Q1: How do you declare a variable in JavaScript?

Q2: What is the difference between var, let, and const?

Q3: Can you change the value of a const variable?

Q4: What will happen if you use a variable without declaring it?

Q5: What is the default value of an uninitialized variable in JavaScript?

Q6: What are the primitive data types in JavaScript?

Q7: What is the difference between null and undefined?

Q8: Is JavaScript a statically typed or dynamically typed language?

Q9: What will be the output of typeof null?

Q10: What happens when you add a number and a string in JavaScript?

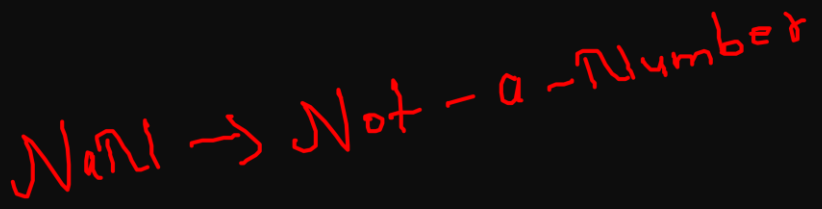
Q11: What is type coercion? Give an example.

Q12: How can you manually convert a string to a number in JavaScript?

Q13: What is the result of "5" - 3 in JavaScript?

Q14: What is NaN in JavaScript, and how do you check if a value is NaN?

Q15: How do you check the type of a variable in JavaScript?



Operators in JavaScript:

An operator is a symbol that tells the computer to do something with values.

```
let result = 5 + 3;
```

‘+’ is the operator, it tells JavaScript to add 5 & 3.

5 and 3 are the operands (value that an operator works on)

👉 Operands are the values.
👉 Operators do something to those values.

TYPES OF OPERATORS:

S.No.	**Operators**	Operator Name	**Example**
1	+, -, *, /, %, **	Arithmetic	5 + 10
2	=, +=, -=, *=, /=, %=, **=	Assignment	a += 15
3	==, ===, !=, !==, >, <, >=, <=	Comparison	5 < 10
4	&&, ||, !	Logical	true && false
5	++, --	Increment & Decrement	a++, --b
6	Condition ? True : False	Ternary	5 < 10 ? 5 : 10
7	&, |, ~, ^, <<, >>	Bitwise	a << 2

1. Arithmetic Operator (Math stuff)

These are like what you use in school math:

