+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

✅ Categories of JavaScript Operators:

1. Arithmetic Operators
2. Assignment Operators
3. Comparison Operators
4. Logical Operators
5. Bitwise Operators
6. String Operators
7. Conditional (Ternary) Operator
8. Type Operators
9. Spread and Rest Operators
10. Unary Operators
11. Comma Operator
12. Nullish Coalescing Operator
13. Optional Chaining Operator

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

✅ Frequently Used in Most Programming Languages

These are foundational and widely used:

1. **Arithmetic Operators**

➢ +, -, *, /, % — Used in almost every program that involves numbers.

2. **Assignment Operators**

➢ =, +=, -=, etc. — Essential for assigning and updating values.

3. **Comparison Operators**

➢ ==, !=, >, <, >=, <= — Used for conditions and control flow.

4. **Logical Operators**

➢ &&, ||, ! — Used for combining conditions in if/else, loops, etc.

5. **Conditional (Ternary) Operator**

➢ condition ? expr1 : expr2 — Very useful for compact conditional logic. 🧠 **Useful but Context-Specific**

Used often in certain areas or languages:

6. **Bitwise Operators**

➢ &, |, ^, ~, <<, >> — Used in low-level programming, embedded systems, flags, and performance optimization.

7. **String Operators**

➢ In JavaScript: + (for concatenation); in Python: +, * — Common in string manipulation

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

tasks.

8. **Unary Operators**

➢ ++, --, typeof, delete (in JS), not, -, + — Used in many languages, but varies.

9. **Type Operators**

➢ typeof, instanceof (JS); type() or isinstance() (Python) — Useful for type checking.

⚙️ Language-Specific or Advanced Use

These are powerful but less common in basic programs:

10. **Spread and Rest Operators**

➢ ... in JavaScript — Used in arrays, function arguments, and objects. Very handy in modern JS.

11. **Comma Operator**

➢ , — Used to evaluate multiple expressions (rare, mainly in for-loops in C/C++ and JS).

12. **Nullish Coalescing Operator**

➢ ?? in JavaScript — Useful for providing defaults only when value is null or undefined.

13. **Optional Chaining Operator**

➢ ?. in JavaScript — Very useful for accessing nested properties safely.

frameboxx 2.0®
animation | visual effects | programming | coding
Premier Academy for IT Development

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

Summary Table

Operator Category	Commonly Used?	Language Specific?	Notes
Arithmetic	✅ Yes	No	Basic math/logical work
Assignment	✅ Yes	No	Essential for all logic
Comparison	✅ Yes	No	Used in conditions
Logical	✅ Yes	No	Combine conditions
Bitwise	🧠 Rarely	No	Low-level programming
String	✅ Yes	No	String manipulation
Conditional (Ternary)	✅ Yes	No	Compact logic
Type	✅ Yes	Yes (syntax varies)	Type safety and introspection
Spread and Rest	✅ Yes	Yes (mainly JS)	Modern JS/TS usage
Unary	✅ Yes	No	Many roles
Comma	🧠 Rarely	Yes (C/JS)	Used mainly in for-loops
Nullish Coalescing (`??`)	✅ Yes	Yes (JS, TS)	Safe defaulting
Optional Chaining (`?.`)	✅ Yes	Yes (JS, TS)	Safe property access

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

1. Arithmetic Operators

📚 **What they do**: Add, subtract, multiply, or divide numbers.

```
let apples = 5 + 3; // 8 apples

let bananas = 10 - 2; // 8 bananas

let candies = 4 * 2; // 8 candies

let friends = 16 / 2; // 8 friends
```

2. Assignment Operators

📚 **What they do**: Put values into variables.

```
let score = 10; // assigns 10

score += 5; // now score is 15

score -= 2; // now score is 13
```

3. Comparison Operators

📚 **What they do**: Compare things (like who's taller or who has more).

```
let age = 10;

console.log(age > 8); // true

console.log(age == 10); // true

console.log(age < 5); // false
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

4. Logical Operators

📚 **What they do**: Combine questions (like “Is it sunny AND hot?”)

