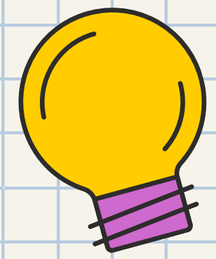
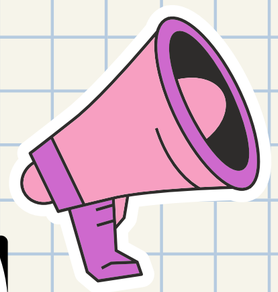
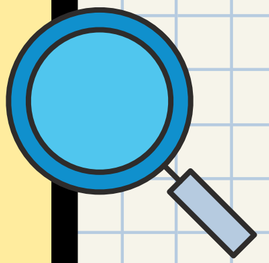
Javascript Operators







Types of Operators

In JavaScript, an operator is a special symbol used to perform operations on operands (values and variables).

- **Assignment Operators**
- **Arithmetic Operators**
- **Comparison Operators**
- **Logical Operators**
- **Bitwise Operators**
- **String Operators**

Arithmetic Operators

Arithmetic operators are used to perform arithmetic calculations.

Operator	Name	Example
`+`	Addition	`x + y`
`-`	Subtraction	`x - y`
`*`	Multiplication	`x * y`
`/`	Division	`x / y`
`%`	Remainder	`x % y`
`++`	Increment (increments by 1)	`++x` or `x++`
`--`	Decrement (decrements by 1)	`--x` or `x--`
`**`	Exponentiation (Power)	`x ** y`

Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Name	Example
`=`	Assignment operator	`a = 7; // 7`
`+=`	Addition assignment	`a += 5; // a = a + 5`
`-=`	Subtraction Assignment	`a -= 2; // a = a - 2`
`*=`	Multiplication Assignment	`a *= 3; // a = a * 3`
`/=`	Division Assignment	`a /= 2; // a = a / 2`
`%=`	Remainder Assignment	`a %= 2; // a = a % 2`
`**=`	Exponentiation Assignment	`a **= 2; // a = a**2`

Comparison Operators

Comparison operators compare two values and return a boolean value, either true or false

Operator	Description	Example
`==`	**Equal to:** returns `true` if the operands are equal	`x == y`
`!=`	**Not equal to:** returns `true` if the operands are not equal	`x != y`
`===`	**Strict equal to:** `true` if the operands are equal and of the same type	`x === y`
`!==`	**Strict not equal to:** `true` if the operands are equal but of different type or not equal at all	`x !== y`
`>`	**Greater than:** `true` if left operand is greater than the right operand	`x > y`
`>=`	**Greater than or equal to:** `true` if left operand is greater than or equal to the right operand	`x >= y`
`<`	**Less than:** `true` if the left operand is less than the right operand	`x < y`
`<=`	**Less than or equal to:** `true` if the left operand is less than or equal to the right operand	`x <= y`

Logical Operators

Logical operators perform logical operations and return a boolean value, either true or false.

Operator	Description	Example
`&&`	**Logical AND:** `true` if both the operands are `true`, else returns `false`	`x && y`
`||`	**Logical OR:** `true` if either of the operands is `true`; returns false if both are `false`	`x || y`
`!`	**Logical NOT:** `true` if the operand is `false` and vice-versa.	`!x`