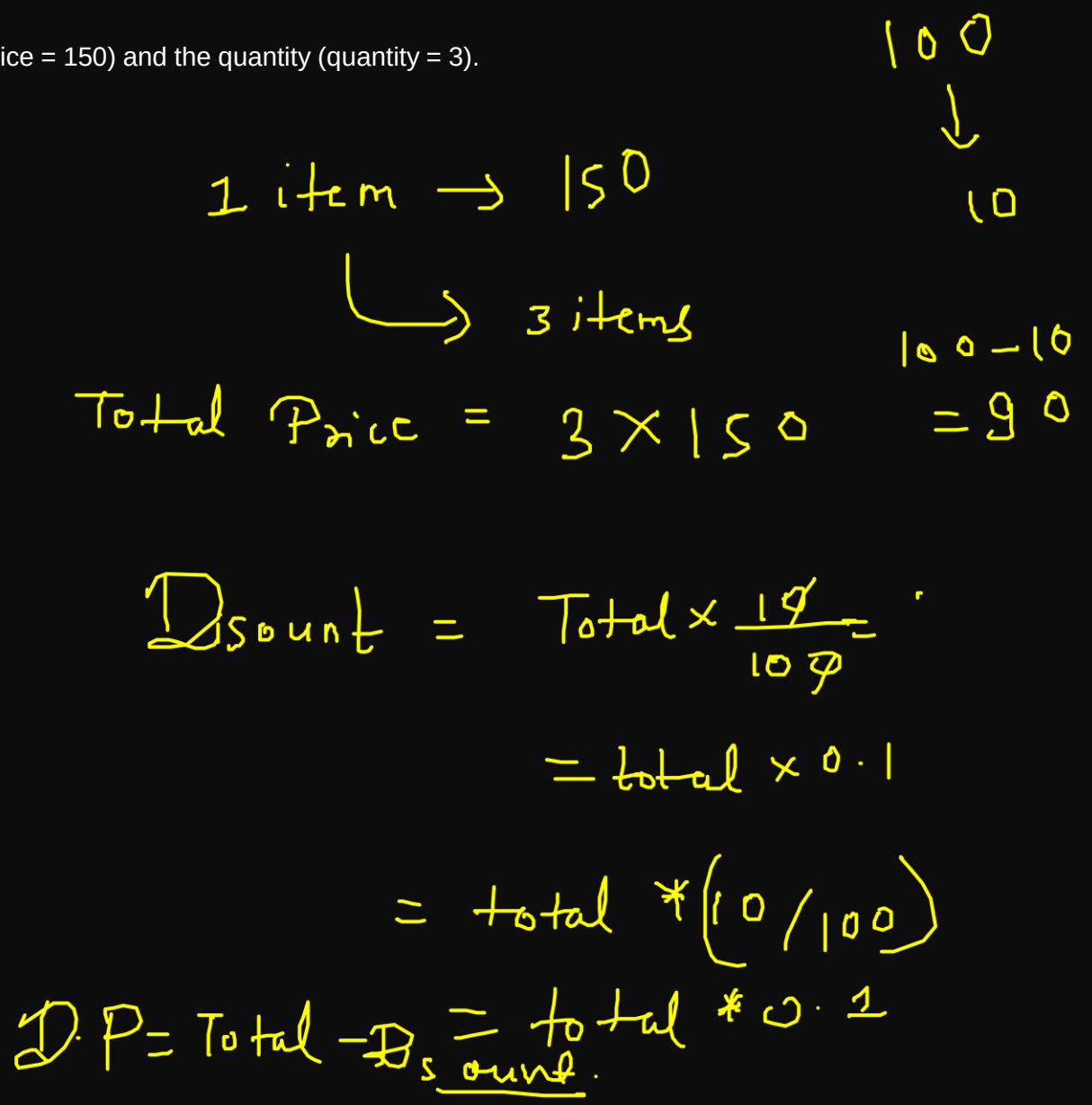
Operators	What it does	**Example**	Result
+	Add	5 + 2	7
-	Subtract	5 – 2	3
*	Multiply	5 * 2	10
/	Divide	10 / 2	5
%	Modulo (Remainder)	5 % 2	1
**	Exponent (Power)	2 ** 3	8

Challenge1:

On a shopping website, calculate the total cost of a product when given the price per item (price = 150) and the quantity (quantity = 3).

Also, calculate a 10% discount on the total cost and display the discounted price.

