JavaScript
TUTORIAL

JavaScript Logical Assignment Operators

Summary: in this tutorial, you'll learn about JavaScript logical assignment operators, including the logical OR assignment operator (`||=`), the logical AND assignment operator (`&&=`), and the nullish assignment operator (`??=`).

ES2021 introduces three logical assignment operators including:

- Logical OR assignment operator (`||=`)
- Logical AND assignment operator (`&&=`)
- Nullish coalescing assignment operator (`??=`)

The following table shows the equivalent of the logical assignments operator:

Logical Assignment Operators	Logical Operators
x ||= y	x || (x = y)
x &&= y	x && (x = y)
x ??= y	x ?? (x = y);

The Logical OR assignment operator

The logical OR assignment operator (`||=`) accepts two operands and assigns the right operand to the left operand if the left operand is falsy:

```
x ||= y
```

In this syntax, the `||=` operator only assigns `y` to `x` if `x` is falsy. For example:

```
let title;
title ||= 'untitled';

console.log(title);
```

Output:

```
untitled
```

In this example, the `title` variable is `undefined`, therefore, it’s falsy. Since the `title` is falsy, the operator `||=` assigns the `'untitled'` to the `title`. The output shows the `untitled` as expected.

See another example:

```
let title = 'JavaScript Awesome';
title ||= 'untitled';

console.log(title);
```

Output:

```
'JavaScript Awesome'
```

In this example, the `title` is `'JavaScript Awesome'` so it is truthy. Therefore, the logical OR assignment operator (`||=`) doesn’t assign the string `'untitled'` to the `title` variable.

The logical OR assignment operator:

```
x ||= y
```

is equivalent to the following statement that uses the logical OR operator:

```
x || (x = y)
```

Like the logical OR operator, the logical OR assignment also short-circuits. It means that the logical OR assignment operator only performs an assignment when the `x` is falsy.

The following example uses the logical assignment operator to display a default message if the search result element is empty:

```
document.querySelector('.search-result').textContent ||= 'Sorry! No result found';
```

The Logical AND assignment operator

The logical AND assignment operator only assigns `y` to `x` if `x` is truthy:

```
x &&= y;
```

The logical AND assignment operator also short-circuits. It means that

```
x &&= y;
```

is equivalent to:

```
x && (x = y);
```

The following example uses the logical AND assignment operator to change the last name of a `person` object if the last name is truthy:

```
let person = {
    firstName: 'Jane',
    lastName: 'Doe',
};

person.lastName &&= 'Smith';

console.log(person);
```

Output:

```
{firstName: 'Jane', lastName: 'Smith'}
```

The nullish coalescing assignment operator

The nullish coalescing assignment operator only assigns `y` to `x` if `x` is `null` or `undefined`:

```
x ??= y;
```

It's equivalent to the following statement that uses the nullish coalescing operator:

```
x ?? (x = y);
```

The following example uses the nullish coalescing assignment operator to add a missing property to an object:

```
let user = {
    username: 'Satoshi'
};
user.nickname ??= 'anonymous';

console.log(user);
```

Output:

```
{username: 'Satoshi', nickname:'anonymous'}
```

In this example, the `user.nickname` is `undefined`, therefore, it's nullish. The nullish coalescing assignment operator assigns the string `'anonymous'` to the `user.nickname` property.

The following table illustrates how the logical assignment operators work:

Summary

- The logical OR assignment (`x ||= y`) operator only assigns `y` to `x` if `x` is falsy.
- The logical AND assignment (`x &&= y`) operator only assigns `y` to `x` if `x` is truthy.
- The nullish coalescing assignment (`x ??= y`) operator only assigns `y` to `x` if `x` is nullish.