JavaScript for Developers

Comparison

The Brown Box

Comparison

5 == '5' True

5 === '5' False

10 < '8' False

'10' < '8' True

true == 1 True

null == undefined True

null === undefined False

```
const x = [];
const y = [];
```

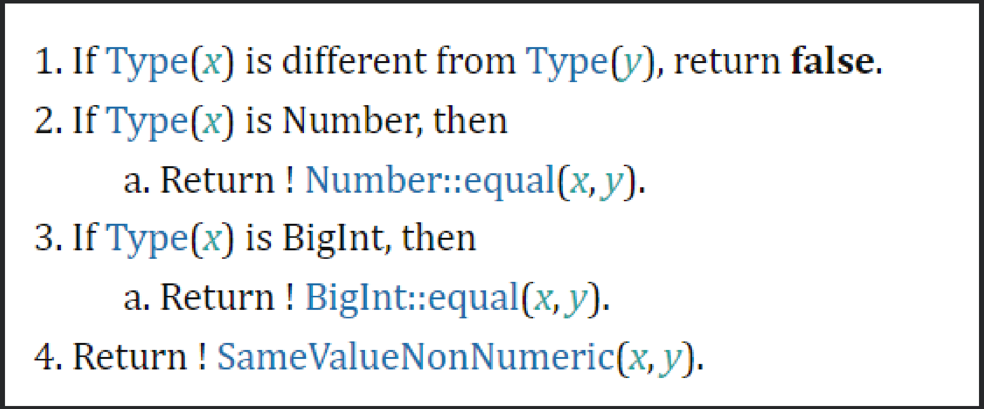
x == y False

```
const x = {};
const y = {};
```

```
JSON.stringify(x) === JSON.stringify(y)
```
True