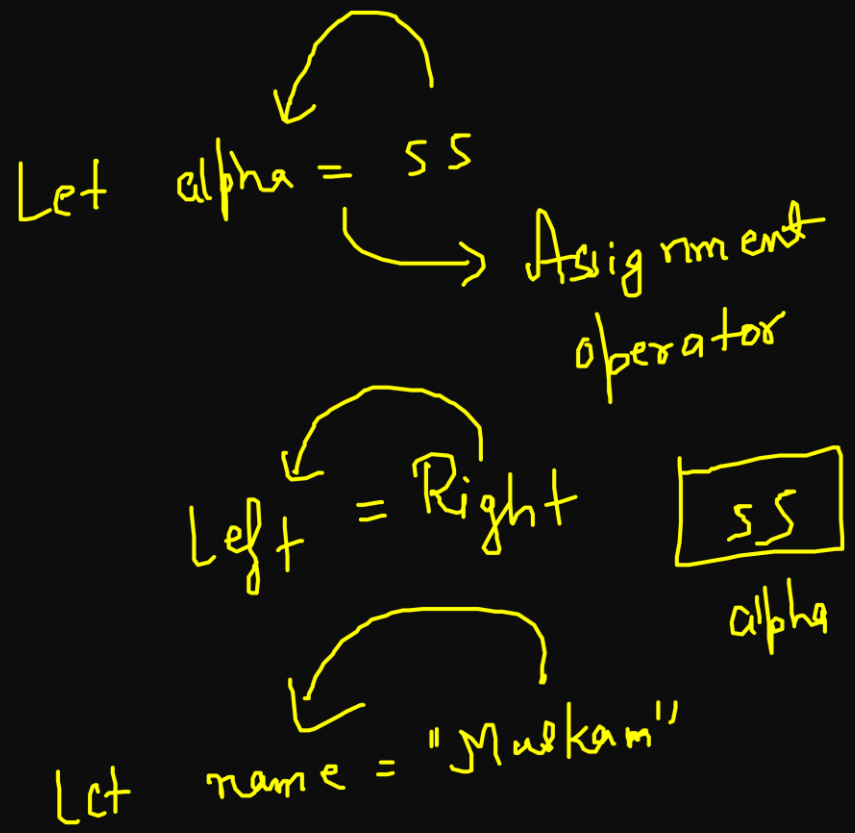
```
// Inputs: Price per item and quantity
let price = 150; // Price of a single item
let quantity = 3; // Number of items purchased

// Calculate the total cost
let totalCost = price * quantity;

// Calculate a 10% discount
let discount = totalCost * 0.10; // 10% of the total cost

// Calculate the final price after discount
let discountedPrice = totalCost - discount;

// Display the results
console.log("Price per item:", price);
console.log("Quantity:", quantity);
console.log("Total Cost:", totalCost);
console.log("Discount (10%):", discount);
console.log("Discounted Price:", discountedPrice);
```



2. Assignment Operators (Give values to variables)

These assign values to variables. You'll use = a lot.

Operators	What it does	Example	Meaning
=	Assign	x = 5	x gets 5
+=	Add and assign	x += 2	x = x + 2
-=	Subtract and assign	x -= 2	x = x – 2
*=	Multiply and assign	x *= 2	x = x * 2
/=	Divide and assign	x /= 2	x = x / 2
%=	Modulo and assign	x %= 2	x = x % 2
**=	Exponent (Power) and assign	x **= 2	x = x ** 2



Let a = 5;

