

Trace: » [iterators\\_and\\_generators](#) » [const\\_functions](#) » [inherited\\_explicit\\_soft\\_fields](#) » [classes\\_as\\_sugar](#) » [egal](#)

## Object.eq(x, y)

An `egal` function simply makes available the internal `SameValue` function from section 9.12 of the ES5 spec. If two values are `egal`, then they are not observably distinguishable. `Egal` can be coded as `Object.eq` in ES5 as follows:

```
Object.eq = function(x, y) {
  if (x === y) {
    // 0 === -0, but they are not identical
    return x !== 0 || 1/x === 1/y;
  }

  // NaN !== NaN, but they are identical.
  // NaNs are the only non-reflexive value, i.e., if x !== x,
  // then x is a NaN.
  // isNaN is broken: it converts its argument to number, so
  // isNaN("foo") => true
  return x !== x && y !== y;
}
```

There seems to be general agreement not to add `egal` to the standard as a primitive concept. At the same time, new collection operations that look up values based on equality should use `egal` rather than `===`. It was an unfortunate price of legacy compatibility that ES5's `Array.prototype.indexOf` and `lastIndexOf` look up by `===` rather than `egal`. The result is that a `NaN` in an array cannot be found. In the absence of legacy compatibility constraints, we should avoid repeating this mistake.

If `egal` is added to a future standard, open questions are what `egal` should be named and where it should be placed. An attractive option that has been mentioned several times is `Object.eq(x, y)`.

The short- vs. long-name complaints (e.g., `keys` vs. `getOwnPropertyNames`) in reaction to ES5 have a point, and `eq` is even shorter than anything yet added to `Object`. If [encapsulated\\_hashcodes](#) results in `Object.hashcode`, then `Object.identity` might be a better name – at least it has the same length.

This reminds me: adding `Object.prototype.hashcode` should trigger adding `egal`.

— *Brendan Eich* 2010/06/09 15:05

Renamed to `Object.prototype.eq` as suggested.

— *Mark S. Miller* 2010/08/19 16:05

## References

---

Henry Baker's Equal Rights for Functional Objects or, The More Things Change, The More They Are the Same

strawman/egal.txt · Last modified: 2010/08/20 00:49 by markm