```
let isSunny = true;

let isWarm = true;console.log(isSunny && isWarm); // true (both are true)

console.log(isSunny || false); // true (one is true)

console.log(!isSunny); // false (opposite of true)
```

5. Bitwise Operators

📚 **What they do**: Work with bits (used in games and hardware).

```
let a = 5; // 101 in binary

let b = 3; // 011 in binary

console.log(a & b); // 1 (001)

console.log(a | b); // 7 (111)
```

6. String Operators

📚 **What they do**: Connect words or letters.

```
let firstName = "Tom";

let lastName = "Cat";

let fullName = firstName + " " + lastName; // "Tom Cat"
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

7. Conditional (Ternary) Operator

📚 **What it does**: Makes short decisions.

```
let age = 12;

let canRide = age >= 10 ? "Yes!" : "No!";

console.log(canRide); // "Yes!"
```

8. Type Operators

📚 **What they do**: Check what kind of thing something is.console.log(typeof 123); // "number"

```
console.log(typeof "hello"); // "string"
```

9. Spread and Rest Operators

📚 **What they do**: Combine or separate things.

```
let toys = ["car", "doll"];

let moreToys = ["ball", ...toys]; // ["ball", "car", "doll"]

function countToys(...allToys) {

console.log(allToys.length);

}

countToys("car", "doll", "ball"); // 3
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

10. Unary Operators

📚 **What they do**: Work on just one thing.

```
let x = 5;

x++; // now x is 6

x--; // back to 5
```

11. Comma Operator

📚 **What it does**: Runs many things and gives back the last one.

```
let result = (1 + 2, 3 + 4);

console.log(result); // 7 (not 3)
```

12. Nullish Coalescing Operator (??)

📚 **What it does**: Gives a default only if value is null or undefined.let name = null;

```
let displayName = name ?? "Guest";

console.log(displayName); // "Guest"
```

13. Optional Chaining Operator (?.)

📚 **What it does**: Safely checks if something exists before accessing it.

```
let pet = { name: "Fluffy",

type: "cat" };

console.log(pet?.name); // "Fluffy"