a = a + 5 = 10

Let b = 10

b = b - 5

5

a = a + 5

a += 5

beta *= 8

beta = beta * 8

Let beta = 10

beta = beta * 8

80

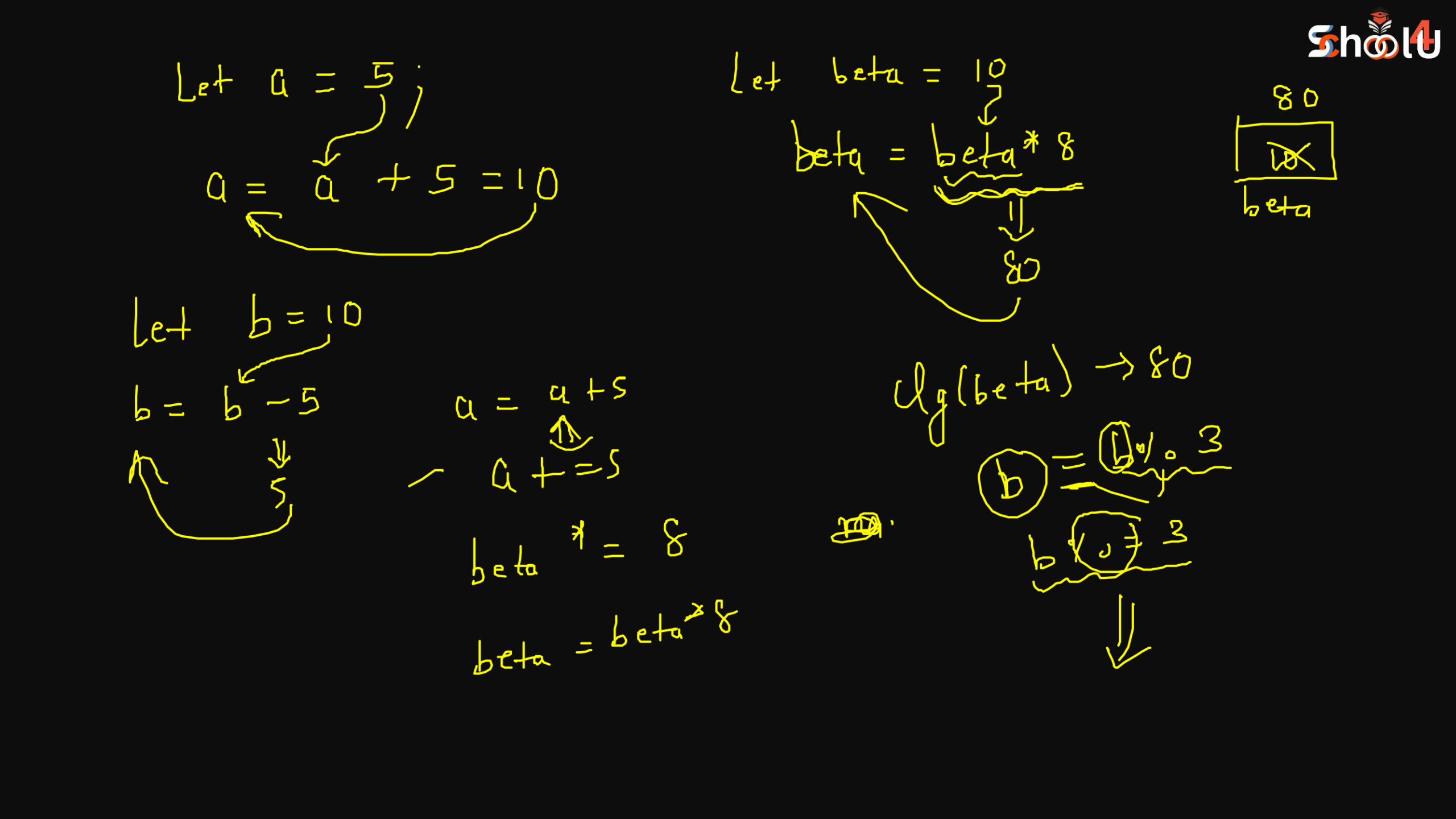
80

beta

clg(beta) → 80

b = b - 3

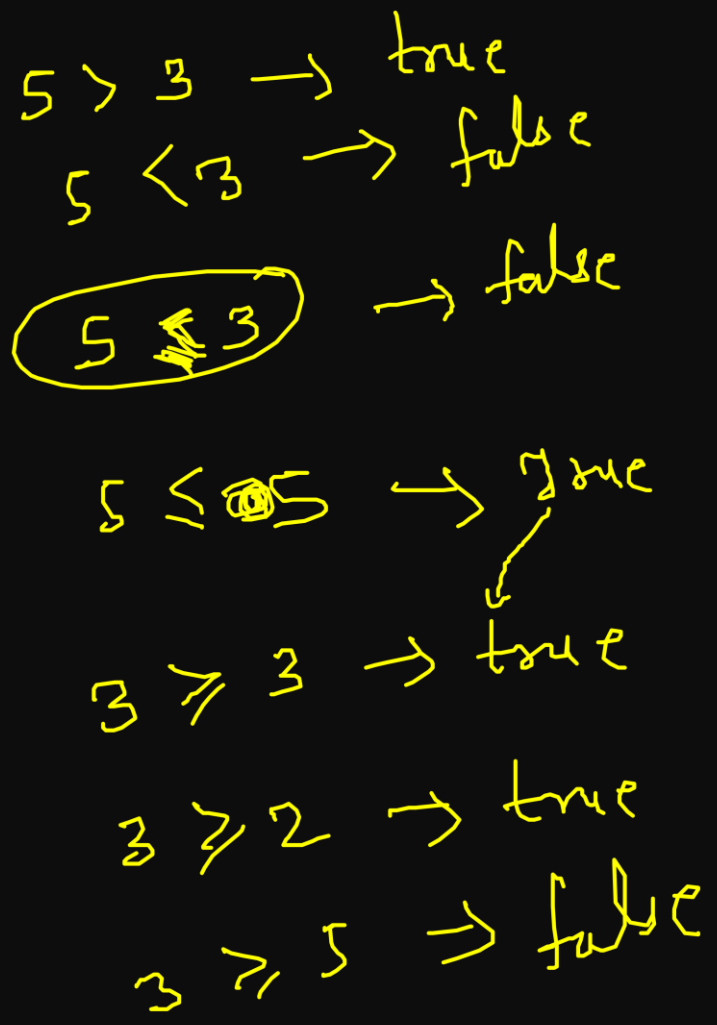
b -= 3



3. Comparison Operators (Check things)

These help you compare values. They give true or false.

Operators	What it does	Example	Result
==	Loosely equal	5 == “5”	true
===	Strictly equal	5 === “5”	false
!=	Loosely not equal	5 != “5”	false
!==	Strictly not equal	5 !== “5”	true
>	Greater than	5 > 2	true
<	Less than	5 < 2	false
>=	Greater than or equal to	5 >= 5	true
<=	Less than or equal to	5 <= 3	false



5 ✕ 3

== → value

=== → DataType, value

5 == 3 → ~~false~~ false

5 == 5 ⇒ true

5 == "5" ⇒ true

5 === "5" ⇒ false

5 → no.

"5" → string

5 === 5 ⇒ true

"5" === "5" ⇒ true

"5" → Datatype → value

