## 每日一题 - 数值+0和-0的区分

### 信息卡片

* 时间：2019-08-13
* tag：Number

### 题目描述

JavaScript的数值Number用64位的浮点数表示，首位是符号位，然后是52位的整数位和11位的小数位。如果符号位为1，其他各位均为0，那么这个数值会被表示成“-0”。

所以JavaScript的“0”值有两个，+0和-0。

那么如何区分呢？

### 解题思路

我们看到+0和-0，大概想尝试把该数字通过toString()转化成字符串，在使用indexOf('-')判断是否等于0，或者charAt(0)判断是否等于-。很不幸，数值在进行toString()的时候就自动将其转为0了，所以此方法行不通。

我们可以尝试另外一个思路，**计算机在进行四则及与或模等数值运算时，符号本身也参与运算**，JavaScript亦是如此。而使用0对一个数做加减操作对本身是无影响的，乘法虽然得到±0的结果，但是又回到了问题本身对±0的判断了，因此我们可以考虑到除法，加上数值本身有Infinity和-Infinity的区分，分别表示正无穷和负无穷。我们很容易想到使用一个数值来除以±0等到±Infinity。我们使用-1/0或1/-0都得到-Infinity的结果。

同样的，JavaScript提供很多函数供你使用，但结果不外乎都是借助一个数值进行判断。如：Math.pow(-0, -1) === -Infinity，Math.atan2(-0, -1) === -Math.PI

### 参考代码

// check the zero is negtive or not, only +0 and -0 can be check  
function isNegtiveZero(value) {  
 if (value !== 0) {  
 throw new RangeError("The argument must be +0 or -0");  
 }  
 return 1 / value === -Infinity;  
}  
  
console.log(isNegtiveZero(+0) === false);  
console.log(isNegtiveZero(-0) === true);

ECMAScript2015添加了一个方法Object.is用于对两数值进行比较，可以用于比较±0

Object.is(+0, 0) === true;  
Object.is(-0, 0) === false;