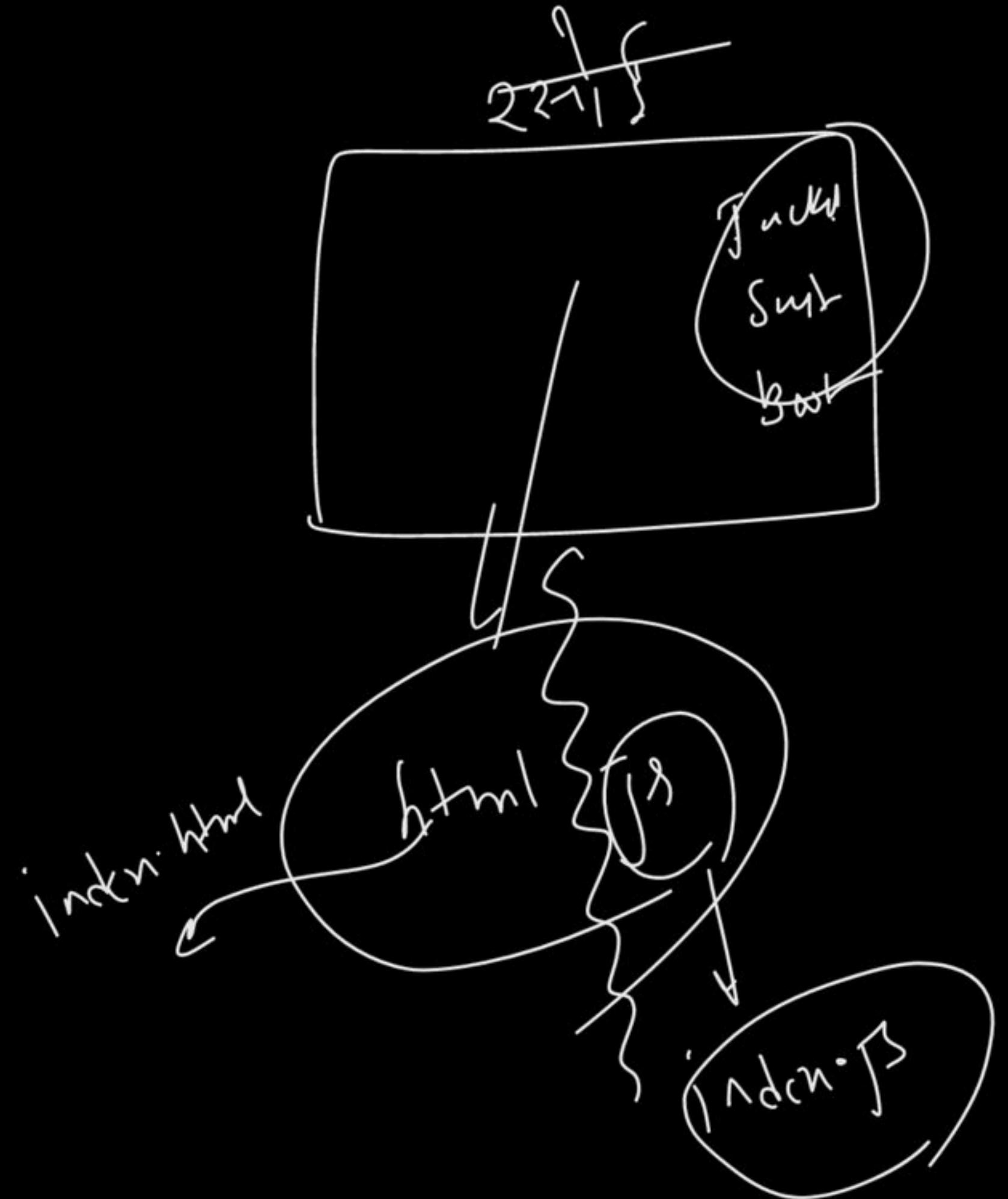
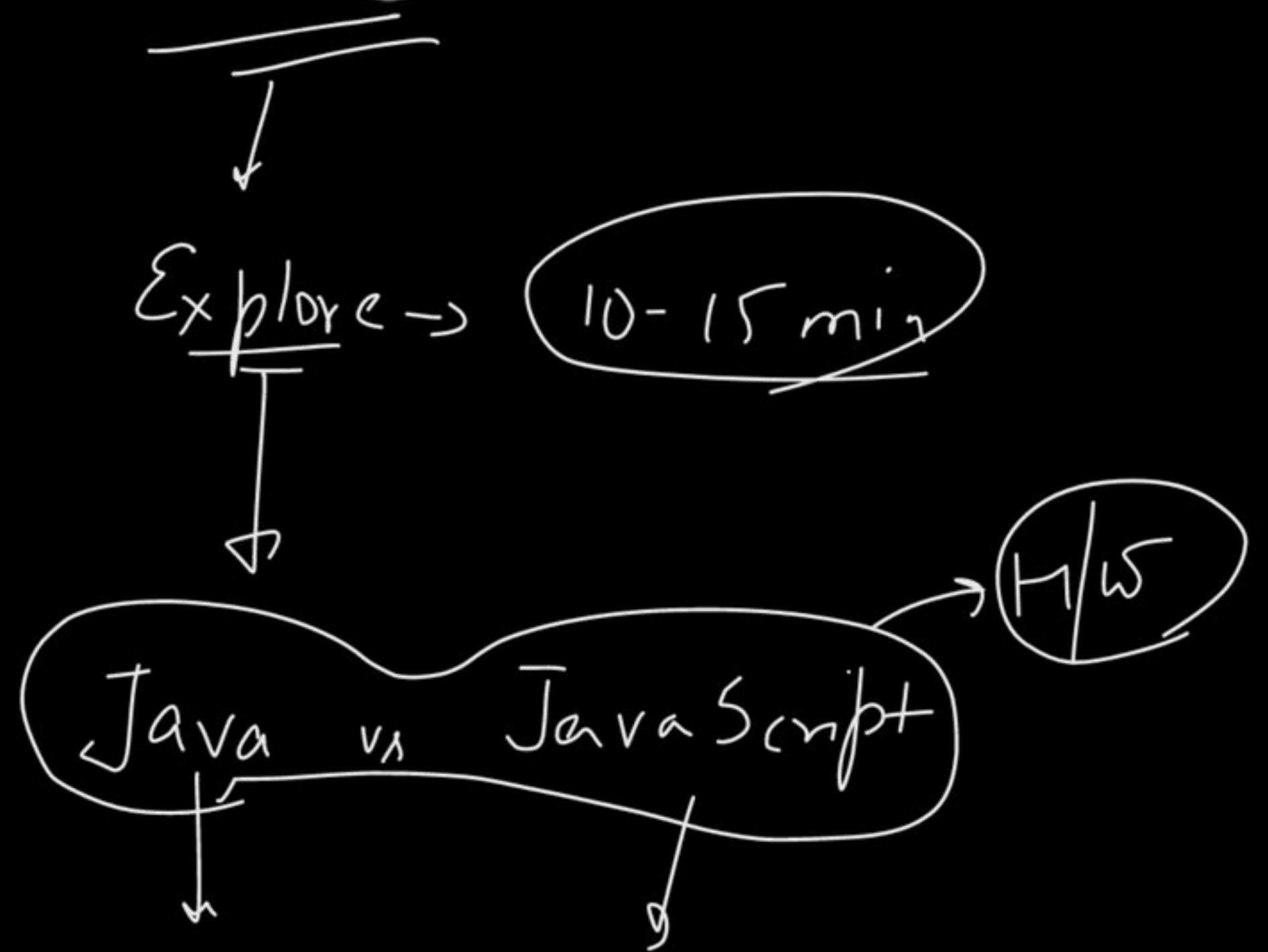
Sublime

Moon

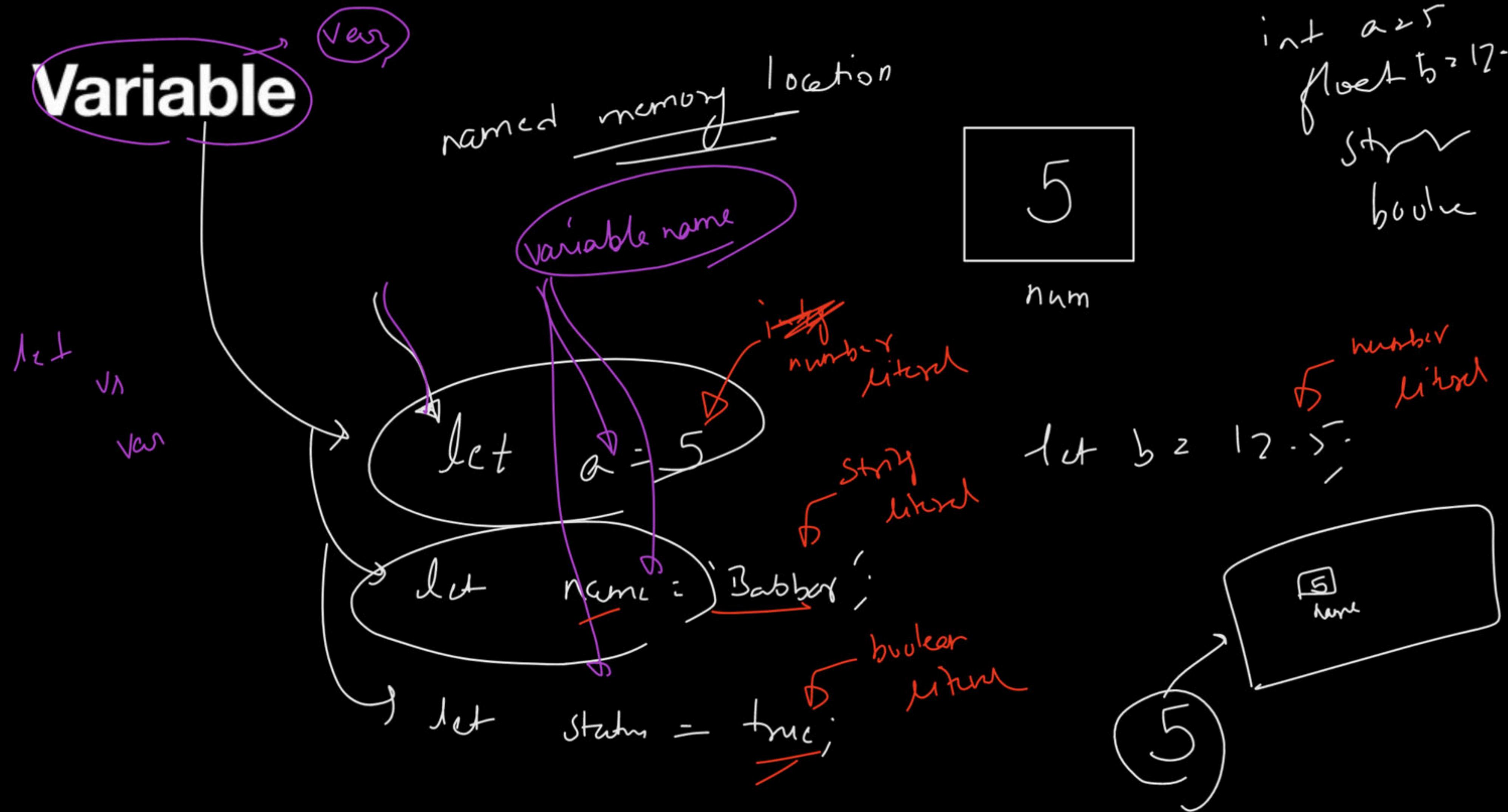
① VS Code → Install

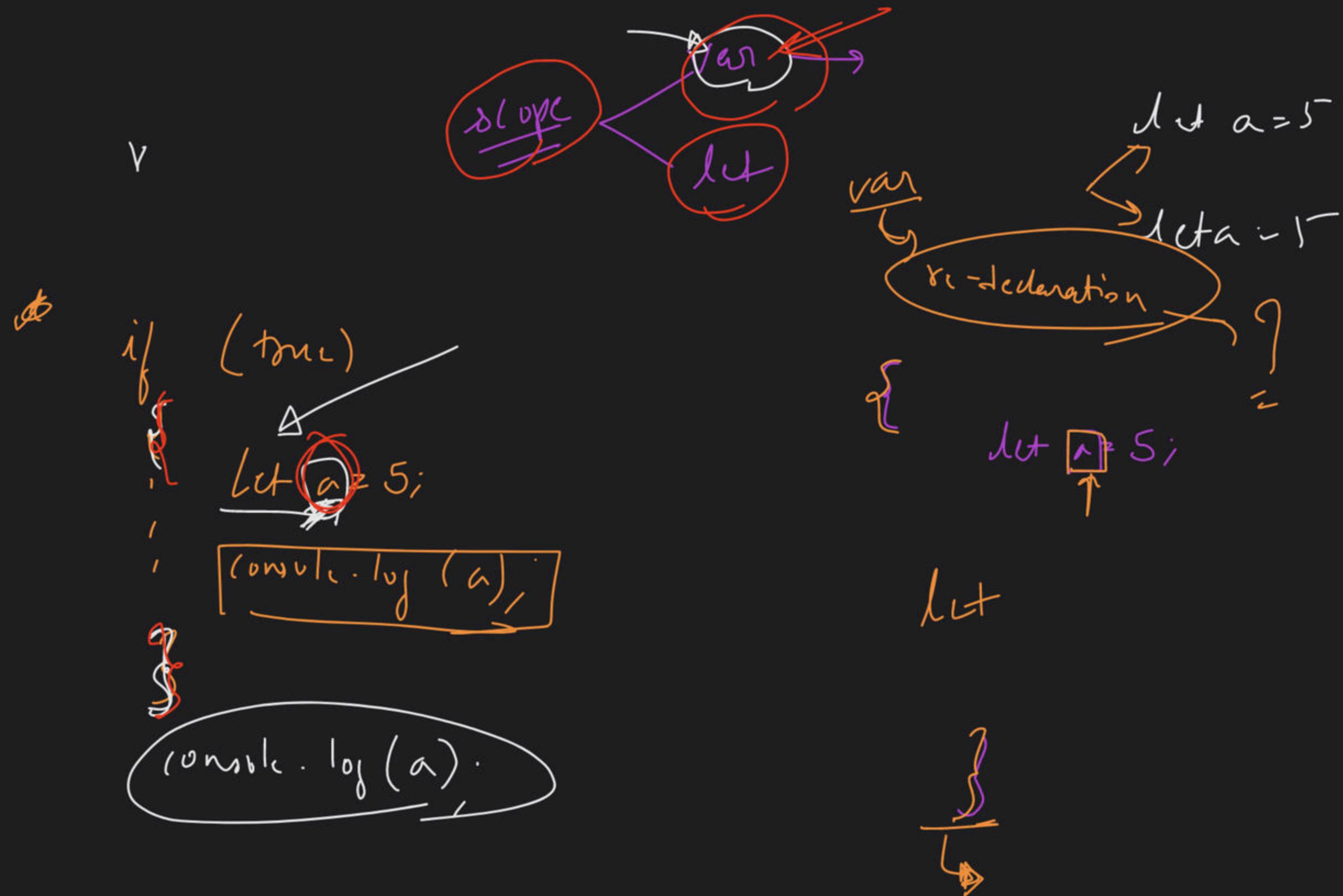
② Node.js install

Adding JS in code



Variable





variable naming :-

Rules

cannot be a reserved keyword

(if, else, var etc)

first Name

age of student

love Bishan

r

Rahul Sharma

meaningful

cannot start with a no.