== → Type Coercion → strictly equal

5 == "5"

"5" → "5"

true → true → true

5 + "3"

5 → string → "5"

"3" → 3

"5" + 3 → "53"

5 != "3" ⇒ true

5 !== "3" ⇒ true

5 != 3 ⇒ true

5 - "3"

"3" → number

5 - 3 = 2

4. Logical Operators (Combine conditions)

Used when you're checking more than one thing.

Operators	What it does	**Example**	Result
&&	AND (both must be true)	true && true	true
||	OR (either one is true)	true || false	true
!	NOT (flips true/false)	!true	false

&& (AND)

true false → false

a	b	?
true	false	false
false	true	false
false	false	false
true	true	true

|| (OR)

a	b	
t	f	t
f	t	t
t	t	t
f	f	f

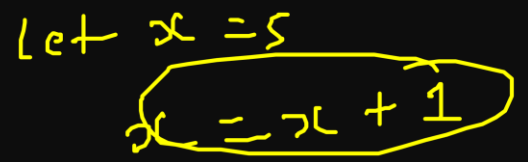
Q. x > 5 && x < 1000

! (NOT)

!f → t

!t → f

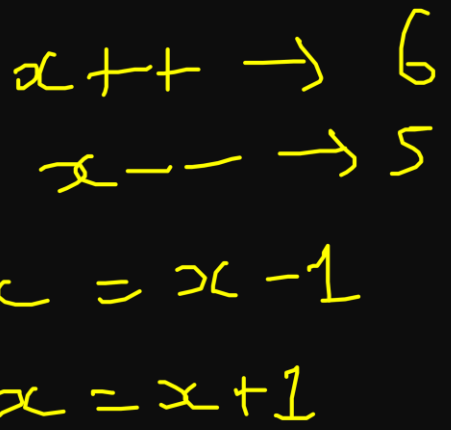
!(x > 5)



5. Increment/Decrement Operators (Add or subtract 1)

Quick way to increase or decrease a value.

