

Table of Contents

| **Sl.no** | **Topic** | **Page No** |
| --- | --- | --- |
| 1 | Operators & Expressions | 03 |
| 2 | Arithmetic Operator | 03 |
| 3 | Assignment Operator | 04 |
| 4 | Comparison Operators | 05 |
| 5 | Logical Operators | 07 |
| 6 | Ternary Operators | 08 |
| 7 | Type Operators | 09 |
| 8 | Bitwise Operators | 10 |
| 9 | Conclusion | 11 |
| 10 | Brain Teaser | 11 |
| 11 | Reference | 12 |

1. **Key Concepts –Operators and Expressions**
   1. Operators- introduction

In JavaScript, **operators** are symbols or keywords used to perform operations on values (called operands). They can be categorized into several types

* 1. Types of Operators

**1. Arithmetic Operators**

**2. Assignment Operators**

**3. Comparison Operators**

**4. Logical Operators**

**5. Ternary Operator**

**6. Type Operators**

**7. Bitwise Operators**

Let we discuss one by one,

**1. Arithmetic Operators**

**These are used to perform mathematical calculations**.

| **+ (Addition)** | Adds two values.  Example: 5 + 3 → 8 |
| --- | --- |
| **- (Subtraction)** | Subtracts one value from another.  Example: 5 - 3 → 2 |
| **\* (Multiplication)** | Multiplies two values. Example: 5 \* 3 → 15 |
| **/ (Division)** | Divide one value by another.  Example: 6 / 3 → 2 |
| **% (Modulus)** | Returns the remainder of a division.  Example: 5 % 3 → 2 |
| **++ (Increment)** | Increases a value by 1.  Example: let x = 5; x++ → 6 |
| **-- (Decrement)** | These are used to assign values to variables. Assigns a value to a variable. Example: let x = 10; |

**Code Snippets**

**Example for Arithmetic Operators:**

| let num1 = 10; let num2 = 5; // Addition (+) let sum = num1 + num2; console.log("Sum:", sum); // Output: Sum: 15 // Subtraction (-) let difference = num1 - num2; console.log("Difference:", difference); // Output: Difference: 5 // Multiplication (\*) let product = num1 \* num2; console.log("Product:", product); // Output: Product: 50 // Division (/) let quotient = num1 / num2; console.log("Quotient:", quotient); // Output: Quotient: 2 // Modulus (%) let remainder = num1 % num2; console.log("Remainder:", remainder); // Output: Remainder: 0 // Increment (++) num1++; console.log("Incremented num1:", num1); // Output: Incremented num1: 11 // Decrement (--) num2--; console.log("Decremented num2:", num2); // Output: Decremented num2: 4 |
| --- |

**2. Assignment Operators:**

* These are used to assign values to variables.

| **=** | **Assigns a value to a variable.**  **Example: let x = 10;** |
| --- | --- |
| **+=** | **Adds the right operand to the left operand and assigns the result. Example: x += 5 → x = x + 5** |
| **-=** | **Subtracts the right operand from the left operand and assigns the result.**  **Example: x -= 5 → x = x - 5** |
| **\*=** | **Multiply the right operand from the left operand and assign the result.**  **Example: x \*= 5 → x = x \* 5** |
| **/=** | **Divide the right operand from the left operand and assign the result.**  **Example: x /= 5 → x = x / 5** |
| **%=** | **Make modules to right operand from the left operand and assign the result.**  **Example: x %= 5 → x = x % 5** |

**Code Snippets:**

**Example program for Assignment Operators in JavaScript**

| let a = 10;  let b = 5;  // Using the addition assignment (+=) a += b; // Equivalent to a = a + b console.log("a += b:", a); // Output: a += b: 15 // Using the subtraction assignment (-=) a -= b; // Equivalent to a = a - b console.log("a -= b:", a); // Output: a -= b: 10 // Using the multiplication assignment (\*=) a \*= b; // Equivalent to a = a \* b console.log("a \*= b:", a); // Output: a \*= b: 50 // Using the division assignment (/=) a /= b; // Equivalent to a = a / b console.log("a /= b:", a); // Output: a /= b: 10 // Using the modulus assignment (%=) a %= b; // Equivalent to a = a % b console.log("a %= b:", a); // Output: a %= b: 0 |
| --- |

