**7 kyu**

**Playing with Sets : Equal or Not ?**

2496% of2556[ZozoFouchtra](https://www.codewars.com/users/ZozoFouchtra)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/playing-with-sets-equal-or-not/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/playing-with-sets-equal-or-not/solutions/javascript)

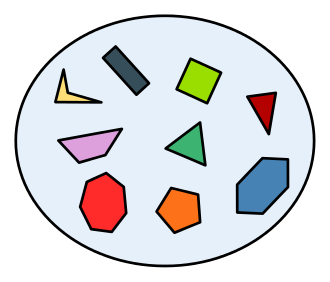
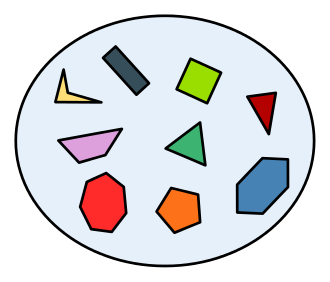
[Discourse](https://www.codewars.com/kata/playing-with-sets-equal-or-not/discuss/javascript)

* Add to Collection
* |
* Share this kata:

[Set](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set) objects are new JavaScript built-in objects defined since [ECMAScript 2015](http://www.ecma-international.org/ecma-262/6.0/" \l "sec-set-objects." \t "_blank)

A **Set** lets you store unique values of any type. It comes with some useful methods like .add, .clear, .has . . . **BUT**some "Set operations" are missing, like . . .

**Equality**



Two sets ( A, B ) are considered "equal" if **all** elements of A are elements of B and **all** elements of B are elements of A no matter "order" of elements.

**Examples:**

{1, 2} == {2, 1}

{1, 2} != {1, 2, 3}

**Task**

Create 2 functions, areEqual and notEqual, getting 2 sets as arguments and returning a true or falsedepending if these 2 sets are "equal" (respectively **not** equal) *ie : if all elements of 1st set are elements of 2d and all elements of 2d set are elements of 1st*.

**Examples:**

A = new Set([1,2]);

A2 = new Set([2,1]);

B = new Set([2,3]);

areEqual(A,B) // -> false

areEqual(A,A2) // -> true

notEqual(A,B) // -> true

" May the Code be with you ! "

<https://www.codewars.com/kata/playing-with-sets-equal-or-not/javascript>

1. **function** areEqual(s1, s2){
3. **for**(**const** item of s1) {
4. **if**(!s2.has(item)) {
5. **return** **false**;
6. }
7. }
8. **return** **true**;
9. }
11. **function** notEqual(s1, s2){
12. *// ...*
13. **return** !areEqual(s1, s2);
15. }