

Fundamentals of JS-II

Arithmetic Operators:

Arithmetic operators are used to perform mathematical operations on numerical values.

Addition (+): Adds two values.

- Example: `a+b`

Subtraction (-): Subtracts the right operand from the left operand.

- Example: `a-b`

Multiplication (*): Multiplies two values.

- Example: `a*b`

Division (/): Divides the left operand by the right operand.

- Example: `a/b`

Modulus (%): Returns the remainder of the division of the left operand by the right operand.

- Example: `a%b`

Exponentiation (>):** Raises the left operand to the power of the right operand.

- Example: `a**b`

Comparison Operators:

Comparison operators are used to compare two values and return a Boolean result.

Equal (==): Returns true if the values on both sides are equal.

- Example: `a==b`

Not Equal (!=): Returns true if the values on both sides are not equal.

- Example: `a!=b`

Greater Than (>): Returns true if the left operand is greater than the right operand.

- Example: `a>b`

Less Than (<): Returns true if the left operand is less than the right operand.

- Example: `a<b`

Greater Than or Equal To (>=): Returns true if the left operand is greater than or equal to the right operand.

- Example: `a >= b`

Less Than or Equal To (`<=`): Returns true if the left operand is less than or equal to the right operand.

- Example: `a <= b`

Logical Operators:

Logical operators are used to perform logical operations on Boolean values.

AND (and): Returns true if both the left and right operands are true.

- Example: `a and b`

OR (or): Returns true if at least one of the operands is true.

- Example: `a or b`

NOT (not): Returns true if the operand is false and vice versa.

- Example: `not a`

Type Conversion:

Type conversion is the process of converting one data type to another.

Implicit Type Conversion (Coercion): Automatically performed by the interpreter.

- Example: `int_var = 5 + 2.0` (Here, the integer 5 is implicitly converted to a float)

Explicit Type Conversion (Casting): Done by the programmer using predefined functions.

- Example: `str_var = str(42)` (Here, the integer 42 is explicitly converted to a string)

Type Coercion:

Type coercion is the automatic conversion of one data type to another.

- Example of Type Coercion

```
const value1 = "5";
const value2 = 9;
let sum = value1 + value2;

console.log(sum); //output: 59
```

- In this example, num_str is explicitly converted from a string to an integer before adding it to num_int.

Overall Importance of this lecture

- These concepts are foundational for building algorithms, making decisions, and performing various operations in JavaScript.
- They are essential for creating dynamic and interactive web applications.
- Understanding operator precedence helps in writing expressions that are evaluated as intended.
- Type conversion and coercion play a crucial role in managing different types of data.

In summary, mastering these concepts is fundamental for anyone working with JavaScript, whether for web development, server-side programming, or any other application where JavaScript is used. They provide the tools needed to manipulate data and control the flow of a program effectively.

Reference

Arithmetic Operators - <https://javascript.info/comparison>

Comparison Operators - <https://javascript.info/comparison>

Logical Operators - <https://javascript.info/logical-operators>

Type conversion - https://developer.mozilla.org/en-US/docs/Glossary/Type_Conversion

Type coercion - https://developer.mozilla.org/en-US/docs/Glossary/Type_coercion