**3. Comparison Operators:**

**These are used to compare two values.**

| **== (Equal to)** | **Checks if two values are equal (without considering data type).**  **Example: 5 == '5' → true** |
| --- | --- |
| **=== (Strict equal to)** | **Checks if two values are equal and of the same type.**  **Example: 5 === '5' → false** |
| **!= (Not equal to)** | **!= (Not equal to)**  **Checks if two values are not equal (without considering type).**  **Example: 5 != '5' → false** |
| **!== (Strict not equal to)** | **Checks if two values are not equal or not of the same type.**  **Example: 5 !== '5' → true** |
| **> (Greater than)** | **Checks if the left operand is greater than the right operand.**  **Example: 5 > 3 → true** |
| **< (Less than)** | **Checks if the left operand is less than the right operand.**  **Example: 3 < 5 → true** |
| **>= (Greater than or equal to)** | **Checks if the left operand is greater than or equal to the right operand. Example: 5 >= 5 → true** |
| **<= (Less than or equal to)** | **Checks if the left operand is less than or equal to the right operand.**  **Example: 3 <= 5 → true** |

**Code Snippets:**

**Example program for Comparison Operators in JavaScript**

| Example program for Comparison Operators in JavaScript  let a = 10; let b = 5; let c = "10";  // Using equality operator (==) console.log("a == c:", a == c);  // Output: a == c: true (because '10' == 10 in loose comparison)  // Using strict equality operator (===) console.log("a === c:", a === c); // Output: a === c: false (because '10' is a string and 10 is a number)  // Using inequality operator (!=) console.log("a != b:", a != b); // Output: a != b: true (because 10 is not equal to 5)  // Using strict inequality operator (!==) console.log("a !== c:", a !== c); // Output: a !== c: true (because '10' is a string and 10 is a number)  // Using greater than operator (>) console.log("a > b:", a > b); // Output: a > b: true (because 10 is greater than 5)  // Using less than operator (<) console.log("b < a:", b < a); // Output: b < a: true (because 5 is less than 10)  // Using greater than or equal to operator (>=) console.log("a >= b:", a >= b); // Output: a >= b: true (because 10 is greater than or equal to 5)  // Using less than or equal to operator (<=) console.log("b <= a:", b <= a); // Output: b <= a: true (because 5 is less than or equal to 10) |
| --- |

**4. Logical Operators**

**These are used to combine multiple conditions.**

| **&& (Logical AND)** | **Returns true if both operands are true. Example: true && false → false** |
| --- | --- |
| **|| (Logical OR)** | **Returns true if at least one operand is true.**  **Example: true || false → true** |
| **! (Logical NOT)** | **Reverses the logical state of its operand.**  **Example: !true → false** |

**Code Snippets:**

**Example program for Logical Operators in JavaScript**

| // Example program for Logical Operators in JavaScript  let a = true; let b = false;  // Using Logical AND (&&) console.log("a && b:", a && b); // Output: a && b: false (because both must be true for AND)  // Using Logical OR (||) console.log("a || b:", a || b); // Output: a || b: true (because at least one must be true for OR)  // Using Logical NOT (!) console.log("!a:", !a); // Output: !a: false (NOT negates the value, so true becomes false) console.log("!b:", !b); // Output: !b: true (NOT negates the value, so false becomes true) |
| --- |

**5. Ternary Operator**

**This is a shorthand for if-else statements.**

**condition ? value\_if\_true : value\_if\_false**

**Example: let result = (5 > 3) ? "Yes" : "No" → "Yes"**

**Code Snippets:**

**Example program for Ternary Operators in JavaScript**

| **// Example program for Ternary Operator in JavaScript**  let age = 18;  // Ternary operator: condition ? value\_if\_true : value\_if\_false let result = (age >= 18) ? "You are an adult." : "You are a minor."; console.log(result); // Output: You are an adult.  // Changing the value of age age = 16; let result2 = (age >= 18) ? "You are an adult." : "You are a minor."; console.log(result2); // Output: You are a minor. |
| --- |

### Explanation:

**The ternary operator is a shorthand for an if-else statement:**

* **Condition: (age >= 18)**
* **True value: "You are an adult."**
* **False value: "You are a minor."**
* **If the condition (age >= 18) is true, it returns the first value "You are an adult."**
* **If the condition is false, it returns the second value "You are a minor."**

**6. Type Operators**

**These are used to check or convert data types.**

* **typeof  
  Returns the type of a variable or expression.  
  Example: typeof 5 → "number"**
* **instanceof  
  Checks if an object is an instance of a particular class.  
  Example: [] instanceof Array → true**

**Code Snippets:**

| let name = "John"; // String let age = 25; // Number let isStudent = true; // Boolean let hobbies = ["Reading", "Coding"]; // Array let address = { city: "New York", zip: 10001 }; // Object let nothing = null; // Null let notDefined; // Undefined let bigNumber = 12345678901234567890n; // BigInt let uniqueId = Symbol("id"); // Symbol // Using typeof to check data types console.log("Type of name:", typeof name); // Output: string console.log("Type of age:", typeof age); // Output: number console.log("Type of isStudent:", typeof isStudent); // Output: boolean console.log("Type of hobbies:", typeof hobbies); // Output: object console.log("Type of address:", typeof address); // Output: object console.log("Type of nothing:", typeof nothing); // Output: object console.log("Type of notDefined:", typeof notDefined); // Output: undefined console.log("Type of bigNumber:", typeof bigNumber); // Output: bigint console.log("Type of uniqueId:", typeof uniqueId); // Output: symbol |
| --- |

**7. Bitwise Operators**

These perform bit-level operations on numbers.