Operators	What it does	**Example**	Result
+ +	Add 1	x++	x = x + 1
- -	Subtract 1	x- -	x = x - 1



Increment

Post++ ++Pre

Decrement

Post Pre

Let a = 10

Let result = a++ + ++a -10;

10 + 12 -10

22 -10 = 12

6: Ternary Operator (Shortcut for if/else)

```
condition ? execute if condition is true : execute if condition is false;
```

Example:

```
let result = age >= 18 ? "Adult" : "Minor";
```

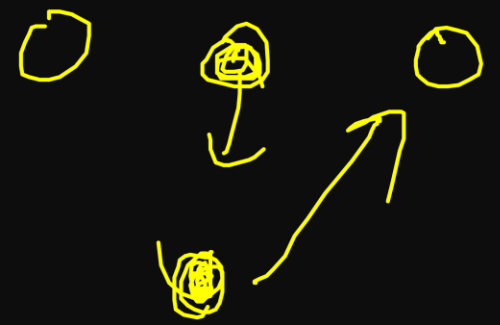
It reads like:

"If age is 18 or more, result is 'Adult', otherwise it's 'Minor'"

find the largest no. from given 3 numbers.

let a = 5 , b = 2 , c = 12

let result = a > b ? (a > c ? a : c) : (b > c ? b : c)



Challenge2:

On a booking website, check if the user's age is valid for booking:
Age should be at least 18.
Write a condition to check and display a message: "Eligible for booking" if the user is 18 or older. "Not eligible for booking" otherwise.

```javascript
// Input: User's age
let age = 18; // Example input, can be changed dynamically

// Use a ternary operator to determine the message
let message = age >= 18 ? "Eligible for booking" : "Not eligible for booking";

// Display the message
console.log(message);
```

Challenge 3:

On a login page, verify the user's credentials: Check if username is not empty AND password is not empty (&& operator).
If either is empty, display an error message: "Both fields are required."

```javascript
// Input: Username and password
let username = ""; // Example: Empty username
let password = ""; // Example: Empty password

// Check credentials using the ternary operator
let message = (username && password)
    ? "Login successful"
    : "Both fields are required.";

// Display the message
console.log(message);
```

7. Bitwise Operators:

super useful in certain situations — especially when you're working at a low level (like with individual bits of data)

Operators	What it does	Example	Meaning
&	Bitwise AND	a & b	a AND b
|	Bitwise OR	a | b	a OR b
~	Bitwise NOT	~a	NOT of a
^	XOR	a ^ b	a XOR b
<<	Left Shift	a << 2	a Left Shift by 2
>>	Right Shift	a >> 1	a Right Shift by 2

They're more common in:

- Algorithms
- Game engines
- Performance-critical code
- System-level programming

~(-5 → ---- 1 1 1 0 1 1

not

---00000100

↓

4

↓

x = 3

-(3 + 1)

-(4) = -4

-(-5 + 1)

= -(-4)

= 4

x = 0

-(0 + 1) = -1

Let’s categorize operator on the basis of number of operands:

Operand Count	Type	Operators
1 (Unary)	Arithmetic	+, -, ++, --
	Logical	!
	Type-checking	typeof, void, delete
Binary	Arithmetic	+, -, *, /, %, **
	Comparison	==, ===, !=, !==, >, <, >=, <=
	Logical	&&, ||
	Assignment	=, +=, -=, /=, %=, **=
	Bitwise	&, |, ^, <<, >>
3 (Ternary)	Conditional	Condition ? true : false

Working with non-Booleans values (Truthy v/s Falsy)

✅Falsy → false, undefined, null, 0, -0, 0n, “”, NaN.

❌Truthy → anything which is not falsy.

Short - Circuiting

Short-circuiting in JavaScript refers to the way logical operators (&&, ||, and ??) evaluate expressions. It allows you to control the flow and return values based on truthiness without writing full if statements.

1. || (Logical OR) : Returns the first truthy value or the last value if none are truthy. (Useful for setting default values.)