contain contain space or --

Camel case

Anuj Kumar Gupta

let fact = 5

variable name

let name = love;

if ()

let if = 5;

let [a] = 5

let $\underline{a \neq b};$

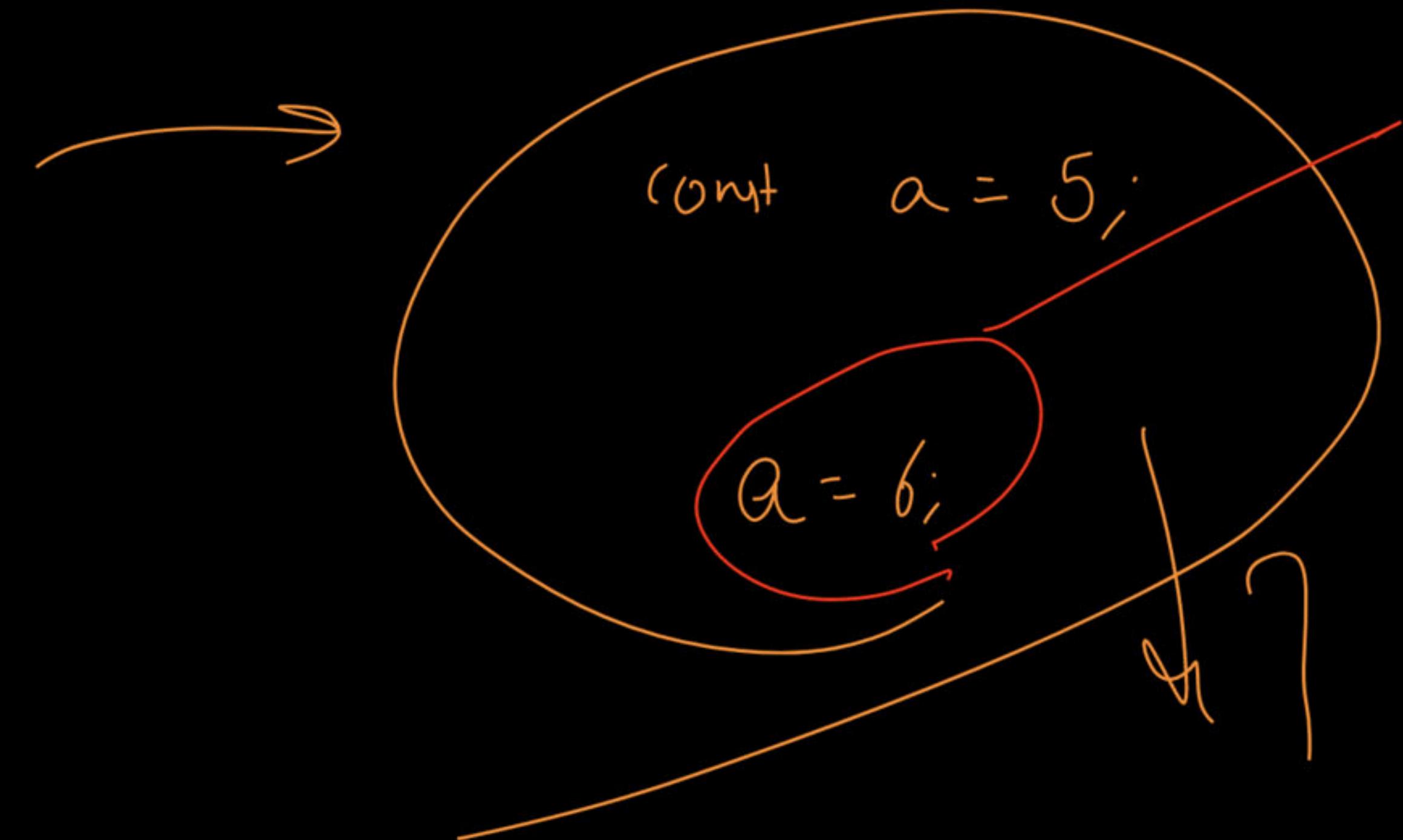
let $a = 1 \wedge b = 2;$

let $a \geq 1;$

let $b \geq 2;$

Constants

\equiv



Primitive Types

- String
- Number
- Boolean
- Undefined
- Null

'Love Barber' → sequence of character

1, 2, 3, 4, 5, -

1.23
12.01

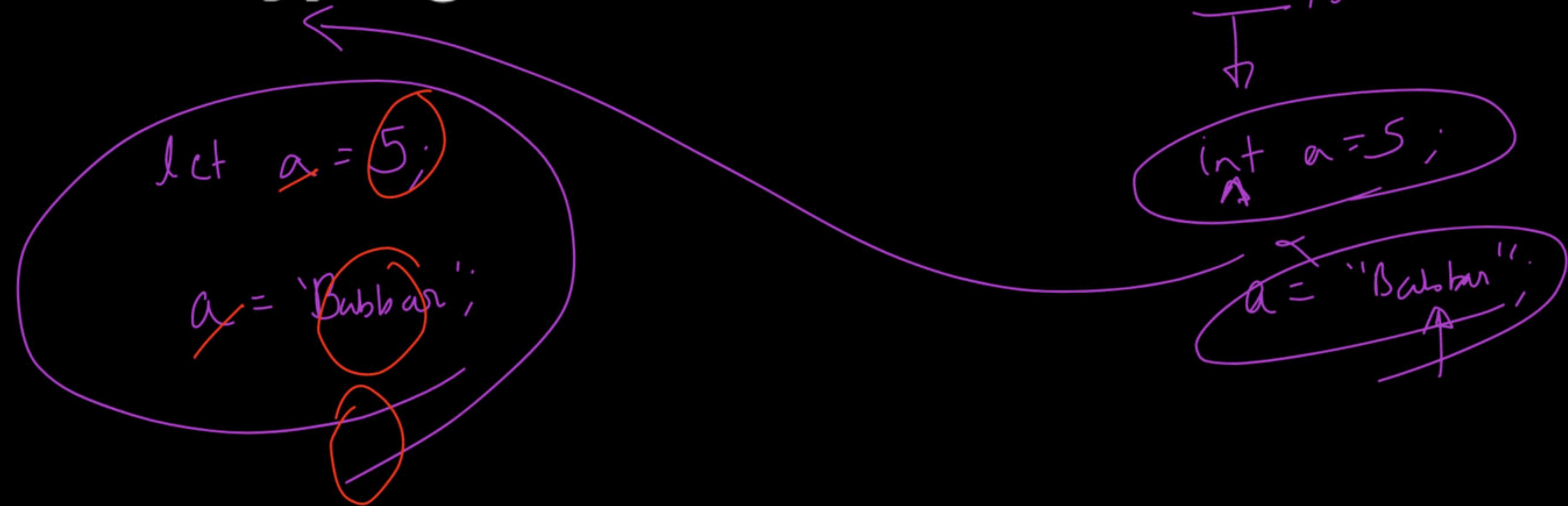
true or false

let a;

console.log(a);

0 | \varnothing undefined

Dynamic Typing



Reference Types

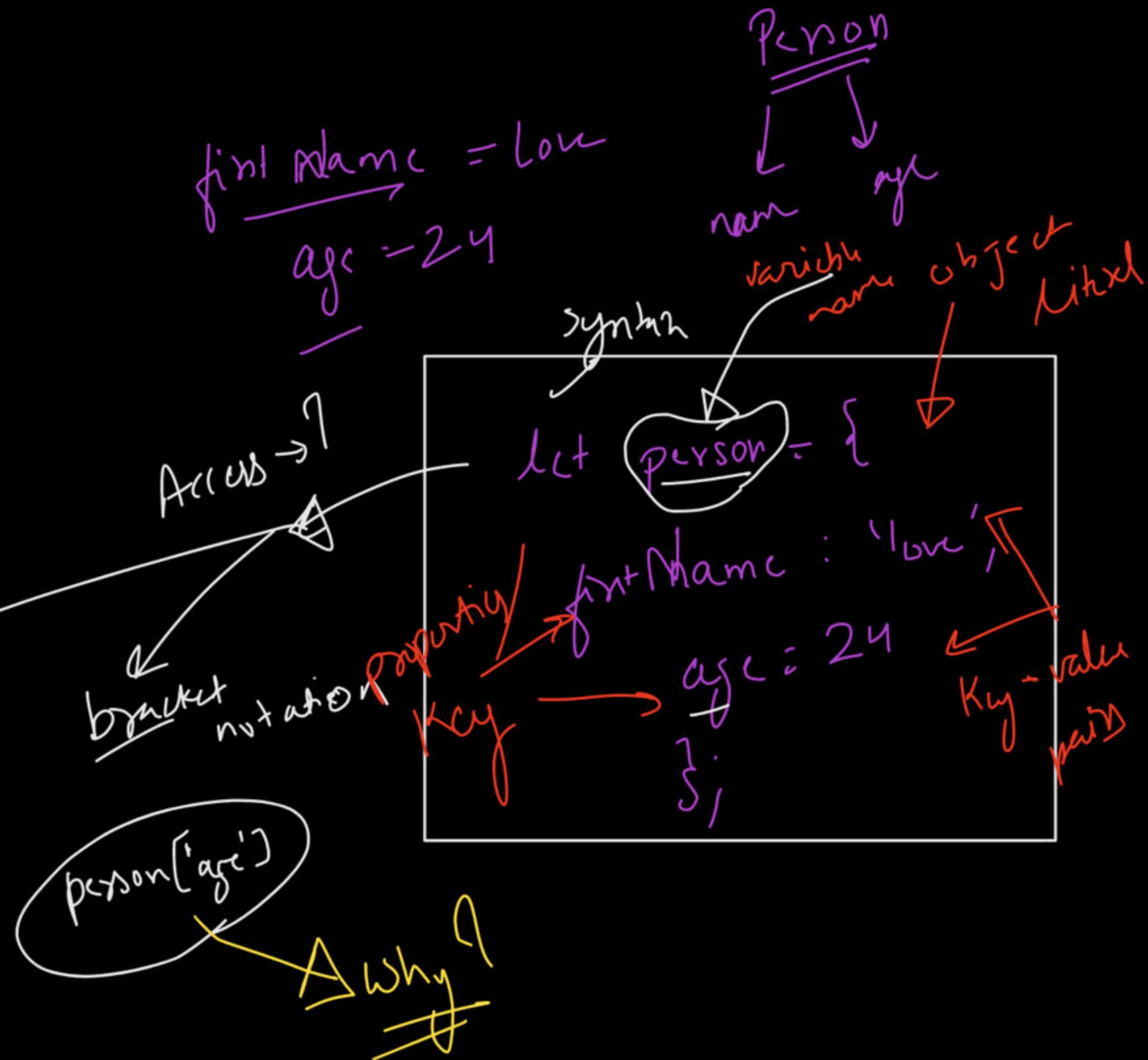
- The diagram illustrates the relationship between objects, arrays, and functions, and how they relate to dot notation.

