# Операторы сравнения

Все операторы сравнения возвращают true или false.

Результат сравнения можно присвоить переменной, как и любое значение.

## [Сравнение строк](https://learn.javascript.ru/comparison" \l "sravnenie-strok)

Строки сравниваются посимвольно.

Алгоритм сравнения двух строк довольно прост:

* Сначала сравниваются первые символы строк.
* Если первый символ первой строки больше (меньше), чем первый символ второй, то первая строка больше (меньше) второй. Сравнение завершено.
* Если первые символы равны, то таким же образом сравниваются уже вторые символы строк.
* Сравнение продолжается, пока не закончится одна из строк.
* Если обе строки заканчиваются одновременно, то они равны. Иначе, большей считается более длинная строка.

## [Сравнение разных типов](https://learn.javascript.ru/comparison" \l "sravnenie-raznyh-tipov)

При сравнении значений разных типов JavaScript приводит каждое из них к числу.

Например:

let a = 0;

alert( Boolean(a) ); // false

let b = "0";

alert( Boolean(b) ); // true

alert(a == b); // true

С точки зрения JavaScript, результат ожидаем. Равенство преобразует значения, используя числовое преобразование, поэтому "0" становится 0. В то время как явное преобразование с помощью Boolean использует другой набор правил.

**Строгое и нестрогое сравнение.**

**Оператор строгого равенства**===**проверяет равенство без приведения типов.**

Оператор нестрогого равенства операнды разных типов преобразует к числу.

## [Сравнение с null и undefined](https://learn.javascript.ru/comparison" \l "sravnenie-s-null-i-undefined)

**При строгом равенстве ===**

Эти значения различны, так как различны их типы.

alert( null === undefined ); // false

**При нестрогом равенстве ==**

Эти значения равны друг другу и не равны никаким другим значениям. Это специальное правило языка.

alert( null == undefined ); // true

**При использовании математических операторов и других операторов сравнения < > <= >=**

Значения null/undefined преобразуются к числам: null становится 0, а undefined – NaN.

alert( null > 0 ); // (1) false

alert( null == 0 ); // (2) false

alert( null >= 0 ); // (3) true

Сравнения преобразуют null в число, рассматривая его как 0. Поэтому выражение (3) null >= 0 истинно, а null > 0 ложно.

А для нестрогого равенства == значений undefined и null действует особое правило: эти значения ни к чему не приводятся, они равны друг другу и не равны ничему другому. Поэтому (2) null == 0 ложно.

### [Несравненное значение undefined](https://learn.javascript.ru/comparison" \l "nesravnennoe-znachenie-undefined)

### Значение undefined несравнимо с другими значениями:

alert( undefined > 0 ); // false (1)

alert( undefined < 0 ); // false (2)

alert( undefined == 0 ); // false (3)

Сравнения (1) и (2) возвращают false, потому что undefined преобразуется в NaN, а NaN – это специальное числовое значение, которое возвращает false при любых сравнениях.

Нестрогое равенство (3) возвращает false, потому что undefined равно только null, undefined и ничему больше.

### [Как избежать проблем](https://learn.javascript.ru/comparison" \l "kak-izbezhat-problem)

* Относитесь очень осторожно к любому сравнению с undefined/null, кроме случаев строгого равенства ===.
* Не используйте сравнения >= > < <= с переменными, которые могут принимать значения null/undefined, разве что вы полностью уверены в том, что делаете. Если переменная может принимать эти значения, то добавьте для них отдельные проверки.