

Strict equality (===)

The strict equality operator (`===`) checks whether its two operands are equal, returning a Boolean result. Unlike the [equality](#) operator, the strict equality operator always considers operands of different types to be different.

JavaScript Demo: Expressions - Strict equality operator

```
1 console.log(1 === 1);
2 // expected output: true
3
4 console.log('hello' === 'hello');
5 // expected output: true
6
7 console.log('1' === 1);
8 // expected output: false
9
10 console.log(0 === false);
11 // expected output: false
12
```

Run ›Reset

Syntax

`x === y`



Description

The strict equality operators (`===` and `!==`) use the

[Strict Equality Comparison Algorithm](#) [↗] to compare two operands

[Strict Equality Comparison Algorithm](#) ↳ to compare two operands.

- If the operands are of different types, return `false`.
- If both operands are objects, return `true` only if they refer to the same object.
- If both operands are `null` or both operands are `undefined`, return `true`.
- If either operand is `NaN`, return `false`.
- Otherwise, compare the two operand's values:
 - Numbers must have the same numeric values. `+0` and `-0` are considered to be the same value.
 - Strings must have the same characters in the same order.
 - Booleans must be both `true` or both `false`.

The most notable difference between this operator and the [equality](#) (`==`) operator is that if the operands are of different types, the `===` operator attempts to convert them to the same type before comparing.

Examples

Comparing operands of the same type

```
console.log("hello" === "hello"); // true
console.log("hello" === "hola");  // false

console.log(3 === 3);             // true
console.log(3 === 4);             // false

console.log(true === true);       // true
console.log(true === false);      // false

console.log(null === null);       // true
```



Comparing operands of different types

```
console.log("3" === 3);           // false
console.log(true === 1);          // false
console.log(null === undefined);  // false
```



Comparing objects

```
const object1 = {
  name: "hello"
}

const object2 = {
  name: "hello"
}

console.log(object1 === object2); // false
console.log(object1 === object1); // true
```

Specifications

Specification
ECMAScript Language Specification (ECMAScript) # sec-equality-operators

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

Strict equality (a === b)	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	4
Safari	1

WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0

 Full support

See also

- [Equality operator](#)
- [Inequality operator](#)
- [Strict inequality operator](#)