

**Q1. Explain the role of operators in JavaScript. Why are they essential in programming?**

Operators in JavaScript are special symbols or keywords used to perform operations on values (also known as operands). They are essential in programming because they allow you to manipulate data and perform calculations or logic that enable your program to function as intended. Without operators, you wouldn't be able to do things like add numbers, compare values, or combine strings, which are fundamental to writing any meaningful program.

**Q2. Describe the categorization of operators in JavaScript based on their functionality. Provide examples for each category.**

JavaScript operators can be categorized based on their functionality:

**1. Arithmetic Operators: Used to perform mathematical operations.**

- Example:
- `let a = 5;`
- `let b = 2;`
- `let sum = a + b; // sum = 7`
- `let diff = a - b; // diff = 3`
- `let product = a * b; // product = 10`
- `let quotient = a / b; // quotient = 2.5`
- `let remainder = a % b; // remainder = 1`

**2. Assignment Operators: Used to assign values to variables.**

- Example:
- `let x = 10; // x gets the value 10`
- `x += 5; // x = x + 5 -> x becomes 15`
- `x -= 3; // x = x - 3 -> x becomes 12`

**3. Comparison Operators: Used to compare values and return boolean results.**

- Example:
- `let a = 5;`
- `let b = 10;`
- `console.log(a == b); // false`
- `console.log(a != b); // true`
- `console.log(a > b); // false`
- `console.log(a < b); // true`

**4. Logical Operators: Used to perform logical operations on boolean values.**

- Example:

- `let x = true;`
- `let y = false;`
- `console.log(x && y); // false (logical AND)`
- `console.log(x || y); // true (logical OR)`
- `console.log(!x); // false (logical NOT)`

5. **Conditional (Ternary) Operator:** Used to assign values based on a condition.

- Example:
- `let age = 18;`
- `let canVote = (age >= 18) ? "Yes" : "No"; // "Yes"`

6. **Increment/Decrement Operators:** Used to increase or decrease a variable's value by 1.

- Example:
- `let num = 5;`
- `num++; // num becomes 6 (post-increment)`
- `num--; // num becomes 5 (post-decrement)`

7. **String Operators:** Used for concatenating strings.

- Example:
- `let firstName = "John";`
- `let lastName = "Doe";`
- `let fullName = firstName + " " + lastName; // "John Doe"`

**Q3. Differentiate between unary, binary, and ternary operators in JavaScript. Give examples of each.**

1. **Unary Operators:** Operate on a single operand.

- Example:
  - Increment/Decrement: `num++`, `num--`
  - Logical NOT: `!x` (inverts a boolean value)
  - Unary minus: `-x` (negates a number)
- `let x = 5;`
- `console.log(++x); // 6 (pre-increment)`
- `console.log(-x); // -5 (unary minus)`

2. **Binary Operators:** Operate on two operands.

- Example:

- Arithmetic: +, -, \*, /
- Comparison: ==, !=, >, <

- `let a = 10, b = 5;`
- `console.log(a + b); // 15 (binary addition)`
- `console.log(a > b); // true (binary comparison)`

3. Ternary Operator: A conditional operator that operates on three operands.

- Example:
- `let age = 18;`
- `let result = (age >= 18) ? "Adult" : "Minor"; // "Adult"`

**Q4. Discuss the precedence and associativity of operators in JavaScript. Why is understanding these concepts important?**

- Precedence refers to the order in which operators are evaluated in an expression. Operators with higher precedence are evaluated before those with lower precedence.
  - Example: \* has higher precedence than +, so in `2 + 3 * 4`, `3 * 4` is evaluated first, resulting in `2 + 12 = 14`.
- Associativity defines the direction in which operators are evaluated when they have the same precedence.
  - Left-to-right associativity (default for most operators): For example, `a - b - c` is evaluated as `(a - b) - c`.
  - Right-to-left associativity (used for assignment operators and some others): For example, `a = b = 10` is evaluated as `a = (b = 10)`.

Understanding precedence and associativity is crucial to avoid unexpected results and ensure your expressions behave as intended.

**Q5. Write a JavaScript program that calculates the simple interest using the formula Simple Interest = (principal \* rate \* time) / 100.**

```
function calculateSimpleInterest(principal, rate, time) {
  let interest = (principal * rate * time) / 100;
  return interest;
}
```

`// Example usage:`

```
let principal = 1000; // Principal amount in currency
let rate = 5;        // Rate of interest in percentage
let time = 2;        // Time in years
```

```
let interest = calculateSimpleInterest(principal, rate, time);
```

```
console.log("Simple Interest: " + interest); // Output: 100
```

**Q6. Write a JavaScript program to calculate the Body Mass Index (BMI) using the formula  $BMI = \text{weight (kg)} / (\text{height} * \text{height})$ .**

```
function calculateBMI(weight, height) {  
    let bmi = weight / (height * height);  
    return bmi;  
}
```

**// Example usage:**

```
let weight = 70; // Weight in kilograms
```

```
let height = 1.75; // Height in meters
```

```
let bmi = calculateBMI(weight, height);
```

```
console.log("BMI: " + bmi.toFixed(2)); // Output: BMI: 22.86
```

**Q7. Write a program in JavaScript to calculate the area of a circle given its radius value of 10. Use appropriate arithmetic operators.**

```
function calculateAreaOfCircle(radius) {  
    let area = Math.PI * radius * radius;  
    return area;  
}
```

**// Example usage:**

```
let radius = 10;
```

```
let area = calculateAreaOfCircle(radius);
```

```
console.log("Area of Circle: " + area.toFixed(2)); // Output: Area of Circle: 314.16
```

**In these programs, arithmetic operators like  $*$  (multiplication) and  $/$  (division) are used to calculate the desired values.**