OPERATORS

ARITHEMATIC OPERATORS

- + (Addition)
- - (Subtraction)
- * (Multiplication)
- / (Division)
- % (Modulus - returns the remainder of a division)
- ++ Increment Operator
- -- Decrement Operator

```
Addition: 15
Subtractoin: 5
Multiplication: 50
Division: 2
Modulus: 0
Increment: 10
Decrement: 5
11
4
```

```
let a = 10;
let b = 5;
console.log("Addition: "+(a + b)); // 15
console.log("Subtractoin: "+a - b); // 5
console.log("Multiplication: "+a * b); // 50
console.log("Division: "+a / b); // 2
console.log("Modulus: "+a % b); // 0
console.log("Increment: "+(a++)); // 2
console.log("Decrement: "+(b--)); // 0
console.log(a)
console.log(b)
```

ASSIGNMENT OPERATORS

Name	Description	Syntax
Assignment (=)	This operator assigns the right operand value to the left operand.	If A = 10 and Y = A then Y = 10
Addition Assignment (+=)	Sums up left and right operand values and then assigns the result to the left operand.	Y += 1 gives Y = Y + 1
Subtraction Assignment (- =)	It subtracts the right side value from the left side value and then assigns the result to the left operand.	Y -= 1 gives Y = Y – 1
Multiplication Assignment (*=)	It multiplies a variable by the value of the right operand and assigns the result to the variable.	Y *= A is equivalent to Y = Y * A
Division Assignment (/ =)	It divides a variable by the value of the right operand and assigns the result to the variable.	Y /= A is equivalent to Y = Y / A
Modules/Remainder Assignment (% =)	It divides a variable by the value of the right operand and assigns the remainder to the variable.	Y %= A is equivalent to Y = Y % A
Exponentiation Assignment (** =)	This raises the value of a variable to the power of the right operand.	Y **= A is equivalent to Y=Y ** A

COMPARISION OPERATORS

Name	Description	Syntax
Equality (==)	Compares the equality of two operands. If equal then the condition is true otherwise false.	Y = 5 and X = 6 Y = = X is false.
Strict equality (===)	Compares the equality of two operands with type. If both value and type are equal then the condition is true otherwise false.	Y = 5 and X = ‘5’ Y = = = X is false.
Inequality (!=)	Compares inequality of two operands. True if operands are not equal.	let X = 10 then X ! = 11 is true.
Strict inequality(!==)	Compares the inequality of two operands with type. If both value and type are equal then the condition is true otherwise false.	let X = 10 then X ! == ’10’ is true.
Greater than (>)	This operator checks whether the left side value is greater than the right side value. If yes then it returns true otherwise it returns false.	let X = 10 then X > 11 is false.
Less than (<)	This operator checks whether the left side value is less than the right side value. If yes then it returns true otherwise it returns false.	let X = 10 then X < 11 is true.
Greater than or Equal to (> =)	This operator checks whether the left side operand is greater than or equal to the right side operand. If yes then it returns true otherwise it returns false.	let X = 10 then X > = 11 is false.
Less than or Equal to (<=)	This operator checks whether the left side operand value is less than or equal to the right side operand value. If yes then it returns true otherwise it returns false.	let X = 10 then X < = 10 is true.