 - Objects
 - Arrays
 - Functions

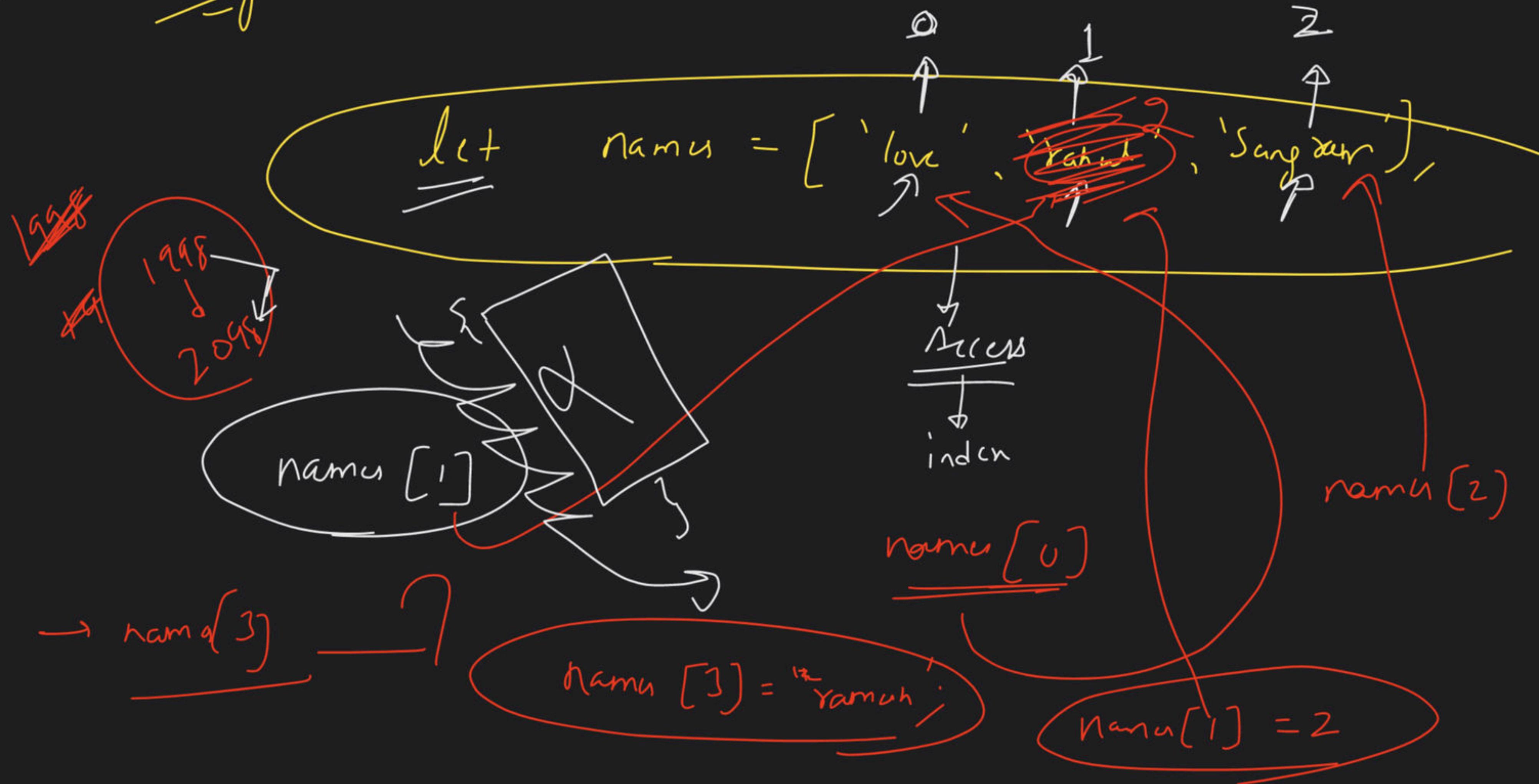
Two ovals at the top represent "Objects" (purple) and "Functions" (red). Arrows point from both to a larger oval below representing "dot notation". A yellow arrow points from the bottom right towards the "dot notation" oval.

dot notation

functional



→ Arrays - D's used to contain a list of items.



$\text{if } \text{count} > 5$

~~name~~

~~name~~

if audio issue

then

Reload

or

Rejoin

function :-

skip

1 min

Break

Operators

- Arithmetic
- Assignment
- Comparison
- Bitwise
- Logical

Arithmetic

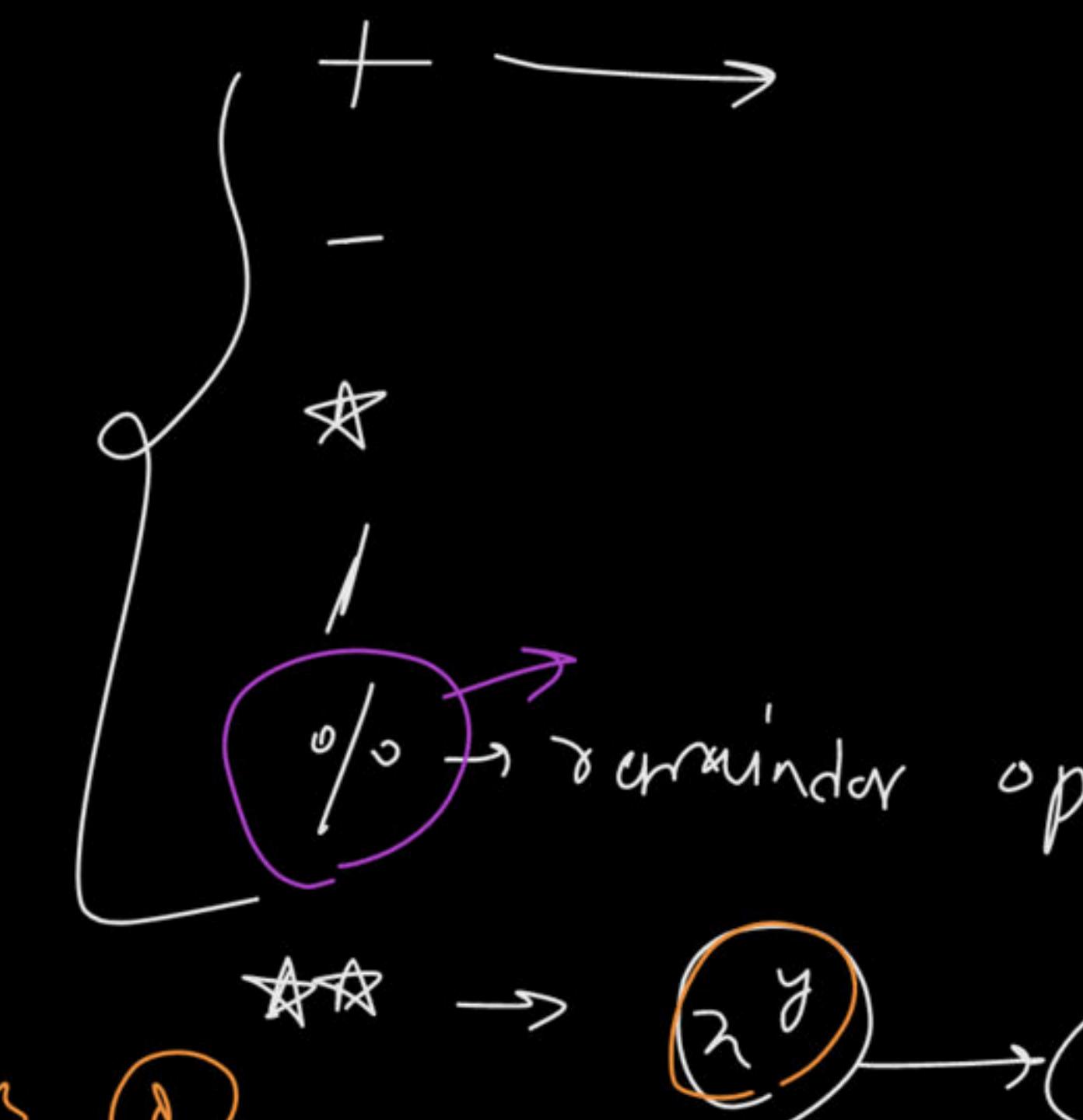
$$\text{let } a = 2$$

$$\text{let } b = 3$$

convolution $(2 \star 3)$

$$8 \cdot 2^3 = 8$$

$$2 \star \star 4 = 2^4 = 16$$



$$\text{let } a = 12$$

$$\text{let } b = 5$$

convolve $a \cdot b$

$$12 \cdot 5 = 1$$

$$12 \cdot b = 2$$

$$12 \cdot 1 = a + b$$

$$\boxed{\text{convolve-}1_0(c)} \rightarrow 13$$

$$12 \cdot 1.5 = 2$$

$$\frac{2\sqrt{5}}{4}^2$$

~~1~~ rem

$$5 \cdot 12$$

$$= 1$$

Pre / Post

Inc / dec

Operator



let $n = 10;$

console.log($++n$);

$+1 n;$

pre increment

first

increment
the
value

secondly

wr the
value

A rectangular box containing the letter 'a'. To its right, the number '6' is written. A horizontal arrow points from the top of the box to the number. Next to the box is a green circle containing a plus sign '+'. Above the box, the text 'data = 6;' is written.

console.log($a++$);

6 < print

(onvalog(a))

(+)

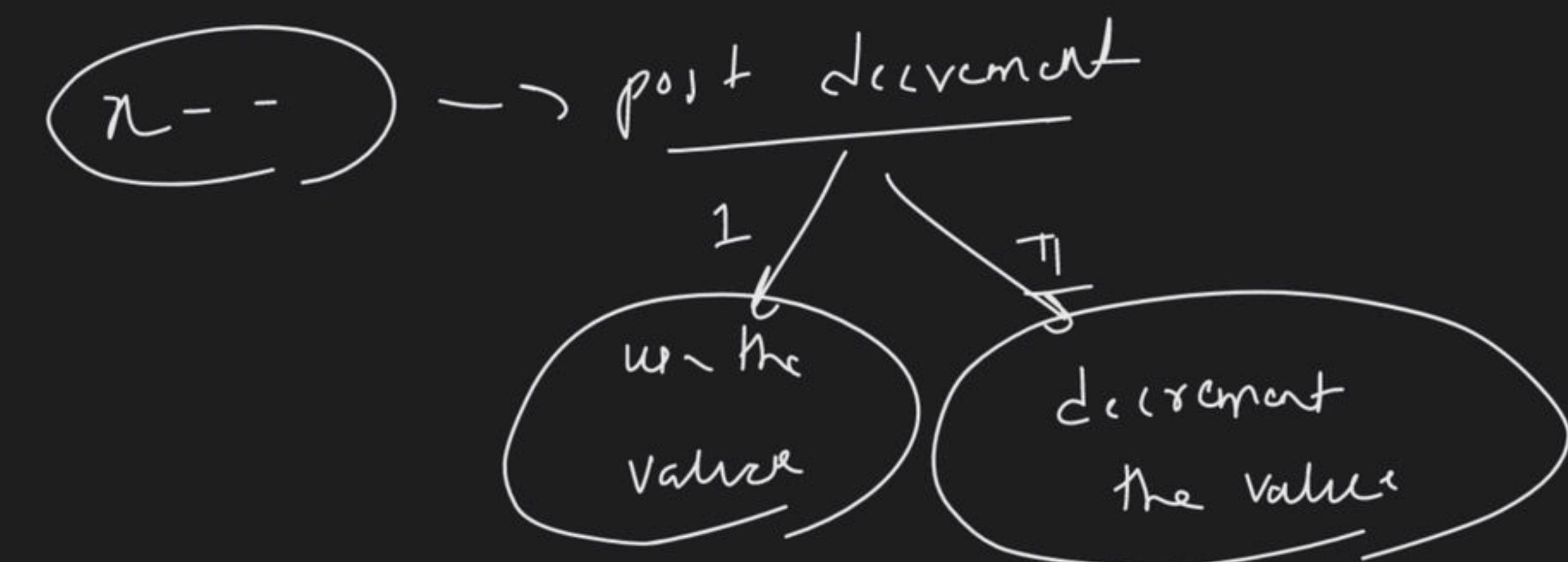
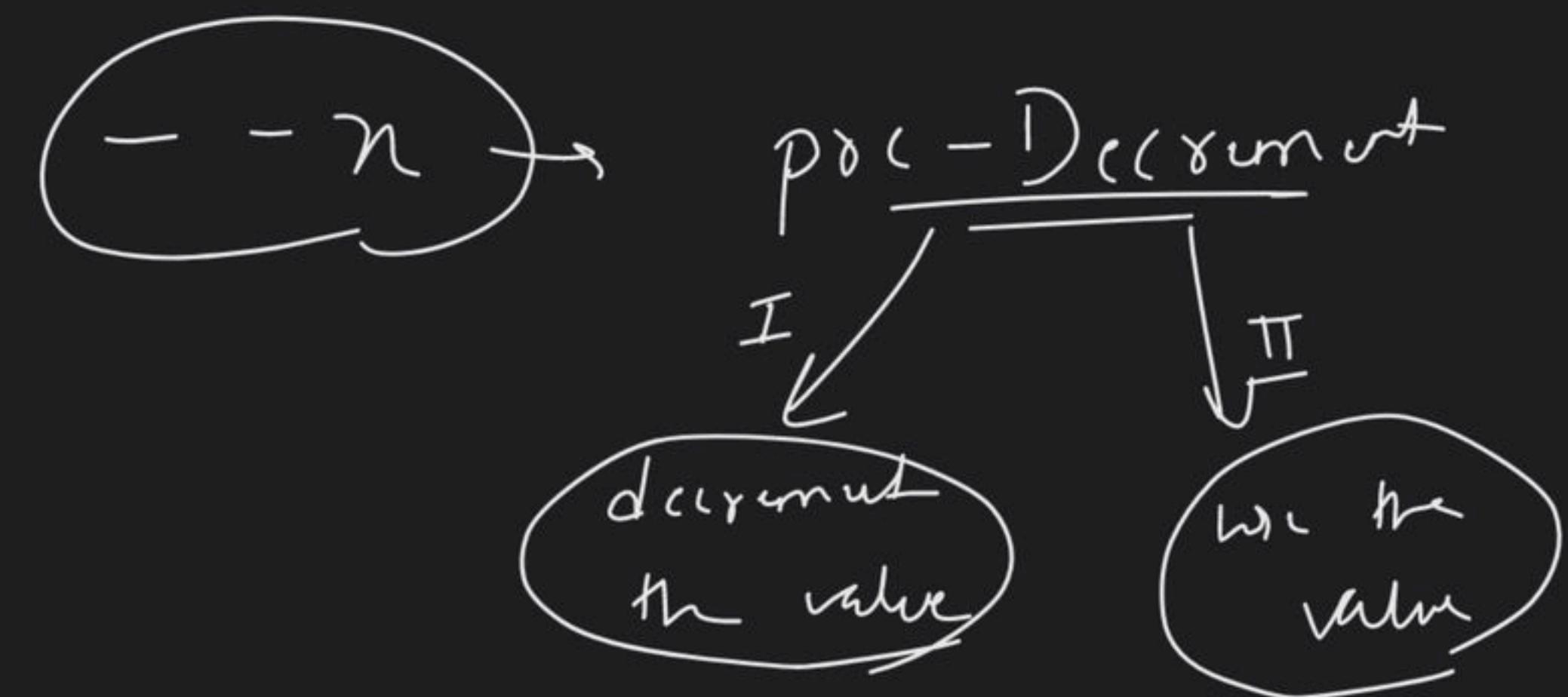
$n++$ → post increment
operator

