**empty**

An **empty statement** is used to provide no statement, although the JavaScript syntax would expect one.

[**Try it**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#try_it)

[**Syntax**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#syntax)

;

Copy to Clipboard

[**Description**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#description)

The empty statement is a semicolon (;) indicating that no statement will be executed, even if JavaScript syntax requires one.

The opposite behavior, where you want multiple statements, but JavaScript only allows a single one, is possible using a [block statement](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/block), which combines several statements into a single one.

[**Examples**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#examples)

[**Empty loop body**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#empty_loop_body)

The empty statement is sometimes used with loop statements. See the following example with an empty loop body:

const arr = [1, 2, 3];

// Assign all array values to 0

for (let i = 0; i < arr.length; arr[i++] = 0) /\* empty statement \*/ ;

console.log(arr);

// [0, 0, 0]

Copy to Clipboard

[**Unintentional usage**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty#unintentional_usage)

It is a good idea to comment *intentional* use of the empty statement, as it is not really obvious to distinguish from a normal semicolon.

In the following example, the usage is probably not intentional:

if (condition); // Caution, this "if" does nothing!

killTheUniverse(); // So this always gets executed!!!