

Equality

equality in JavaScript ES5 using '==' and '==='; and introduction to the Object.is() method in ES6, which is very similar to '==='

In this lesson, we will cover object and function improvements in ES6. As with most updates in ES6, we will be able to solve the same problems as before, with less code, and more clarity.

We will start with a nitpicky subject: comparisons. Most developers prefer `===` to `==`, as the first one considers the type of its operands.

In ES6, `Object.is(a, b)` provides *same value equality*, which is almost the same as `===` except the following differences:

- `Object.is(+0, -0)` is `false`, while `-0 === +0` is `true`
- `Object.is(NaN, NaN)` is `true`, while `NaN === NaN` is `false`

```
console.log(Object.is(+0, -0)) // is `false`, while
console.log(-0 === +0)        // is `true`

Object.is(NaN, NaN)           // is `true`, while
console.log(NaN === NaN)      // is false
```



I will continue using `===` for now, and pay attention to `NaN` values, as they should normally be caught and handled prior to a comparison using the more semantic `isNaN` built-in function. For more details on `Object.is`, visit [this thorough article](#).

In the next lesson, we'll talk about another important concept - class hierarchy using mixins.