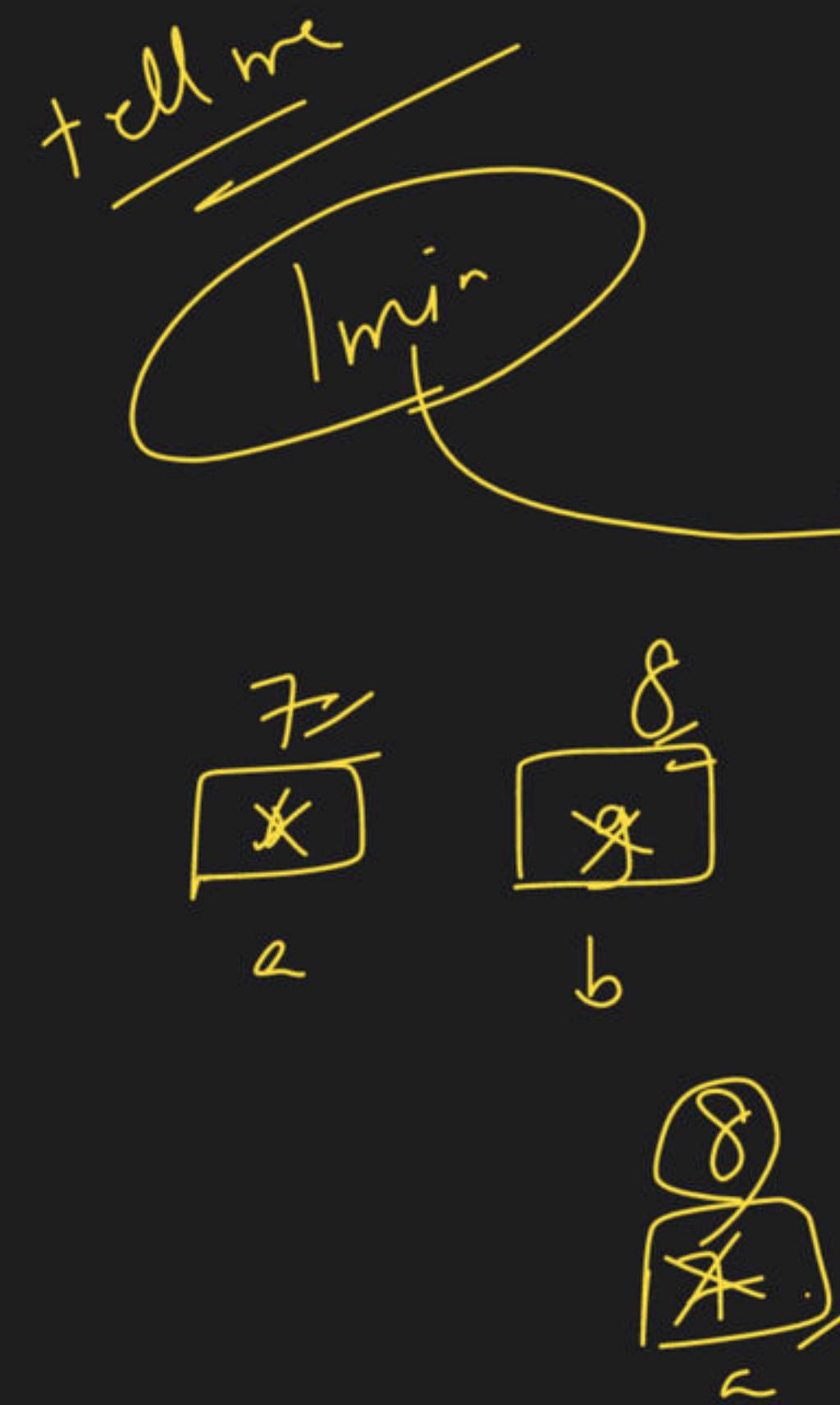
+

wr the
value

++

increment
the
value





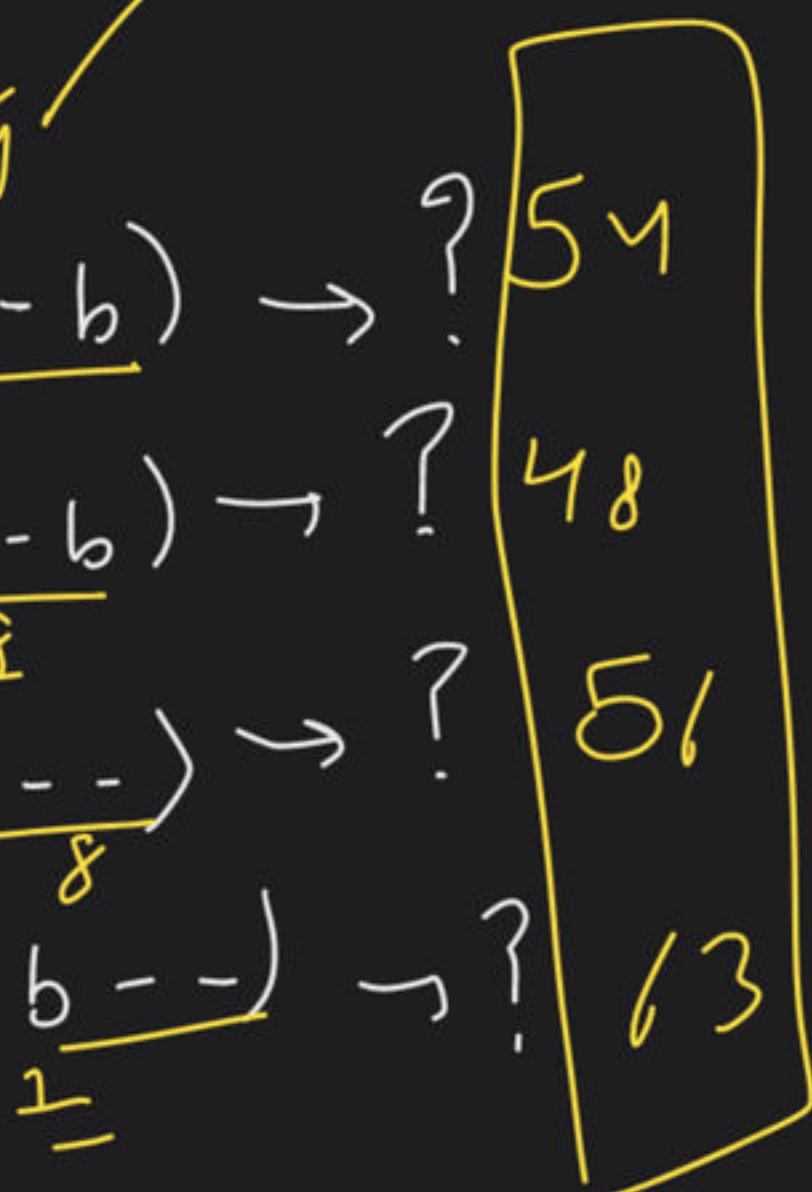
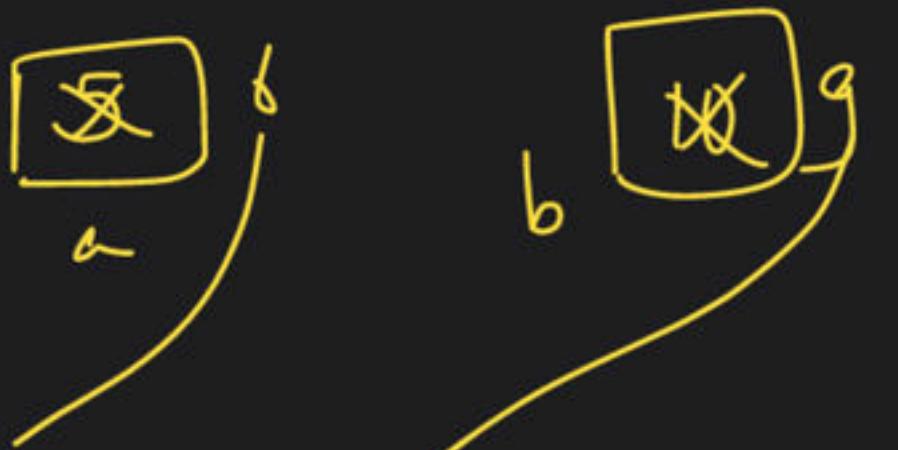
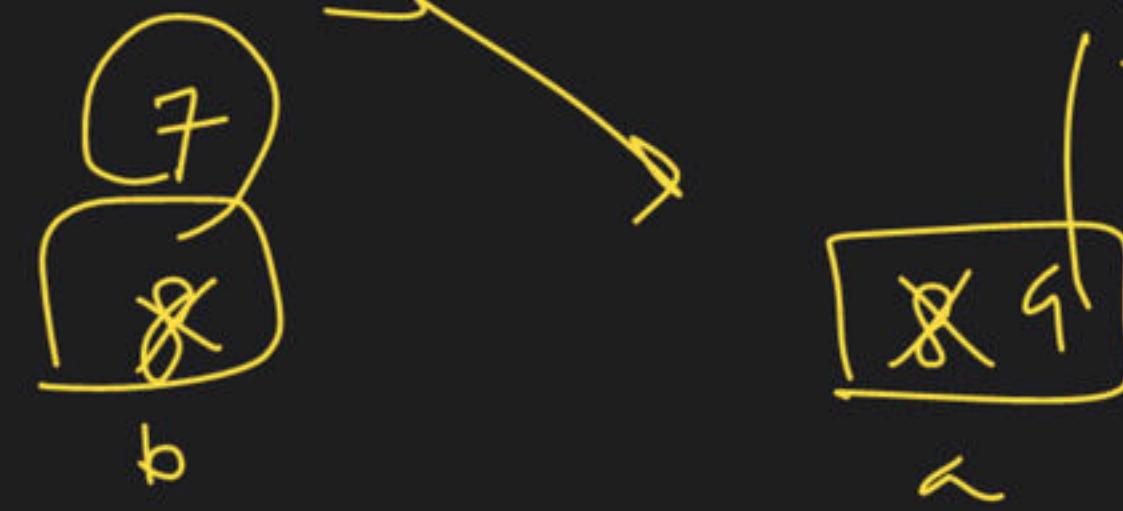
$$l_1 + a = 5$$

$$l_1 + b = 10;$$

$$\text{let ans 1} = \left(\frac{1+a}{6} \right) * \left(\frac{g}{-b} \right) \rightarrow ? \begin{cases} 5 \\ 48 \end{cases}$$

$$\text{let ans 2} = \left(\frac{a+7}{6} \right) * \left(\frac{g}{-b} \right) \rightarrow ? \begin{cases} 51 \\ 63 \end{cases}$$

$$\text{let ans 3} = \left(\frac{a+7}{7} \right) * \left(\frac{b-g}{8} \right) \rightarrow ? \begin{cases} 51 \\ 63 \end{cases}$$



Assignment

let num = 53;

num = num / 2;

num /= 2;

n = n + 5

n += 5

g

let a = 5;