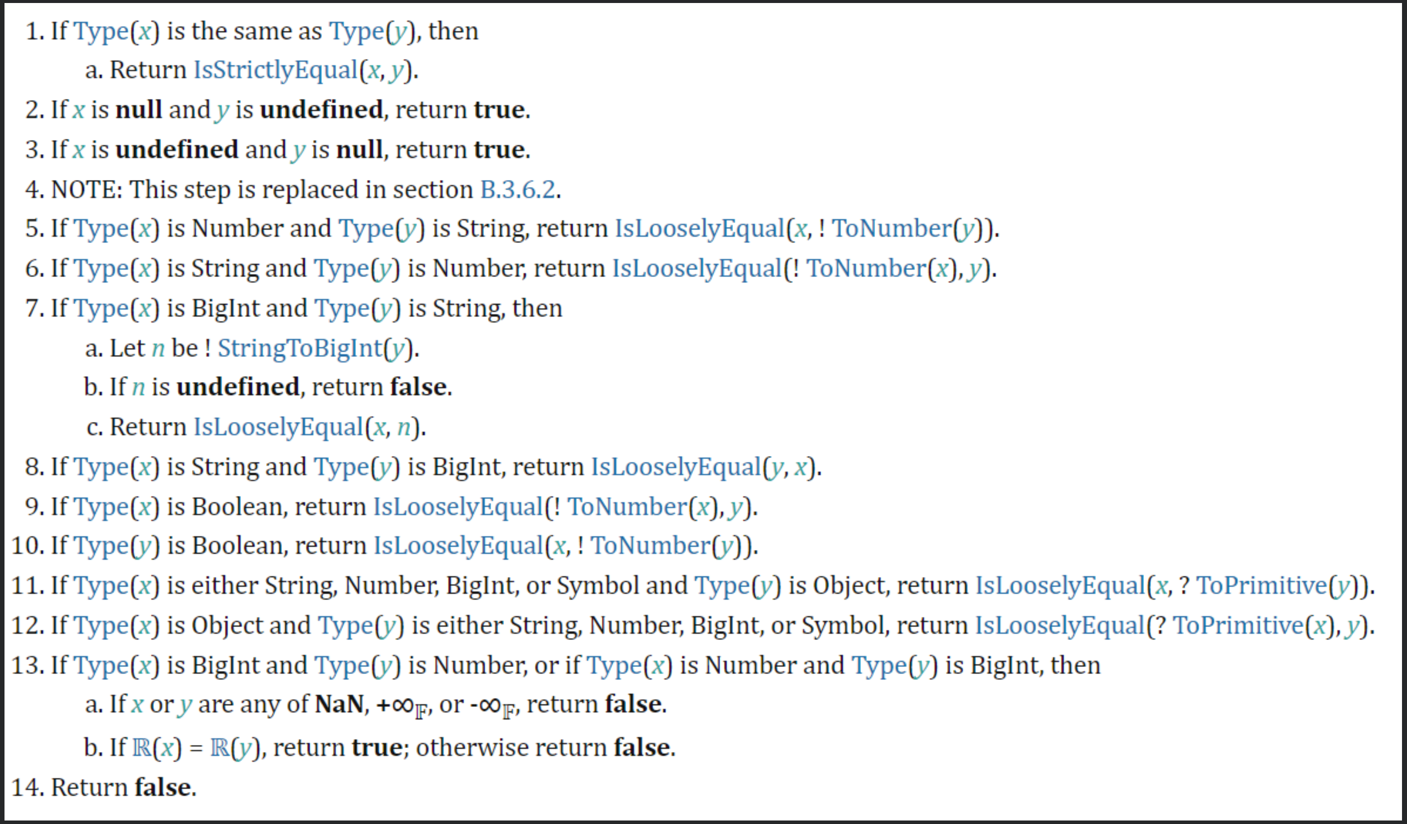
Refs

===



```
1. If Type(x) is different from Type(y), return false.
2. If Type(x) is Number, then
    a. Return ! Number::equal(x, y).
3. If Type(x) is BigInt, then
    a. Return ! BigInt::equal(x, y).
4. Return ! SameValueNonNumeric(x, y).
```

==



```
1. If Type(x) is the same as Type(y), then
    a. Return IsStrictlyEqual(x, y).
2. If x is null and y is undefined, return true.
3. If x is undefined and y is null, return true.
4. NOTE: This step is replaced in section B.3.6.2.
5. If Type(x) is Number and Type(y) is String, return IsLooselyEqual(x, ! ToNumber(y)).
6. If Type(x) is String and Type(y) is Number, return IsLooselyEqual(! ToNumber(x), y).
7. If Type(x) is BigInt and Type(y) is String, then
    a. Let n be ! StringToBigInt(y).
    b. If n is undefined, return false.
    c. Return IsLooselyEqual(x, n).
8. If Type(x) is String and Type(y) is BigInt, return IsLooselyEqual(y, x).
9. If Type(x) is Boolean, return IsLooselyEqual(! ToNumber(x), y).
10. If Type(y) is Boolean, return IsLooselyEqual(x, ! ToNumber(y)).
11. If Type(x) is either String, Number, BigInt, or Symbol and Type(y) is Object, return IsLooselyEqual(x, ? ToPrimitive(y)).
12. If Type(x) is Object and Type(y) is either String, Number, BigInt, or Symbol, return IsLooselyEqual(? ToPrimitive(x), y).
13. If Type(x) is BigInt and Type(y) is Number, or if Type(x) is Number and Type(y) is BigInt, then
    a. If x or y are any of NaN, $+\infty_{\mathbb{F}}$, or $-\infty_{\mathbb{F}}$, return false.
    b. If $\mathbb{R}(x) = \mathbb{R}(y)$, return true; otherwise return false.
14. Return false.
```

- https://tc39.es/ecma262/#sec-isstrictlyequal
- https://tc39.es/ecma262/#sec-islooselyequal

Best Practice

Use == when checking null or undefined

Otherwise, use ===

JavaScript for Developers

Comparison

The Brown Box