console.log(pet?.age); // undefined (no error)
```

frameboxx 2.0®
animation | visual effects | programming | coding
Premier Academy for IT Development

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

🧠 Tasks to Understand All JavaScript Operators

🔢 1. Arithmetic Operators Tasks

- Operators: +, -, *, /, %, ++, --

Tasks:

1. Add two numbers using + and display the result.2. Subtract two numbers using -.

3. Multiply two variables using *.

4. Divide two values using /.

5. Find the remainder using %.

6. Increment a number using ++.

7. Decrement a number using --.

8. Chain multiple arithmetic operations with brackets.

9. Combine arithmetic with user input (prompt).

10. Calculate area and perimeter of a rectangle using operators.

✍️ 2. Assignment Operators Tasks

- Operators: =, +=, -=, *=, /=, %=, **=

Tasks:

1. Assign a value to a variable with =.

2. Increase a value by 10 using +=.

3. Reduce a value using -=.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

4. Multiply and reassign using *=.5. Divide and reassign using /=.

6. Use modulus assignment %=.

7. Use exponentiation assignment **=.

8. Chain multiple assignment operators.9. Display results at each step to observe the changes.

10. Practice all assignments in a calculator app.

🧾 3. Comparison Operators Tasks

- Operators: ==, ===, !=, !==, >, <, >=, <=

Tasks:

1. Compare two numbers using == and ===.

2. Compare a string and a number using ==.

3. Use !== to check strict inequality.

4. Use > and < between variables.

5. Create a grading system using >=, <=.

6. Compare strings (e.g., "apple" > "banana").

7. Use comparisons in an if-else condition.

8. Compare dates using comparison operators.

9. Use comparison inside a loop (e.g., i < 5).

10. Create a login check (username === "admin").

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

🧠 4. Logical Operators Tasks

- Operators: &&, ||, !

Tasks:1. Use && to check if age is between 18 and 60.

2. Use || to check if user is admin or manager.

3. Use ! to invert a boolean.

4. Combine logical operators in a complex condition.

5. Use logical operators in if statements.

6. Validate multiple form fields.

7. Short-circuit evaluation with ||.

8. Short-circuit evaluation with &&.

9. Use ! with falsy values (!0, !null, etc.).

10. Create access control using logical operators.

⚙️ 5. Bitwise Operators Tasks

- Operators: &, |, ^, ~, <<, >>

Tasks:

1. Use & to perform AND on two numbers.

2. Use | for OR operation.

3. Use ^ to find differing bits.

4. Use ~ to invert a number.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

5. Use << to shift bits left.6. Use >> to shift bits right.

7. Compare bitwise vs logical operators.8. Convert binary to decimal and apply bitwise.

9. Visualize how numbers change in bits.

10. Create a binary toggle system using XOR.

🔤 6. String Operators Tasks

- Mainly: +, +=

Tasks:

1. Concatenate two strings using +.

2. Append strings using +=.

3. Combine strings and numbers.

4. Use template literals instead of +.

5. Concatenate multiple values in a single line.

6. Build a sentence from variables.

7. Show difference between number addition and string concatenation (1 + "2").

8. Create a welcome message using variables.

9. Practice using backticks and ${}.

10. Convert number to string before concatenation.

❓ 7. Conditional (Ternary) Operator Tasks

- Operator: condition ? expr1 : expr2

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

Tasks:1. Use ternary to check if a number is even or odd.2. Replace a basic if-else with ternary.

3. Use ternary to display login status.

4. Nest ternary operators for multiple conditions.

5. Use ternary in DOM manipulation (document.write).

6. Assign result of ternary to a variable.

7. Use ternary for age group: child/adult/senior.

8. Practice readability with ternary.

9. Use ternary in arrow function return.

10. Refactor a multi-line if into a ternary.

🕵️ 8. Type Operators Tasks

- Operators: typeof, instanceof

Tasks:

1. Use typeof to log the type of a string, number, boolean, null, undefined, etc.

2. Check if a variable is a function.

3. Check type before performing an operation.

4. Create a function that logs type of any input.

5. Use typeof in conditional logic.

6. Check if object is instance of Array.

7. Use instanceof to verify custom class.8. Compare typeof null and explain.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

9. Practice with built-in types.10. Test edge cases (typeof NaN, typeof undefined).

📦 9. Spread and Rest Operators Tasks

- Operators: ...

Tasks:

1. Copy an array using spread syntax.

2. Merge two arrays using spread.

3. Clone an object using spread.

4. Use rest parameters in a function.

5. Use spread to pass arguments to a function.

6. Combine spread with destructuring.

7. Write a function that sums all numbers using rest.

8. Use spread in array destructuring.

9. Add new properties to an object immutably.

10. Practice spread with nested arrays.

🔄 10. Unary Operators Tasks

- Operators: +, -, ++, --, !, typeof, delete

Tasks:1. Use unary + to convert string to number.

2. Use unary - to negate a number.

3. Pre-increment and post-increment example.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

4. Use typeof with unary.5. Use delete to remove object properties.

6. Invert a boolean with !.

7. Combine multiple unary operations.

8. Track changes using ++ in a loop.

9. Use delete in arrays.

10. Compare behavior of ++x and x++.

⛓ 11. Comma Operator Task

- Operator: ,

Tasks:

1. Use comma in for loop for multiple expressions.

2. Chain multiple variable assignments in one line.

3. Use comma to execute multiple expressions in a return statement.

4. Create an expression with side effects using ,.

5. Practice readability and usage of comma operator.

❗ 12. Nullish Coalescing Operator Tasks

- Operator: ??

Tasks:

1. Use ?? to set default value for null/undefined.

2. Compare || vs ?? behavior with falsy values.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

3. Use ?? in function arguments.4. Use ?? in a fallback UI message.

5. Chain multiple ?? checks for nested defaults.

🧭 13. Optional Chaining Operator Tasks

- Operator: ?.

Tasks:

1. Access nested object property safely.

2. Use ?. in method call.

3. Combine ?. with ??.

4. Avoid runtime errors with optional chaining.

5. Access deep properties of an API response safely.