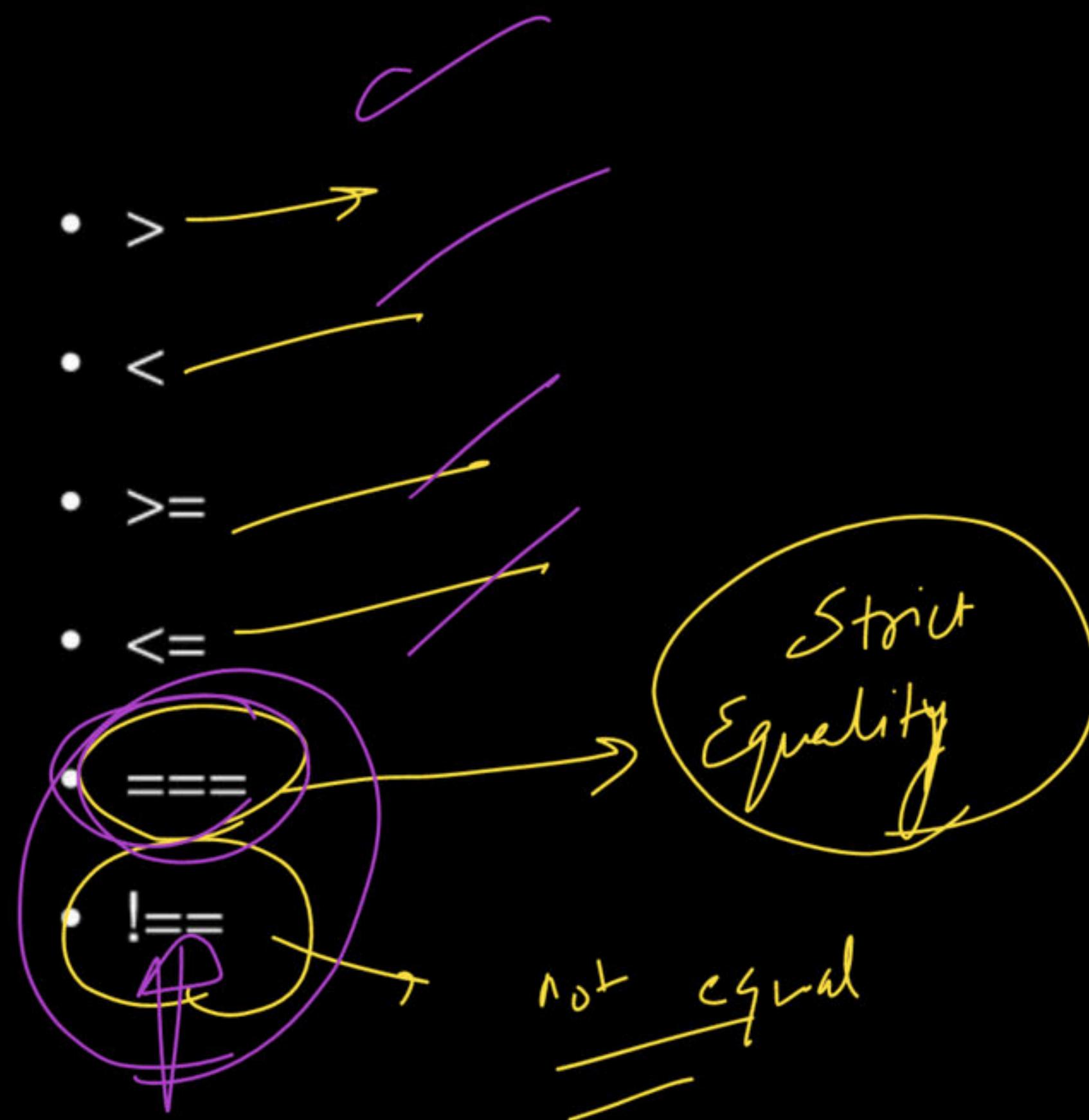
5
a

a = a * 3

↓

a *= 3

Comparison



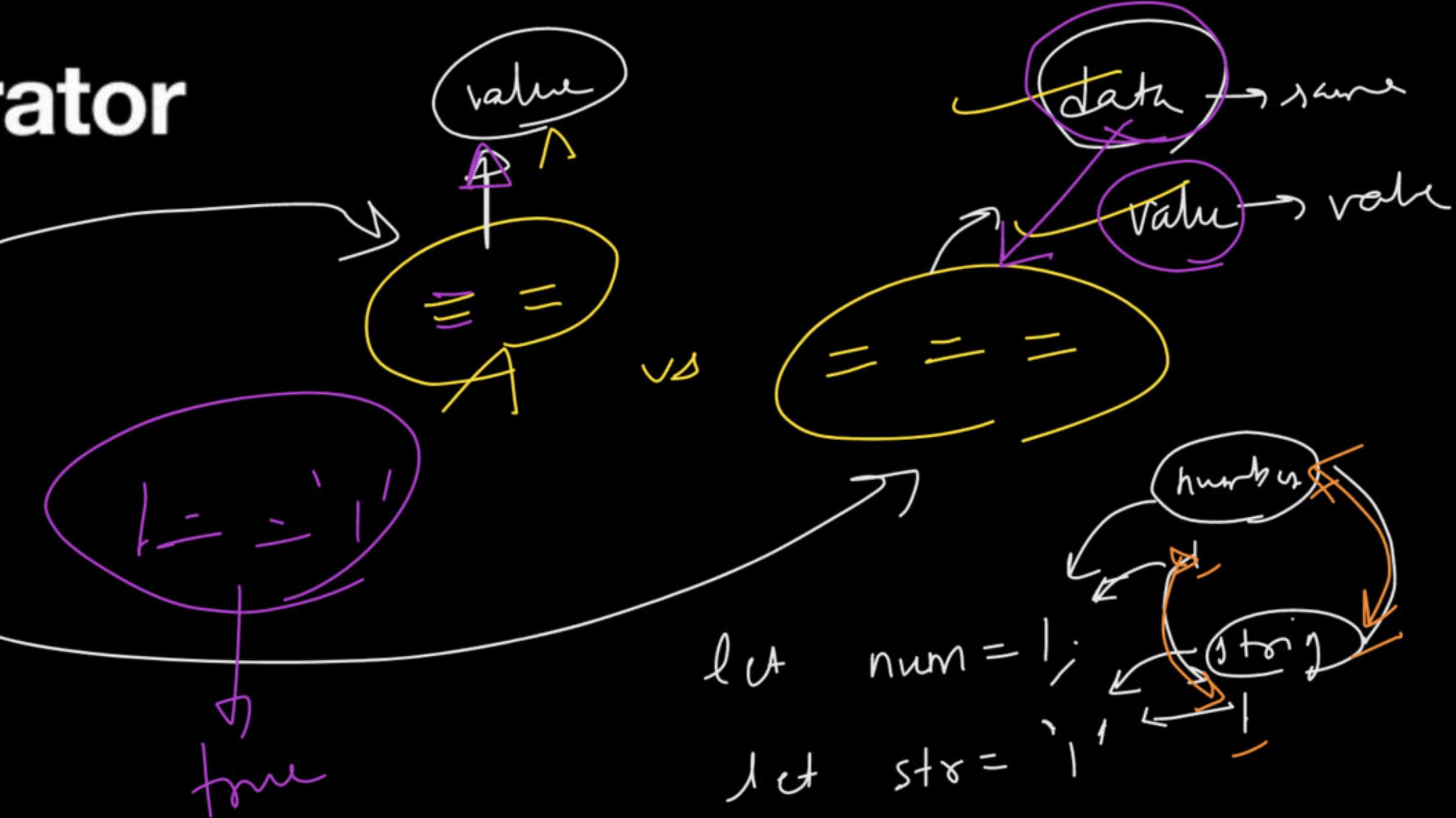
$5 > 3 \rightarrow$ true

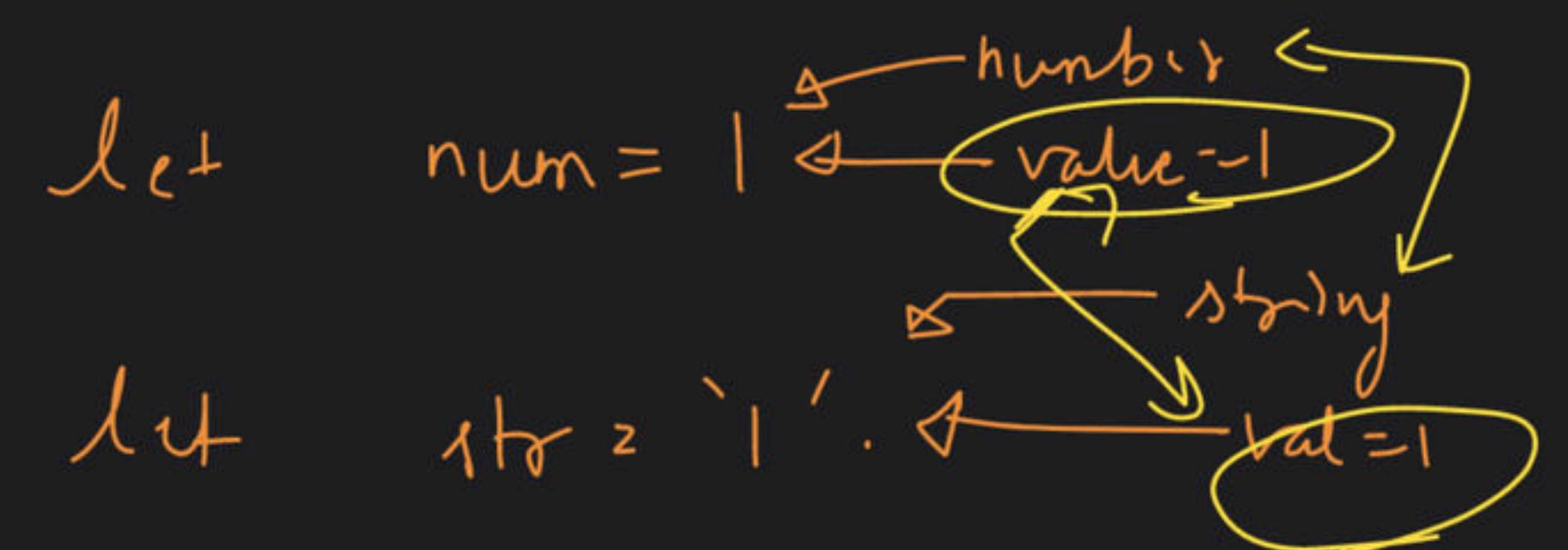
$5 < 3 \rightarrow$ false

not equal

Equality Operator

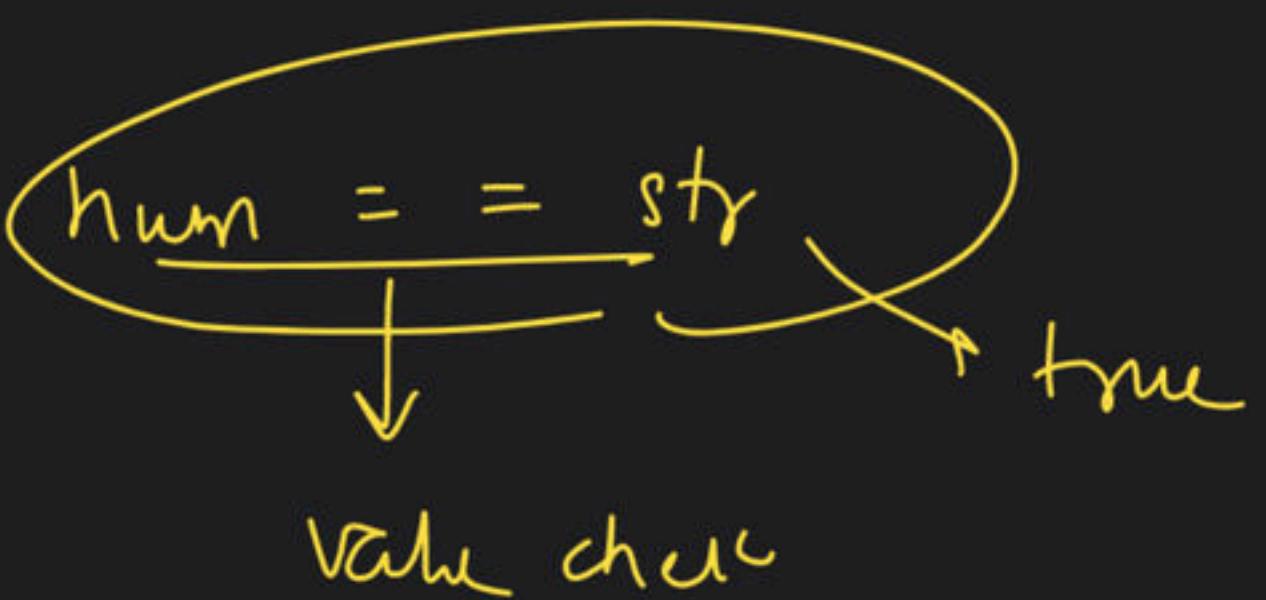
- Loose equality
- Strict Equality





`num == str`

The diagram shows the comparison `num == str`. A bracket indicates that the result is `false`.



