Arithmetic operators (+, -, *, /, %, **)

Sure! Let's explore **Arithmetic Operators** in JavaScript in detail.

🔢 What Are Arithmetic Operators?

Arithmetic operators are used to **perform mathematical operations** on numbers (both literals and variables).

🔷 List of Arithmetic Operators in JavaScript

Operator	Name	Description	Example
`+`	Addition	Adds two numbers	`5 + 2 = 7`
`-`	Subtraction	Subtracts the right operand from the left	`5 - 2 = 3`
`*`	Multiplication	Multiplies two numbers	`5 * 2 = 10`
`/`	Division	Divides left operand by right operand	`10 / 2 = 5`
`%`	Modulus (Remainder)	Returns remainder of division	`5 % 2 = 1`
`**`	Exponentiation	Raises the first operand to the power of the second	`2 ** 3 = 8`

✅ Examples

➕ Addition (`+`)

```
let x = 10;
let y = 5;
console.log(x + y); // 15
```

🔶 Also used for **string concatenation**:

```
console.log("Hello " + "World"); // Hello World
```

➖ Subtraction ()

```
let x = 10;
let y = 5;
console.log(x - y); // 5
```

✖️ Multiplication ()

```
let x = 4;
let y = 3;
console.log(x * y); // 12
```

➗ Division (/)

```
let x = 10;
let y = 2;
console.log(x / y); // 5
```

🔶 Dividing by zero returns **Infinity** or **-Infinity**:

```
console.log(10 / 0);  // Infinity
console.log(-10 / 0); // -Infinity
```

🔢 Modulus (%)

Returns **remainder** of division:

```
let x = 10;
let y = 3;
```

```
console.log(x % y); // 1
```

Useful for:

- Checking even/odd numbers:

```
console.log(5 % 2 === 0 ? "Even" : "Odd"); // Odd
```

🧱 Exponentiation (`*`)

Raises base to the power of exponent:

```
console.log(2 ** 3); // 8 (2 * 2 * 2)
console.log(5 ** 2); // 25
```

Equivalent to:

```
Math.pow(2, 3); // Also returns 8
```

ℹ️ Notes

- These operators follow **operator precedence** (e.g., and `/` are evaluated before `+` and).
- You can use **parentheses** `()` to control the order:

```
console.log(2 + 3 * 4);     // 14
console.log((2 + 3) * 4);   // 20
```

✅ Summary

Operation	Example	Result
Addition	`4 + 2`	6
Subtraction	`4 - 2`	2

Multiplication	`4 * 2`	8
Division	`4 / 2`	2
Modulus	`4 % 2`	0
Exponentiation	`4 ** 2`	16

Would you like to continue with the **Comparison Operators** next?