Object.is()

`Object.is()` 方法判断两个值是否是相同的值。

语法

```
Object.is(value1, value2);
```

参数

`value1`
需要比较的第一个值。

`value2`
需要比较的第二个值。

返回值

表示两个参数是否相同的 `Boolean` 。

描述

`Object.is()` 判断两个值是否相同。如果下列任何一项成立，则两个值相同：

- 两个值都是 `undefined`
- 两个值都是 `null`
- 两个值都是 `true` 或者都是 `false`
- 两个值是由相同个数的字符按照相同的顺序组成的字符串
- 两个值指向同一个对象
- 两个值都是数字并且
    - 都是正零 `+0`
    - 都是负零 `-0`
    - 都是 `NaN`
    - 都是除零和 `NaN` 外的其它同一个数字

这种相等性判断逻辑和传统的 `==` 运算符所用的不同， `==` 运算符会对它两边的操作数做隐式类型转换（如果它们类型不同），然后才进行相等性比较，（所以才会有类似 `"" == false` 为 `true` 的现象），但 `Object.is` 不会做这种类型转换。

这与 `===` 运算符也不一样。 `===` 运算符（和 `==` 运算符）将数字值 `-0` 和 `+0` 视为相等，并认为 `Number.NaN` 不等于 `NaN` 。

示例

```
Object.is('foo', 'foo');     // true
Object.is(window, window);   // true

Object.is('foo', 'bar');     // false
Object.is([], []);           // false

var test = { a: 1 };
Object.is(test, test);       // true

Object.is(null, null);       // true

// 特例
Object.is(0, -0);            // false
Object.is(-0, -0);           // true
Object.is(NaN, 0/0);         // true
```

Polyfill

```
if (!Object.is) {
  Object.is = function(x, y) {
    // SameValue algorithm
    if (x === y) { // Steps 1-5, 7-10
      // Steps 6.b-6.e: +0 != -0
      return x !== 0 || 1 / x === 1 / y;
    } else {
      // Step 6.a: NaN == NaN
      return x !== x && y !== y;
    }
  };
}
```

规范

Specification	Status	Comment
ECMAScript 2015 (6th Edition, ECMA-262) Object.is	ST Standard	Initial definition.
ECMAScript Latest Draft (ECMA-262) Object.is	LS Living Standard	

浏览器兼容

Desktop Mobile

Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
Basic Support	30	(Yes)	22	No	(Yes)	9

相关链接

- JavaScript 中的相等性判断 — JavaScript 中的三种相等性判断方法的比较