Ternary Operator

$$a > b \begin{cases} T \\ F \end{cases}$$

Syntax

cond ? val1 : val2 ;

age = 15

age = 27

let status = (age >= 18) ? 'I can vote' : 'Cannot Vote';

true

false

but

Logical Operator

- AND

- OR

- NOT

~~all~~

all condition

have to be true

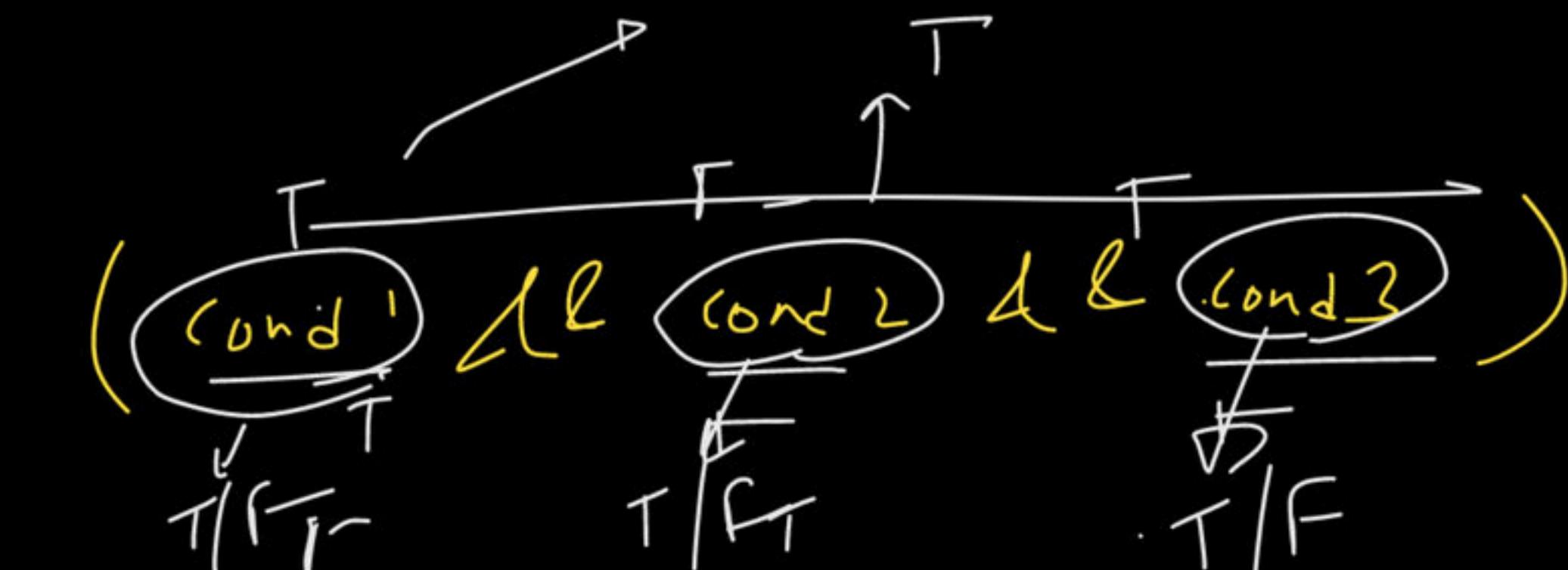
|| → T

any single condition is true

True → False
False → True

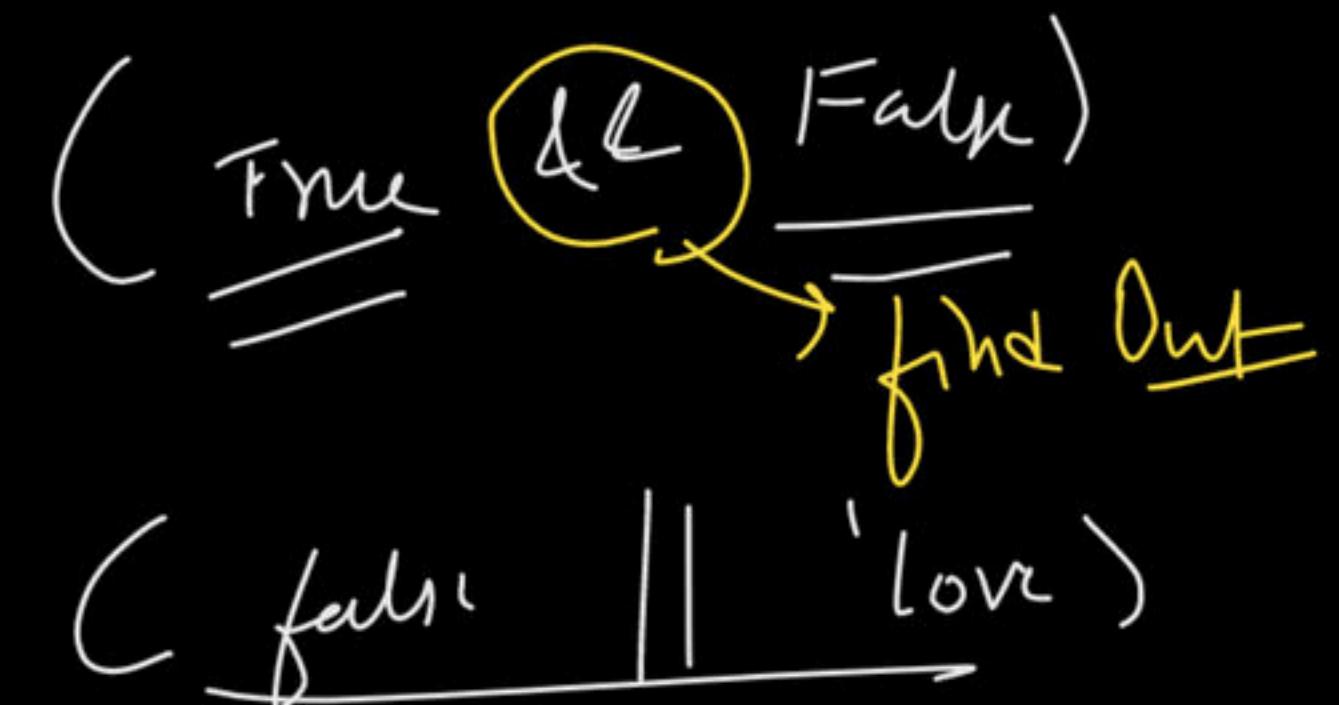
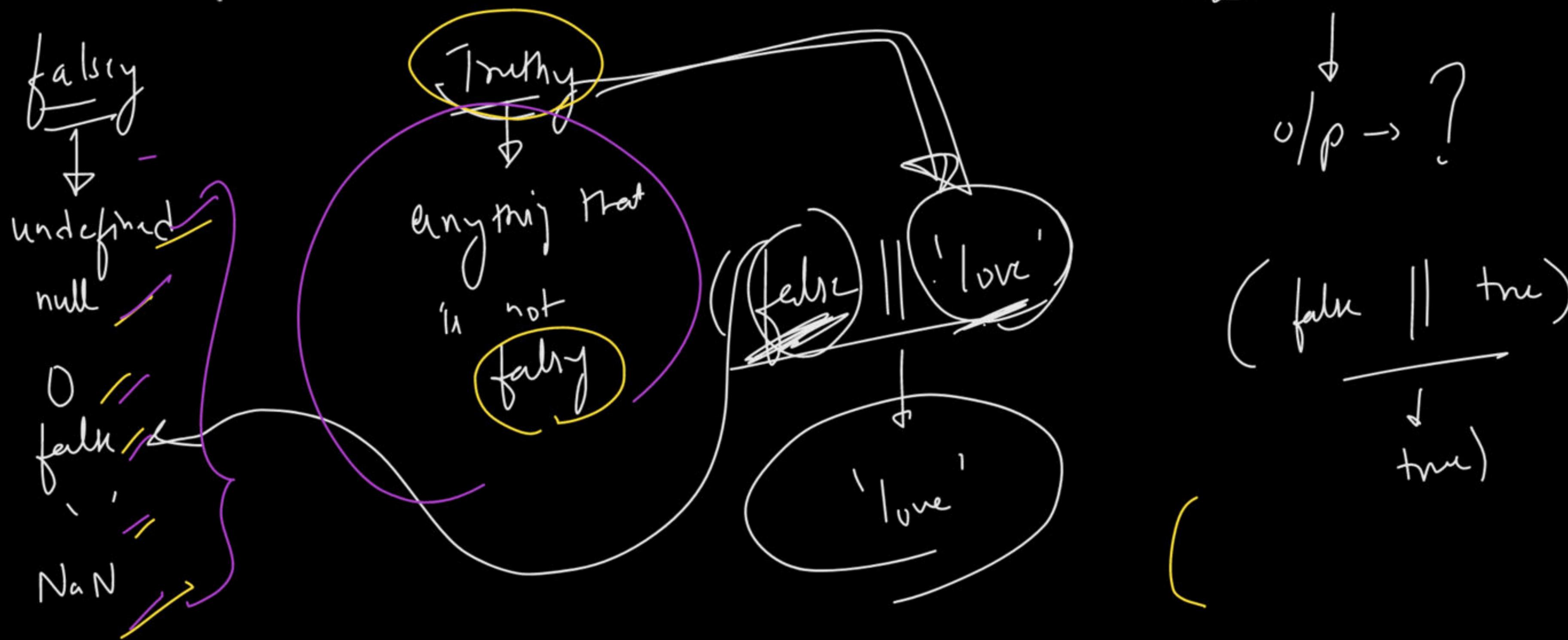
(age > 18 || baliKa vadhu)

A
Shaadi

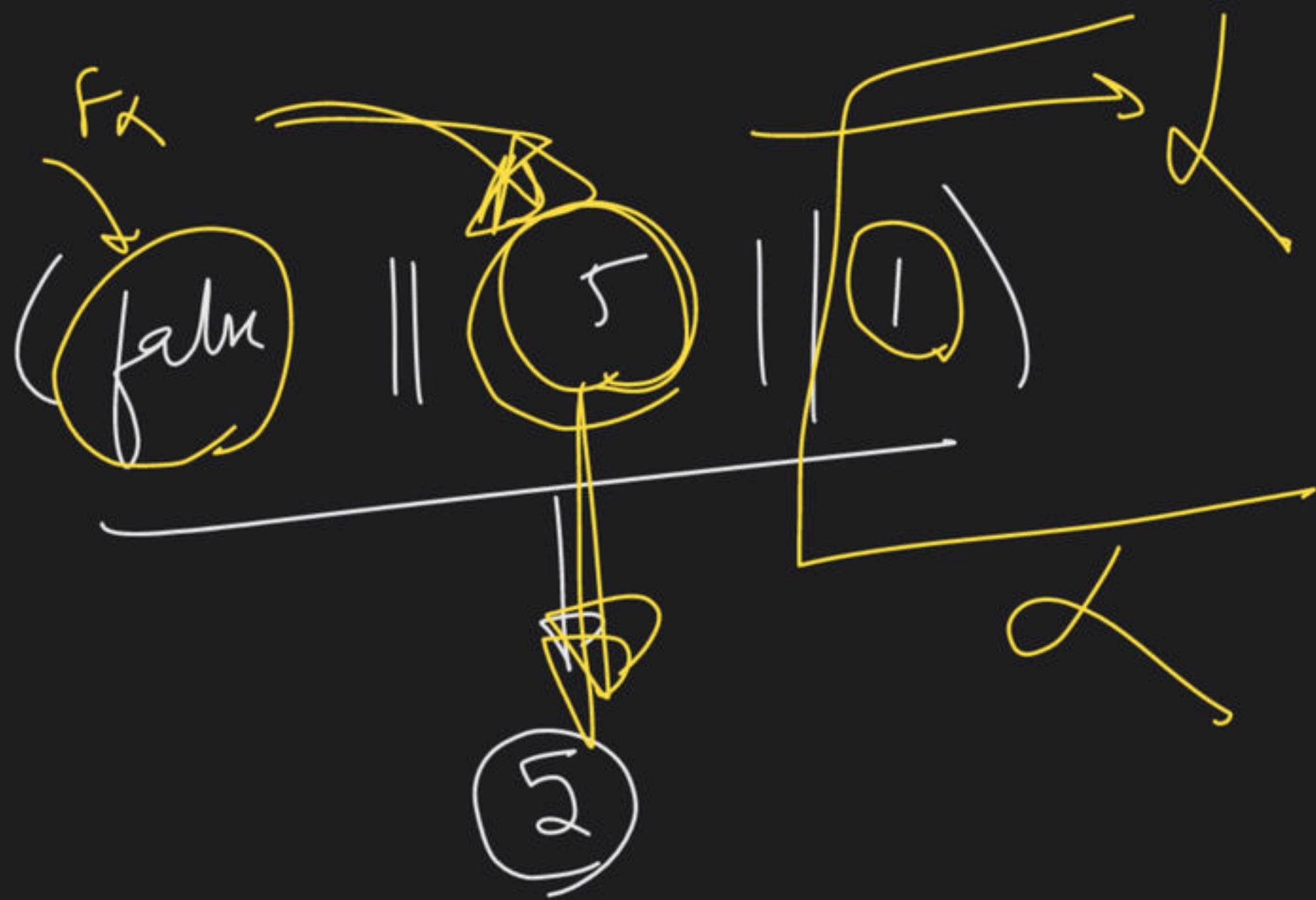


With Non-Booleans

- Concept of False and Truth



OR
Short circuiting



Bitwise Operator

- Bitwise AND

$\&$

- Bitwise OR

$\|$

$a = 2$

$b = 3$

FE

A	B	$\&$
0	0	0
0	1	0
1	0	0



011

a	b	$\ $
0	0	0
0	1	1
1	0	1

00000010

00000011

00000011

&

01

1 → true

0 → false

(2 | 3 → 3)

- 1

Operator Precedence

Brackets

let c = a + b * d / e ;

let c = a + (b * d) / e ;

