**if...else**

The **if...else** statement executes a statement if a specified condition is [truthy](https://developer.mozilla.org/en-US/docs/Glossary/Truthy). If the condition is [falsy](https://developer.mozilla.org/en-US/docs/Glossary/Falsy), another statement in the optional else clause will be executed.

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

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

if (condition)

statement1

// With an else clause

if (condition)

statement1

else

statement2

Copy to Clipboard

condition

An expression that is considered to be either [truthy](https://developer.mozilla.org/en-US/docs/Glossary/Truthy) or [falsy](https://developer.mozilla.org/en-US/docs/Glossary/Falsy).

statement1

Statement that is executed if *condition* is [truthy](https://developer.mozilla.org/en-US/docs/Glossary/Truthy). Can be any statement, including further nested if statements. To execute multiple statements, use a [block](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/block) statement ({ /\* ... \*/ }) to group those statements. To execute no statements, use an [empty](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/Empty) statement.

statement2

Statement that is executed if condition is [falsy](https://developer.mozilla.org/en-US/docs/Glossary/Falsy) and the else clause exists. Can be any statement, including block statements and further nested if statements.

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

Multiple if...else statements can be nested to create an else if clause. Note that there is no elseif (in one word) keyword in JavaScript.

if (condition1)

statement1

else if (condition2)

statement2

else if (condition3)

statement3

// …

else

statementN

Copy to Clipboard

To see how this works, this is how it would look if the nesting were properly indented:

if (condition1)

statement1

else

if (condition2)

statement2

else

if (condition3)

statement3

// …

Copy to Clipboard

To execute multiple statements within a clause, use a block statement ({ /\* ... \*/ }) to group those statements.

if (condition) {

statements1

} else {

statements2

}

Copy to Clipboard

Not using blocks may lead to confusing behavior, especially if the code is hand-formatted. For example:

function checkValue(a, b) {

if (a === 1)

if (b === 2)

console.log("a is 1 and b is 2");

else

console.log("a is not 1");

}

This code looks innocent — however, executing checkValue(1, 3) will log "a is not 1". This is because in the case of [dangling else](https://en.wikipedia.org/wiki/Dangling_else), the else clause will be connected to the closest if clause. Therefore, the code above, with proper indentation, would look like:

function checkValue(a, b) {

if (a === 1)

if (b === 2)

console.log("a is 1 and b is 2");

else

console.log("a is not 1");

}

Copy to Clipboard

In general, it is a good practice to always use block statements, especially in code involving nested if statements.

function checkValue(a, b) {

if (a === 1) {

if (b === 2) {

console.log("a is 1 and b is 2");

}

} else {

console.log("a is not 1");

}

}

Do not confuse the primitive Boolean values true and false with truthiness or falsiness of the [Boolean](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Boolean) object. Any value that is not false, undefined, null, 0, -0, NaN, or the empty string (""), and any object, including a Boolean object whose value is false, is considered [truthy](https://developer.mozilla.org/en-US/docs/Glossary/Truthy) when used as the condition. For example:

const b = new Boolean(false);

if (b) // this condition is truthy

statement

Copy to Clipboard

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

[**Using if...else**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else#using_if...else)

if (cipherChar === fromChar) {

result += toChar;

x++;

} else {

result += clearChar;

}

Copy to Clipboard

[**Using else if**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else#using_else_if)

Note that there is no elseif syntax in JavaScript. However, you can write it with a space between else and if:

if (x > 50) {

/\* do something \*/

} else if (x > 5) {

/\* do something \*/

} else {

/\* do something \*/

}

Copy to Clipboard

[**Using an assignment as a condition**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else#using_an_assignment_as_a_condition)

You should almost never have an if...else with an assignment like x = y as a condition:

if (x = y) {

/\* do something \*/

}

However, in the rare case you find yourself wanting to do something like that, the [while](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while) documentation has a [Using an assignment as a condition](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while#using_an_assignment_as_a_condition) section with an example showing a general best-practice syntax you should know about and follow.