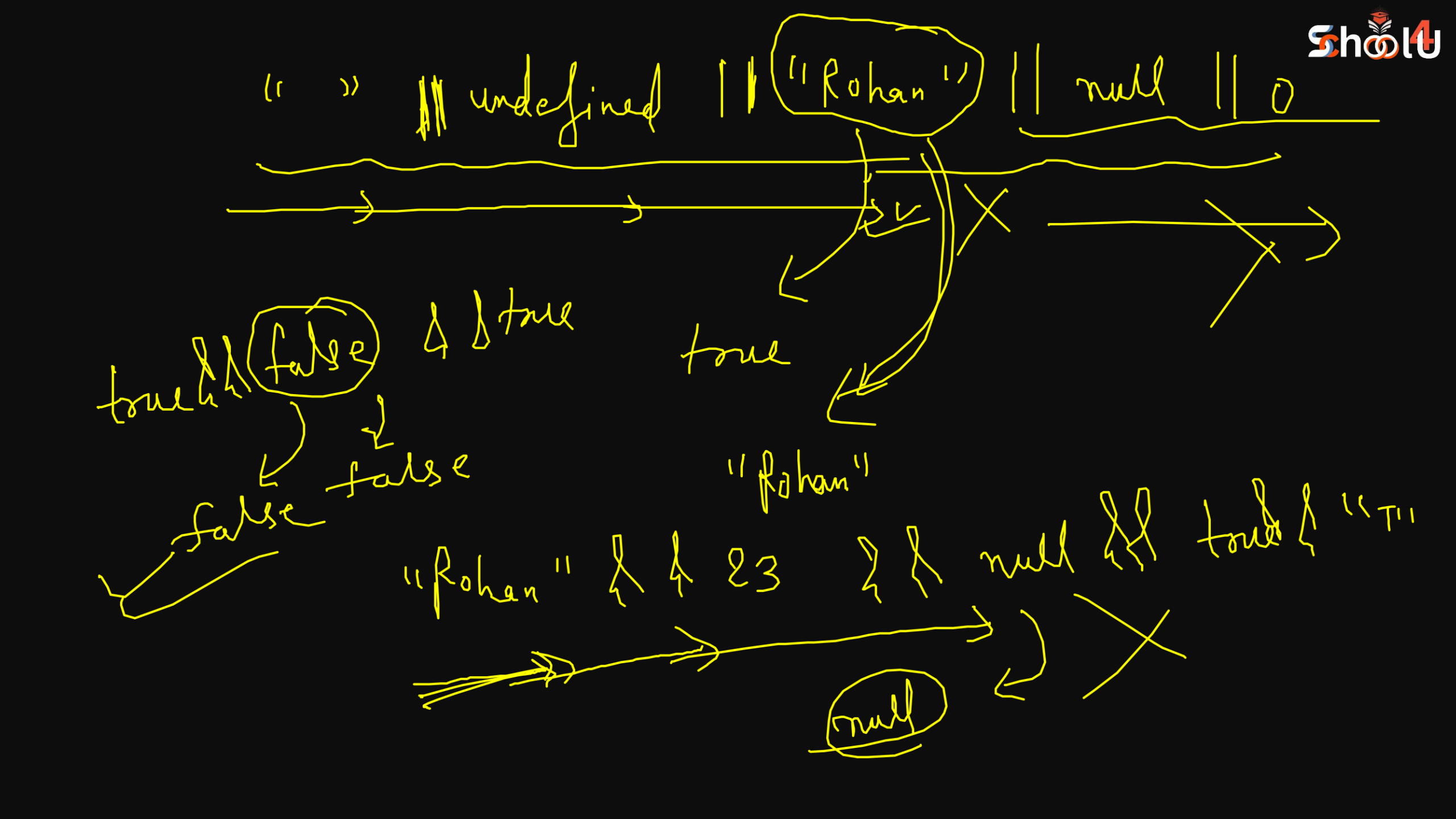
```
let result = "" || "Guest" || null || 23;
console.log(result); // "Guest"

let result2 = undefined || 0 || null;
console.log(result2); // null
```

2. && (Logical AND): Returns the first falsy value or the last value if none are falsy. (Commonly used to safely access properties.)

```
let result = "Ram" && true && undefined && 55;
console.log(result); // undefined

let isAuthenticated = true;
let user = "Manas Kumar Lal";
let result2 = isAuthenticated && user;
console.log(result2); // "Manas Kumar Lal"
```