Control Statements

- If-else
- Switch

month $\geq 90 \rightarrow A$

month $>= 80 \rightarrow B$

else
C

If-elseif-else

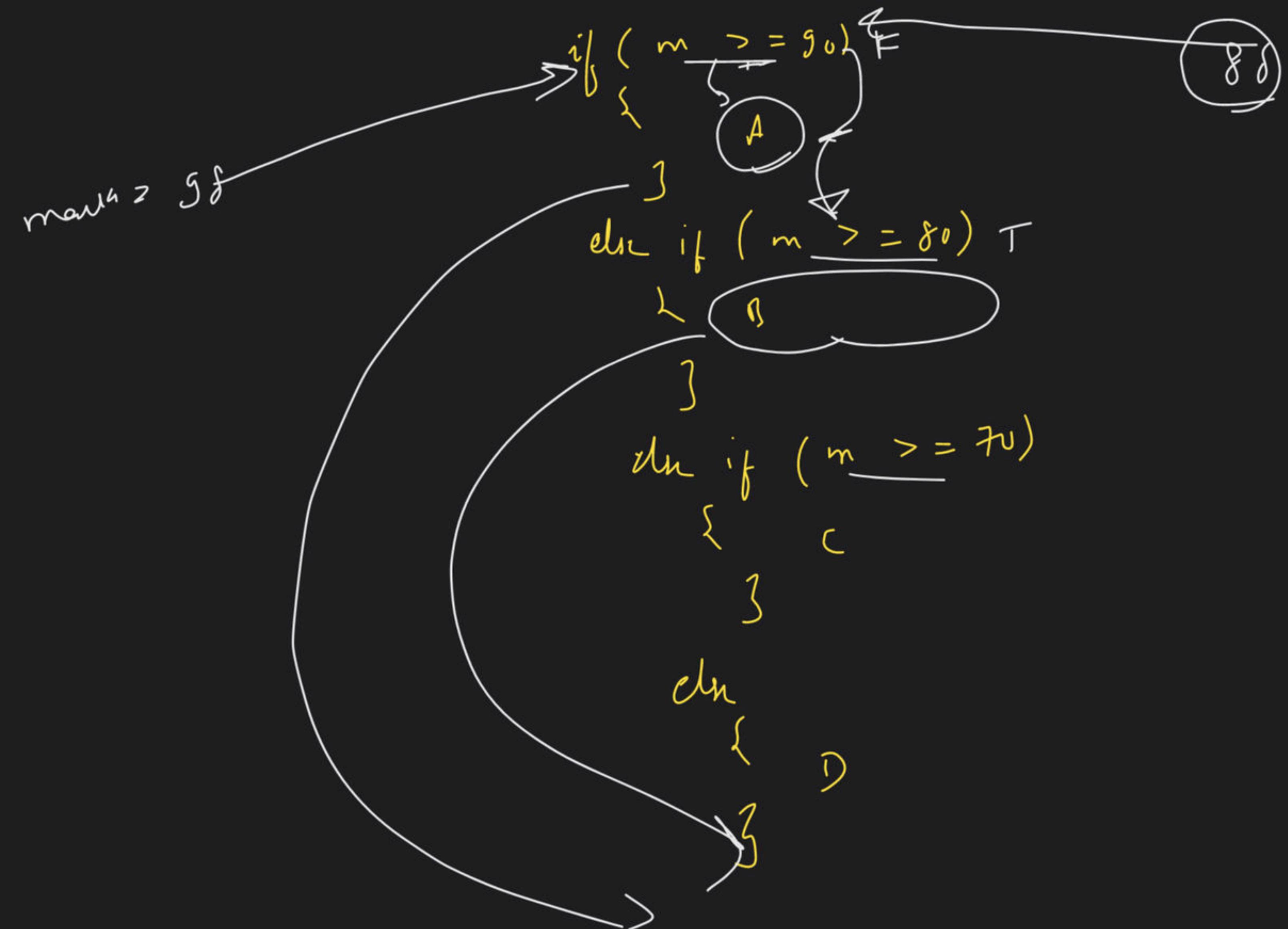
```
if (cond) {  
    // code  
}  
else if (cond) {  
    // code  
}  
else {  
    // code  
}
```

Diagram illustrating the structure of an If-elseif-else conditional statement:

- The outermost curly brace groups the entire structure.
- The first brace groups the `if` block, which contains the condition `(cond)` and a pair of braces for the block body.
- The second brace groups the `else if` block, which contains the condition `(cond)` and a pair of braces for the block body.
- The third brace groups the `else` block, which contains a pair of braces for the block body.

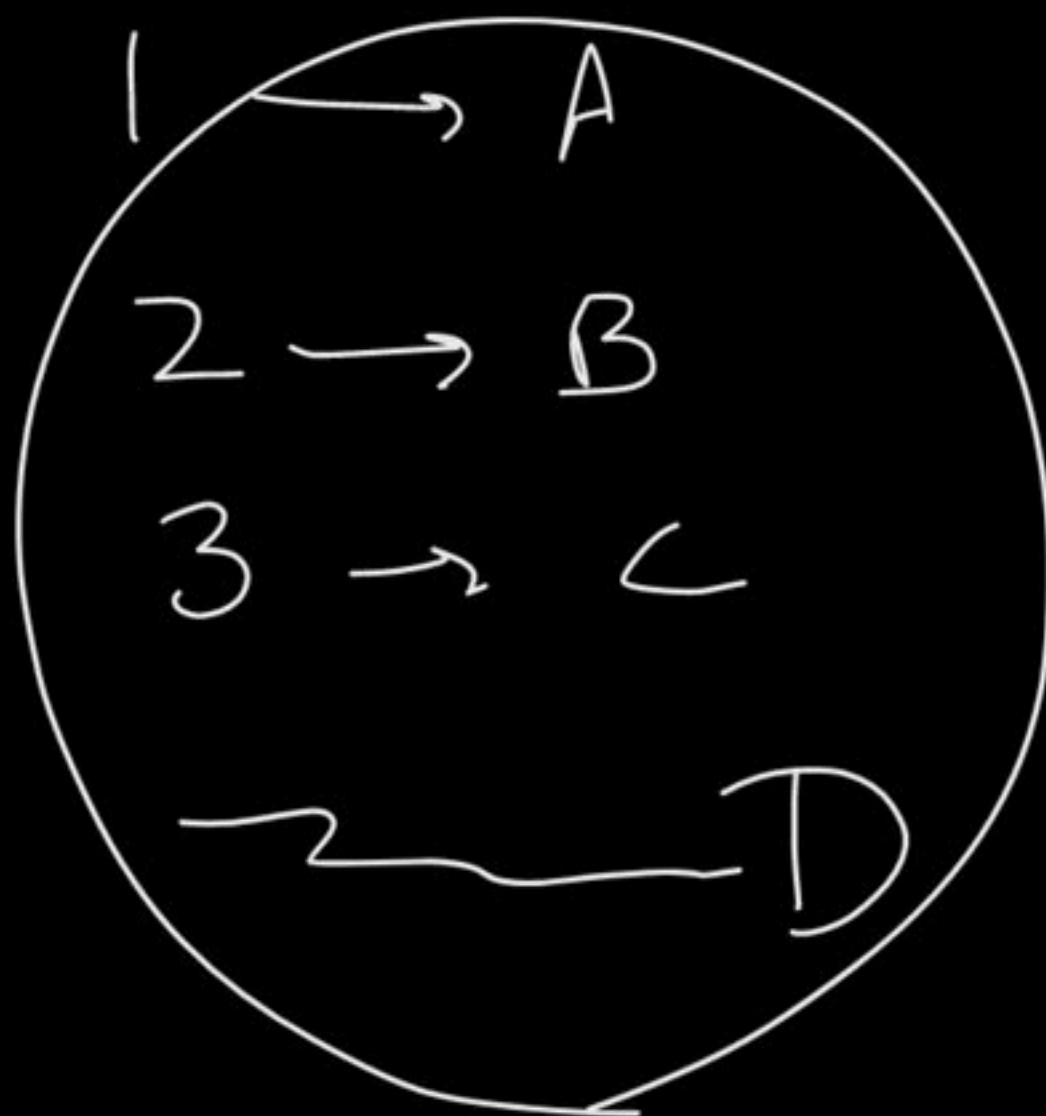
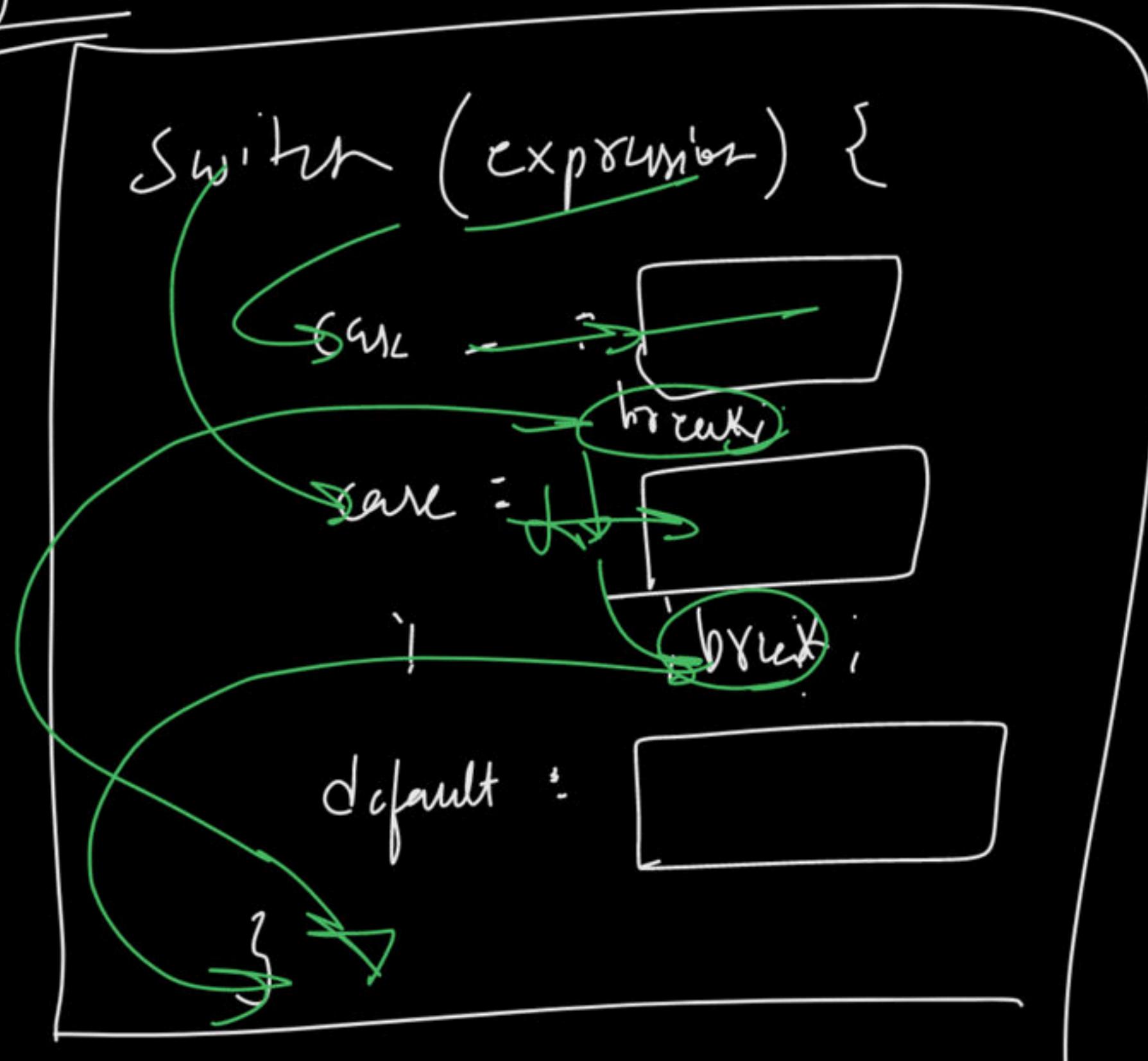
Annotations:

- A purple double-headed arrow points from the word "multiple" to the brace grouping the `else if` block.
- A yellow double-headed arrow points from the word "single" to the brace grouping the `if` block.
- A yellow double-headed arrow points from the word "else" to the brace grouping the `else` block.



Switch case

Syntax:-



Loops:

Rep' of work

15 min
or

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

~~for loop :-~~

~~for loop :-~~

~~Syntax :-~~

~~i =~~

~~0 < 5 → True~~

~~1 < 5 → True~~

~~2 < 5 → True~~

~~3 < 5 → True~~

~~4 < 5 → True~~

~~5 < 5 → False~~

initialisation

Condition

Inc/Dec/Update

for (~~int~~ i = 0 ; ~~i < 5~~ ; ~~i = i + 1~~)

Console-Log (i);

false
break

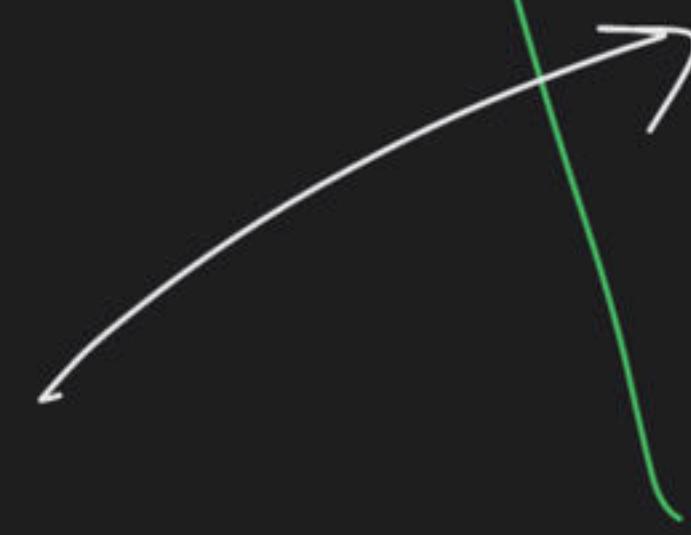
O/P → 0 1 2 3 4

for (~~int i = 0;~~ ; ~~i < 10~~ ; ~~i++~~)
init cond updation

{

compl. $\log(i);$

}



0 → <10

(0 1 2 3 4 5 6 7 8 9)

The diagram illustrates the execution flow of a while loop. It starts with an initial state where $i = 0$. The loop condition is $(\underline{\text{und}}^k)$, which is highlighted in yellow. The flow then proceeds through three continuation states, each showing an increase in the value of i : $i = 1$, $i = 2$, and $i = 3$. The update step is indicated by a transition arrow pointing from $i = 1$ to $i = 2$, enclosed in a yellow oval.

