

# Equality in Javascript

## Relevant Links

- Flanagan’s book, “Relational Expressions” section of chapter 4
- MDN’s description of equality rules<sup>1</sup>. In particular check the table for loose equality cases.
- The official description of strict equality, from the standard<sup>2</sup>

## Equality

- Javascript has two different “equality” operations, == (loose equality) and === (strict equality).
- You should almost never use ==.
- Stick to ===, which is a more strict equality test.
- Essentially == does a “type conversion” before comparing. This leads to some weird behavior (for instance it is no longer transitive). Some examples:

```
0 == "0"           // true
0 == ""            // true
"" == "0"          // false
0 === "0"          // false
false == "false"   // false
0 == false         // true
```

- Two objects are only equal if they are literally the same object:

```
var o = { foo: 2 }; var o2 = { foo: 2 };
o == o2;             // false
o == o;              // true
```

Exception: There is one case where using loose equality works well: If you want to capture both “undefined” and “null”. So `o == null` is going to be true both when `o` is undefined and when it is null.

---

<sup>1</sup>[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Equality\\_comparisons\\_and\\_sameness](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Equality_comparisons_and_sameness)

<sup>2</sup><http://ecma-international.org/ecma-262/5.1/#sec-11.9.6>