JavaScript Assignment Operators

Summary: in this tutorial, you will learn how to use JavaScript assignment operators to assign a value to a variable.

Introduction to JavaScript assignment operators

In JavaScript, an assignment operator (`=`) assigns a value to a variable.

Here's the syntax of the assignment operator:

```
let a = b;
```

In this syntax, JavaScript evaluates the expression `b` first and assigns the result to the variable `a` .

For example, the following declares the `counter` variable and initializes its value to zero:

```
let counter = 0;
```

The following example increases the `counter` variable by one and assigns the result to the `counter` variable:

```
let counter = 0;
counter = counter + 1;
```

When evaluating the second statement, JavaScript evaluates the expression on the right-hand first (`counter + 1`) and assigns the result to the `counter` variable. After the second assignment, the `counter` variable is `1` .

To make the code more concise, you can use the `+=` operator like this:

```
let counter = 0;
counter += 1;
```

In this syntax, you don't have to repeat the `counter` variable twice in the assignment.

The following table illustrates assignment operators that are shorthand for another operator and the assignment:

Operator	Meaning	Description
a = b	a = b	Assigns the value of b to a.
a += b	a = a + b	Assigns the result of a plus b to a.

Operator	Meaning	Description
`a -= b`	`a = a - b`	Assigns the result of a minus b to a.
`a *= b`	`a = a * b`	Assigns the result of a times b to a.
`a /= b`	`a = a / b`	Assigns the result of a divided by b to a.
`a %= b`	`a = a % b`	Assigns the remainder of a and b to a
`a &=b`	`a = a & b`	Assigns the result of a AND b to a.
`a |=b`	`a = a | b`	Assigns the result of a OR b to a.
`a ^=b`	`a = a ^ b`	Assigns the result of a XOR b to a.
`a <<= b`	`a = a << b`	Assigns the result of a shifted left by b to a.
`a >>= b`	`a = a >> b`	Assigns the result of a shifted right (sign preserved) by b to a.
`a >>>= b`	`a = a >>> b`	Assigns the result of a shifted right by b to a.

+= operator

The following example uses the += operator to add one to variable `x`:

```
let x = 10;
x += 1;
console.log(x); // 11
```

-= operator

The following example uses the `-=` operator to minus one from the variable `x`:

```
let x = 10;
x -= 1;
console.log(x); // 9
```

Output:

```
9
```

*= operator

The following example uses the `*=` operator to multiply 10 with the variable `x`:

```
let x = 10;
x *= 10;
console.log(x); // 100
```

/= operator

The following example uses the `/=` operator to divide x by 2 and assign the result back to x:

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

Output:

```
5
```

%= operator

The following example uses the `%=` operator to get the remainder of `x` is divided by `2` and assigns the remainder back to `x`:

```
let x = 5;
x = x % 2;

console.log(x);
```

Output:

```
1
```

Chaining JavaScript assignment operators

If you want to assign a single value to multiple variables, you can chain the assignment operators. For example:

```
let a = 10, b = 20, c = 30;
a = b = c; // all variables are 30
```

In this example, JavaScript evaluates from right to left. Therefore, it does the following:

```
let a = 10, b = 20, c = 30;

b = c; // b is 30
a = b; // a is also 30
```