3. ?? (Nullish Coalescing): Returns the right-hand value only if the left is null or undefined.
(Better than || when dealing with falsy values like 0 or "" that are still valid.)

```
let result = null ?? "Default";
console.log(result); // "Default"

let result2 = 0 ?? "MKL";
console.log(result2) // 0
```


"" || undefined || "Rohan" || null || 0
true && true
true
false
true
false
false
"Rohan"
"Rohan" && 23 && null && true && "T"
null

1. Get user to input two number using prompt and print their possible arithmetic results.

2. Can you chain assignment operators?

3. Get user to input a number using prompt and check whether even or odd using ternary operator.

4. What is the final value of x?

```
let x = 5;
x += 3;
x -= 2;
x *= 4;
x /= 6;
x %= 3;
```

5. Check if a number is within a range between 10 and 20 (inclusive).

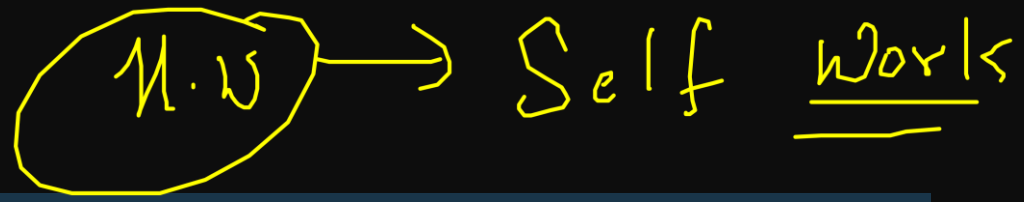
6. Write a program to find the largest number between 3 numbers using ternary operator.

7. Take an email and password from the user. If the email or password is incorrect or does not match the stored values, display the message "Invalid email or password." If both email and password match the stored values, display "You are logged in successfully!"

8. What will be the output of the following JavaScript code?

```
let a = 5, b = 3, c = 2;

let result = a++ + --b * c-- - ++a + b-- / --c;

console.log("a:", a);
console.log("b:", b);
console.log("c:", c);
console.log("result:", result);
```

9. What is the output of console.log(~a) where a = 0



9. What will be the output of the following JavaScript code?

```
let x = 10;
let y = 5;
let z = "10";

x += y * 2; // Arithmetic + Assignment
let isEqual = x == z; // Loose comparison
let isStrictEqual = x === z; // Strict comparison
let logicTest = (isEqual || isStrictEqual) && !(y > 10); // Logical + Comparison + NOT

let result = logicTest ? ++x : --y; // Ternary + Pre-increment/Pre-decrement

console.log("x:", x);
console.log("y:", y);
console.log("z:", z);
console.log("isEqual:", isEqual);
console.log("isStrictEqual:", isStrictEqual);
console.log("logicTest:", logicTest);
console.log("result:", result);
console.log("Type of z:", typeof z); // Unary operator typeof
```

10. What will be the output of the following JavaScript code?

```
let a = 6;
let b = 3;
let c = "6";

a += b << 1; // Bitwise left shift + assignment
let d = a & b; // Bitwise AND
let e = a | b; // Bitwise OR
let f = a ^ b; // Bitwise XOR
let g = ~a; // Bitwise NOT

let check = (a == c) && (d < e) || !(f === e); // Mixed logical, comparison

let result = check ? typeof g : --b; // Ternary + unary + typeof

console.log("a:", a);
console.log("b:", b);
console.log("c:", c);
console.log("d (a & b):", d);
console.log("e (a | b):", e);
console.log("f (a ^ b):", f);
console.log("g (~a):", g);
console.log("check:", check);
console.log("result:", result);
```

$\Rightarrow$ $a \mathbin{+=} b << 1$

$a += 6$

$\Rightarrow 12$

$\rightarrow$ Arithmetic

$\rightarrow$ $<<$, $>>$

$\rightarrow$ Assignment

$\Rightarrow a = a + b << 1$

$\Rightarrow a = 15$