* & (AND), | (OR), ^ (XOR), << (Left shift), >> (Right shift)

| let a = 5; // Binary: 0101 let b = 3; // Binary: 0011  let andResult = a & b; // 0101 & 0011 = 0001 (1 in decimal) console.log("a & b:", andResult); // Output: a & b: 1   let orResult = a | b; // 0101 | 0011 = 0111 (7 in decimal) console.log("a | b:", orResult); // Output: a | b: 7   let xorResult = a ^ b; // 0101 ^ 0011 = 0110 (6 in decimal) console.log("a ^ b:", xorResult); // Output: a ^ b: 6   let notResult = ~a; // ~0101 = 1010 (Two's complement: -6 in decimal) console.log("~a:", notResult); // Output: ~a: -6   let leftShift = a << 1; // 0101 << 1 = 1010 (10 in decimal) console.log("a << 1:", leftShift); // Output: a << 1: 10   let rightShift = a >> 1; // 0101 >> 1 = 0010 (2 in decimal) console.log("a >> 1:", rightShift); // Output: a >> 1: 2   let unsignedRightShift = a >>> 1; // 0101 >>> 1 = 0010 (2 in decimal) console.log("a >>> 1:", unsignedRightShift); // Output: a >>> 1: 2 |
| --- |

### Explanation:

* **& (AND)**: Compares each bit of two numbers. The result is 1 if both bits are 1, otherwise 0.
* **| (OR)**: Compares each bit of two numbers. The result is 1 if at least one bit is 1.
* **^ (XOR)**: Compares each bit of two numbers. The result is 1 if the bits are different, otherwise 0.
* **~ (NOT)**: Inverts each bit of the number (bitwise complement).
* **<< (Left Shift)**: Shifts all bits to the left by a specified number of positions, filling with 0s.
* **>> (Right Shift)**: Shifts all bits to the right, preserving the sign bit for signed numbers.
* **>>> (Unsigned Right Shift)**: Shifts all bits to the right, filling with 0s (does not preserve the sign bit).

**4. Conclusion:**

Operators in JavaScript are essential tools for performing calculations, comparisons, logical evaluations, and data manipulation. They include categories like arithmetic, assignment, comparison, logical, bitwise, and ternary operators, each serving specific purposes.

Mastering these operators allows for writing efficient code, evaluating conditions, and controlling program flow effectively, making them fundamental for building dynamic and interactive applications.

**5. Brain teaser :**

### Quiz Question: Operators in JavaScript

**Question 1.:  
You are given the following JavaScript code:**

| let x = 10;  let y = 5;  let result = (x > y) && (x < 15) ? x + y : x - y;  console.log(result); |
| --- |

**Options:**

1. 15
2. 5
3. 10
4. 0

### Question 2: Logical Operators

**Question:**What will be the output of the following code?

| let a = true;  let b = false;  let c = a || b && !a;  console.log(c);  **Options:**   1. true 2. false 3. undefined 4. null |
| --- |

### Question 3: Comparison Operators

**Question:**What will the following code output?

| let x = "10";  let y = 10;  console.log(x == y, x === y);  **Options:**   1. true false 2. true true 3. false false 4. false true |
| --- |

**4. References**

### Official Documentation:

1. [**MDN Web Docs - JavaScript**](https://developer.mozilla.org/en-US/docs/Web/JavaScript)**A comprehensive resource with tutorials, guides, and references, ideal for beginners and experienced developers.**

### Interactive Tutorials:

1. **freeCodeCamp - JavaScript Algorithms and Data Structures  
   An excellent hands-on course with exercises to learn JavaScript basics and algorithms.**
2. [**JavaScript.info**](https://javascript.info/)**A detailed and beginner-friendly guide to modern JavaScript, starting from the basics.**
3. **W3Schools - JavaScript Tutorial  
   A simple and easy-to-understand introduction to JavaScript with examples.**

### Online Coding Platforms:

1. **CodeAcademy - Learn JavaScript  
   Offers interactive lessons to learn JavaScript fundamentals step-by-step.**
2. **HackerRank - JavaScript Skills Practice  
   Perfect for practicing coding tasks related to JavaScript basics.**