Theory: Arithmetic operators

5 minutes 0 / 5 problems solved

In real life, we often perform arithmetic operations. They help us calculate the price of purchases, determine the size dimensions of the room or garden area, calculate the number of people invited to a party, and so on. Programmers often have to use the same operations to write scripts.

§1. Binary arithmetic operators

The JavaScript programming language provides operators to perform arithmetic operations. They are called **binary** because they apply to two **operands** (objects over which the operation is performed).

- addition `+`

```
console.log(12 + 26); // 38
console.log(5 + 4.5); // 9.5
```

Usually `+` is used to add numbers, but if you apply this operator to strings, it combines them into one:

```
console.log("My name is " + "John"); // My name is John
```

- subtraction `-`

```
console.log(5 - 3);   // 2
console.log(6 - 0.1); // 5.9
```

- multiplication `*`

```
console.log(10 * 4);  // 40
console.log(2 * 1.5); // 3
```

- division `/`

```
console.log(8 / 2); // 4
console.log(12 / 5); // 2.4
```

- remainder `%`. This operator finds the residue from the division:

```
console.log(10 % 3); // 1, because 10 divided by 3 leaves a remainder of 1
console.log(12 % 4); // 0, because 12 divided by 4 leaves no remainder
```

- exponentiation `**`

```
console.log(2 ** 3); // 8, because (2 * 2 * 2) is 8
```

§2. Writing complex expressions

Arithmetic operations can be combined to write more complex expressions:

```
console.log(1 + 3 * 4 - 2) // 11
```

The calculation order follows the basic rules of arithmetic operations. Multiplication has a higher priority level than addition and subtraction, so `3 * 4` is calculated first.

To specify the order of execution, we can use **parentheses** as in the following example:

```
console.log((1 + 3) * (4 - 2)); // 8
```

Current topic:

Arithmetic operators

Topic depends on:

× Strings and numbers

Topic is required for:

Functions

Template literals

Type conversion

ForEach method

Table of contents:

↑ Arithmetic operators

§1. Binary arithmetic operators

§2. Writing complex expressions

§3. Unary operators

§4. Precedence order

Feedback & Comments

As in arithmetics, parentheses can be nested in each other. Feel free to use them for better code clarity.

§3. Unary operators

Unary refers to operators that apply to a single operand.

- The unary plus operator indicates a positive value. It's an optional operator if you only work with numbers:

```
console.log(+7); // 7
```

- The unary minus operator makes a value or an expression negative:

```
console.log(-9);         // -9
console.log(-(100 + 5)); // -105
```

§4. Precedence order

If several operators are involved in an expression, they will be executed in the order of priority. The list below is sorted from the highest to the lowest precedence level:

- parentheses
- unary plus/minus
- exponentiation
- multiplication, division
- addition and subtraction

The order of performing arithmetic operations in JavaScript basically follows that in mathematics, so you don't have to learn it specifically.

Report a typo

142 users liked this theory. 1 didn't like it. What about you?

Start practicing

Comments (3) Hints (0) Useful links (0) Show discussion