**7 kyu**

**Playing with Sets : Sup/Sub**

3297% of3162[ZozoFouchtra](https://www.codewars.com/users/ZozoFouchtra)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/playing-with-sets-sup-slash-sub/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/playing-with-sets-sup-slash-sub/solutions/javascript)

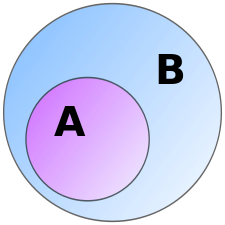
[Discourse](https://www.codewars.com/kata/playing-with-sets-sup-slash-sub/discuss/javascript)

* 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 . . .

**Subset and Superset**



If every member of set A is also a member of set B, then A is said to be a subset of B, written A ⊆ B (also pronounced "A *is contained in* B"). Equivalently, we can write B ⊇ A, read as "B *is a superset of* A", "B *includes* A", or "B *contains* A".

**Example:**

{1, 3} ⊆ {1, 2, 3, 4}.

{1, 2, 3, 4} ⊆ {1, 2, 3, 4}.

{1, 2, 3, 4} ⊇ {1, 3}.

{1, 2, 3, 4} ⊇ {1, 2, 3, 4}.

**Task**

Create 2 functions:

* isSubsetOf getting 2 sets as arguments and returning true if 2d set contains **all elements** of 1st one.
* isSupersetOf getting 2 sets as arguments and returning true if 1st set contains **all elements** of 2d one.

**Examples:**

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

B = new Set([1,2,3]);

isSubsetOf(A,B) // -> true

isSubsetOf(B,A) // -> false

isSupersetOf(A,B) // -> false

isSupersetOf(B,A) // -> true

" May the Code be with you ! "

<https://www.codewars.com/kata/playing-with-sets-sup-slash-sub/javascript>

1. **function** isSubsetOf(s1, s2){
3. **for**(**const** item of s1) {
4. **if**(!s2.has(item)) {
5. **return** **false**;
6. }
7. }
8. **return** **true**;
9. }
11. **function** isSupersetOf(s1, s2){
12. **for**(**const** item of s2) {
13. **if**(!s1.has(item)) {
14. **return** **false**;
15. }
16. }
17. **return** **true**;
19. }