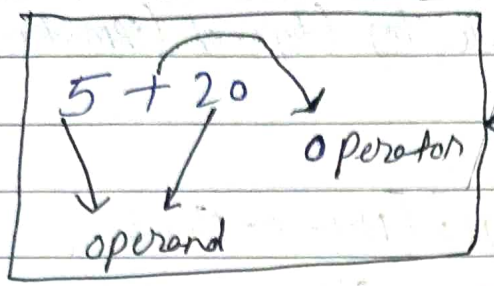
3# EXPRESSION AND OPERATOR [1:11:00]



operand + operator = expression

- Assingment Operators
- Arithmetic Operators
- Comparison Operators
- Logical Operators
- String Operators
- Conditional Operator or ternary

1. Assingment Operators :-
   - An assingment operators assign a value left operan
   - Based on the value its right operand.
   - The simple assingment is equal (=)

back string

```
Var x = 5;
Var y = 5;
Console.log (`Is both the x & y are equal: ${x==y}`);
```

⇒ True

2. Comp Arithmetic Operator (+, −, *, %, /)

- An arithmetic Operator takes numirical values. (either literal or variables) as their operands and returns a single numric value.

Addition	Console.log (3+3);	⇒ 6
Substraction	Console.log (10−5)	⇒ 5
Divison	console.log (20/5)	⇒ 4
Multiply	console.log (5*6)	⇒ 30

Modulus console.log ("Remainder Operator" + 81%.8); ⇒ 1

★★★

⇒ Increament and operator Decreament operator

operators : x++ or ++x or x-- or --x

postfix: Its means operator after operand,

The increament operator → increament & return the value before incrementing.

```
Var num = 15
Var newNum = 16; num++;
```

		Decreament
console.log (num);	⇒ 16	14
console.log (~~num++~~ newNum);	⇒ 15	15

Same

(जो दिखाई पड़ता है)

Prefix :

Postfix :- first using original value of variable & then variable is increament.

- First increament then the expression evaluated.

Programem :- Var num = 15;

Var newNum = ++num;

--num

Same

Decreament

		Decreem
Console.log (num);	⇒ 16	14
console.log (new num);	⇒ 16	14

3. Comparison- A comparison operator compar its operand and return a logical value based on whether the comparison true.

Var a = 30;

Var b = 10;

a) Equal → (==)

console.log (a == b) ⇒ ~~True~~ False ✓

b) Not Equal (≠)

console.log (a != b) ⇒ True

c) Greater Than (>)

console.log (a > b) ⇒ True

d) Less Than

console.log (a < b) ⇒ False.

e) Greater Than or equal (>=)

Console.log (a>=b) ⇒ True

f) Less than or equal (<=)

Console.log (a<=b) ⇒ False.

(a=<b) — Wrong method

Write

4. Logical Operators

- Logical operators are usually typically used boolean (Logical) values:
- return in boolean value.

Var a = 30

Var b = -20;

a) Logical AND (&&)

⇒ There are both all operand true.

b) Logical OR (||)

In this only one or more True.

c) Logical NOT (!)

It is take vice-versa

True → false

false → True

AND

console.log (a>b && b>-50 && b<0) ⇒ ~~false~~ True

true ~~false~~ True True

OR

console.log ((a>b) || (b>0) || (b>0)); ⇒ True

NOT

Console.log (!(a>0) || (b<0)); false ⇒ True

True True ⇒

False →

concatenation

5. String ^ Operator

- ~~The concatenation operator (#) two~~ Concatenates returning o
- Two string add use concatenate '+' returning as a union two operand strings.

console.log ("Hello world") → Hello World
console.log ("Hello" + "world") → HelloWorld
console.log ("Hello " + "World") → Hello World

4. ⇒ Challenging Time

a⇒ What will be output 3**3? → exponential
(3^3)

Exponentiation operator console.log (3**3) ⇒ 27

console.log (10**-1) ⇒ 1/10

b⇒ What will be output, when we add a number and strings?

console.log (5 + "thapa") ⇒ 5thapa
console.log (5 - "thapa" ⇒ NaN

***

c⇒ Write a programme to swap two numbers?

Input	output
Var a = 5;	Var a = 10
Var b = 10;	Var b = 5

__/__/____

d⇒ Write a programme to swap two number without using third variable?

```
Var a = 5;
Var b = 10;          // output = b⇒5, a⇒10

Var c = b;           // c is 10
b = a;               // b is 5
a = c;

console.log("The value of a is": +a);
console.log("The value of b is": +b);
```

→

- Interview question

★★ ⇒ Difference between == Vs ===?

(Double Equal) == this mean value is equal.
(Tripple Equal) === this mean value & data Type same