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

to

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



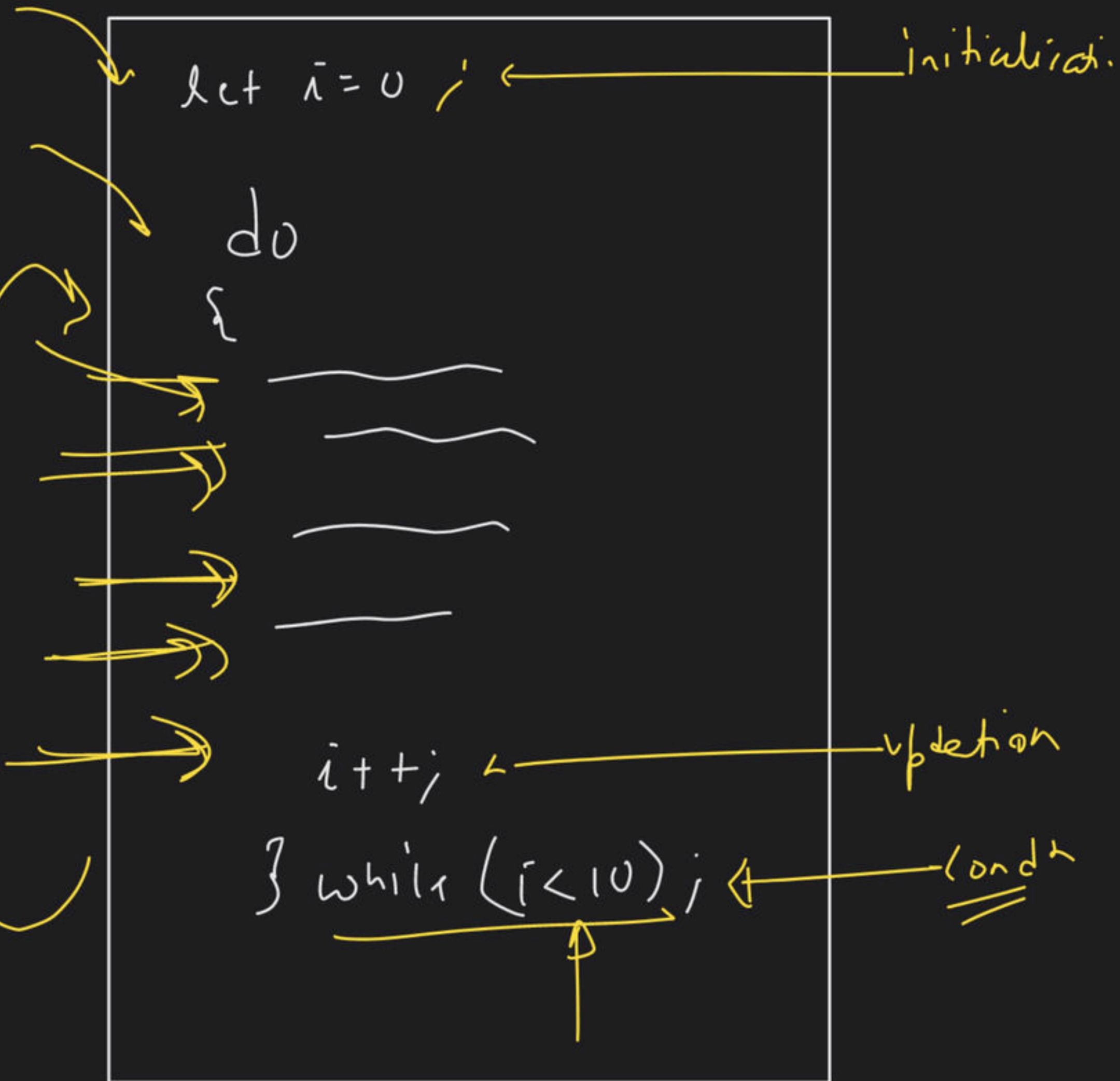
do - while loop :-



Syntax

Bch

adlcast | time
execute kernel



```
for (int i = 1; i <= 10; i++)  
{  
    cout << log(i);  
}
```

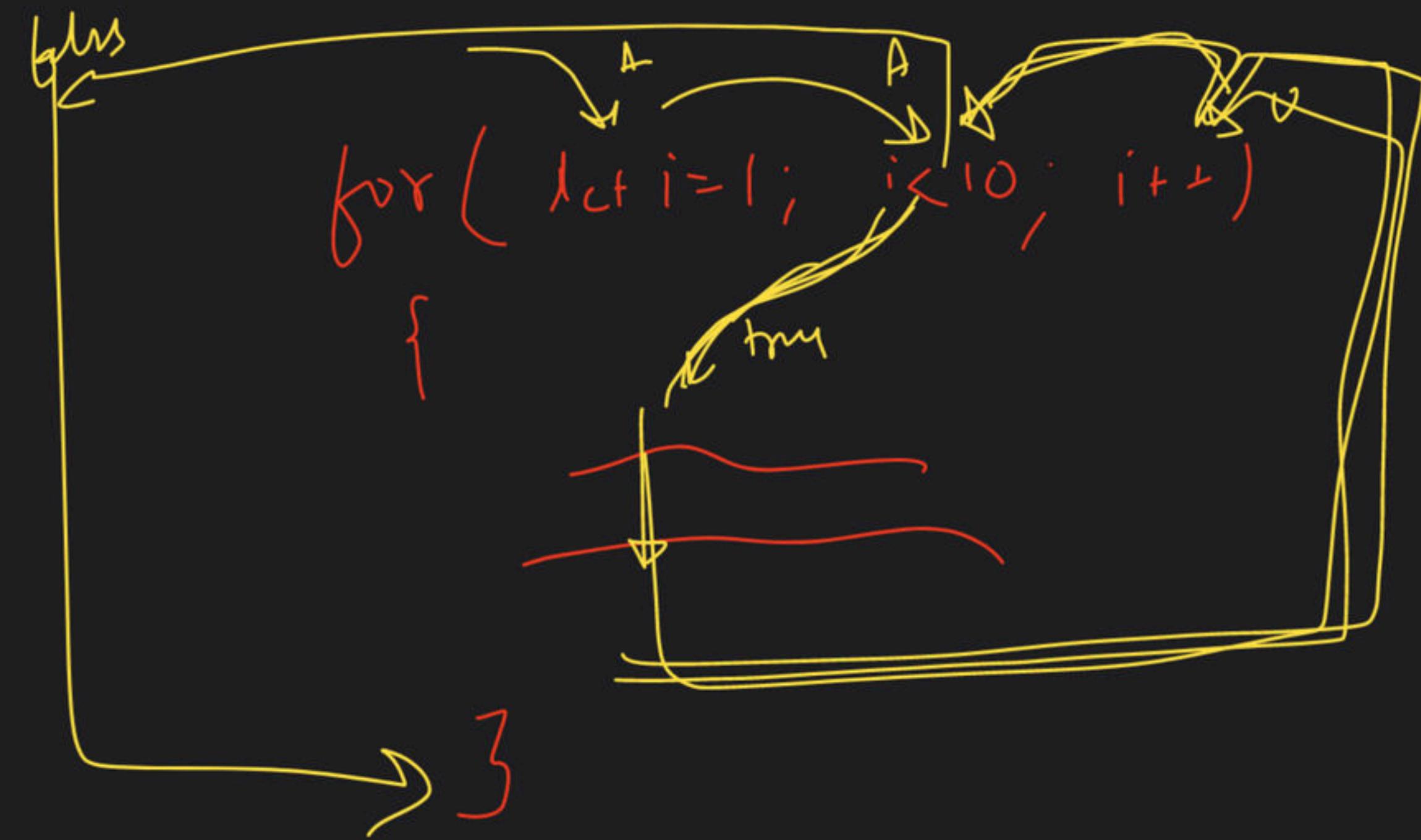
White (tree)
so much
plenty

Counting from
1 to 10

```
let x = 1;  
while (i < 10)  
{  
    console.log(i);  
    i++;  
}
```

~~100~~ 1

$i \leftarrow i + 1$
 for i
 $2 \leq 10$
 T_n
 $i <= 10$
 T
 $i <= 10$
 $1 <= 10$
 1
 T

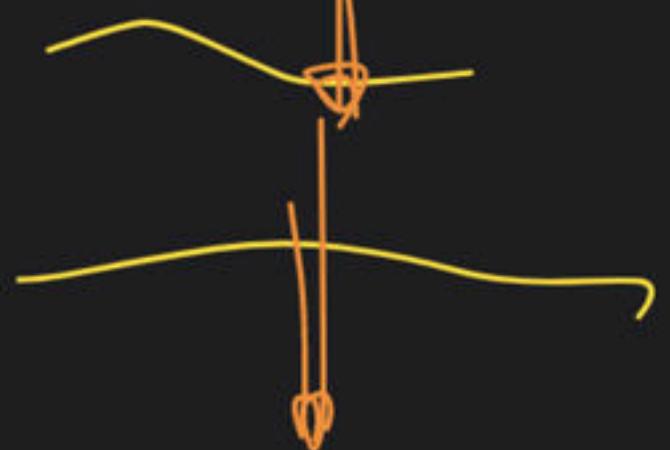


$\leftarrow \text{let } i = 0;$ *lahn*

$\text{while } (i < 10)$

{

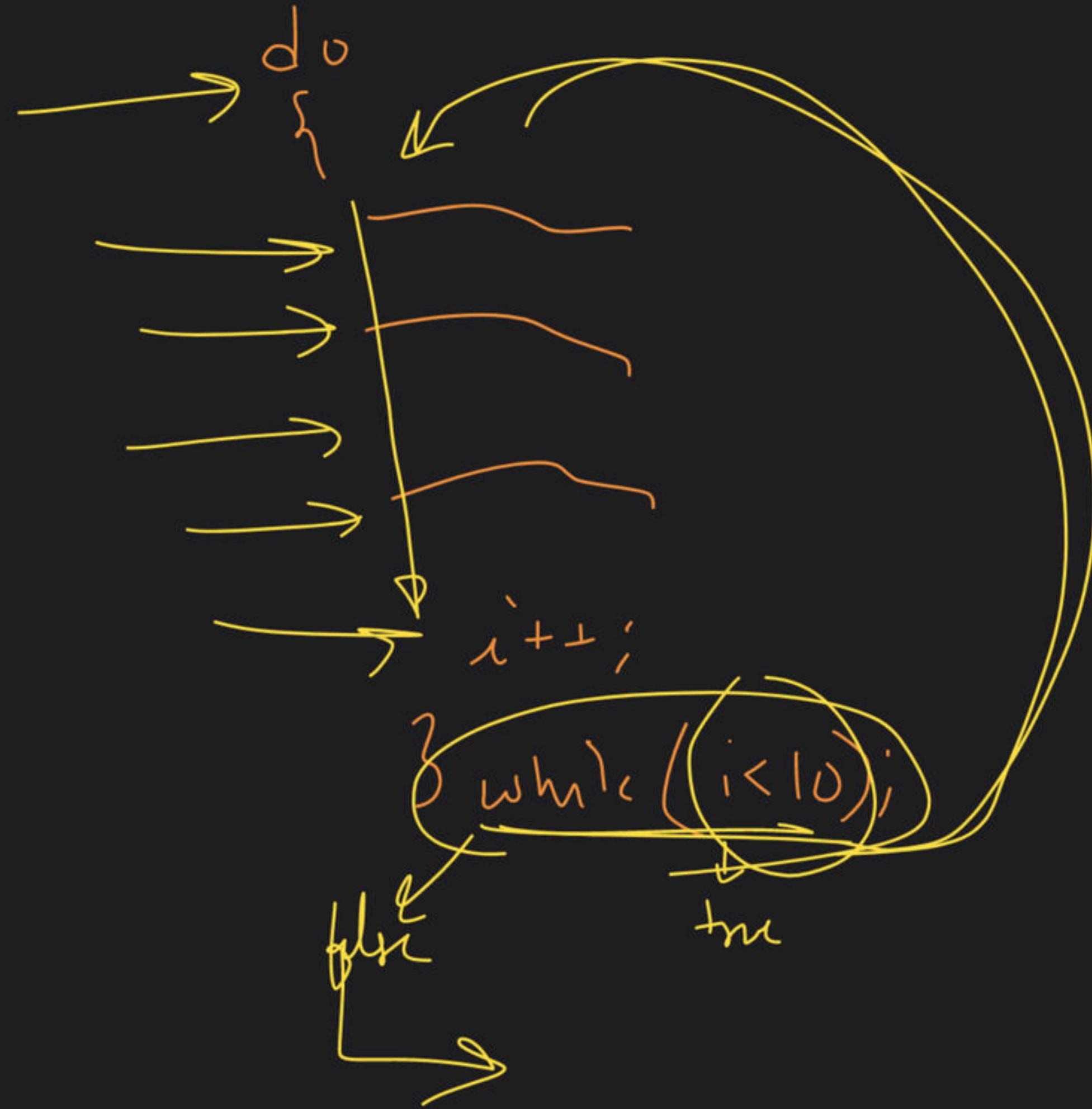
true



$i++;$

}

$\text{slit } i = 0;$



for (let i = 0; i < 5; i++)

{

 x
 y

z

3

Break & Continue: