Theory: Increment and decrement

7 minutes 0 / 5 problems solved

In this topic, we will discuss some of the most frequently used operators in programming: **increment** and **decrement**. They exist in many languages including JavaScript. These two operators are used to concisely record the operations of increasing and decreasing the value of a variable by one. In many programming tasks, it is required to change the value this way, so knowing increment and decrement is fundamental.

§1. Using ++ and -- in JavaScript

JavaScript has two opposite operations called **increment** (`++`) and **decrement** (`--`) to increase or decrease the value of a variable by one, respectively. For example:

```
let a = 10;
let b = 10;
console.log(++a); // 11
console.log(--b); // 9
```

The code above actually gives the same result as the code below:

```
let a = 10;
let b = 10;
console.log(a + 1); // 11
console.log(b - 1); // 9
```

§2. Prefix increment

Both **increment** and **decrement** operators have two forms that are very important when using the result in the current statement:

- **prefix** (`++n` or `--n`) increases/decreases the value of a variable before it is used;
- **postfix** (`n++` or `n--`) increases/decreases the value of a variable after it is used.

Let's look at the prefix increment:

```
let a = 4;
let b = ++a;

console.log(a); // 5
console.log(b); // 5
```

In this case, the value of `a` has been incremented and then assigned to `b`. Thus, `b` is 5.

§3. Postfix increment

Postfix increment increases or decreases the value of a variable *after* it is used. Consider an example:

```
let a = 4;
let b = a++;

console.log(a); // 5
console.log(b); // 4
```

In JavaScript, the postfix operator has higher precedence than the assignment operator. It returns the original value of `a`, not the incremented one. That's why when we assign `a++` to `b`, we actually assign 4, while `a` itself has already been incremented. So, `b` is 4 and `a` is 5.

Current topic:

Increment and decrement

Topic depends on:

- ✕ Variables
- ✕ Functions

Topic is required for:

- For loops
- While loops

Table of contents:

If that's still not clear enough for you, take a look at the code:

```
let a = 4;
console.log(a++ + a); // 9
```

§4. Conclusion

JS uses two operators, increment and decrement, for shorter writing of operations that increase or decrease the variable value by one. There are prefix and postfix forms of their recording: the prefix changes the variable before using it, and the postfix changes it after. These operators are useful, but don't abuse them if you're worried about code readability.

Report a typo

64 users liked this theory. **2** didn't like it. **What about you?**

Start practicing

Comments (2) Hints (0) Useful links (0) Show discussion