**7 kyu**

**Playing with Sets : Complement**

3394% of2768[ZozoFouchtra](https://www.codewars.com/users/ZozoFouchtra)

JavaScript

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

Details

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

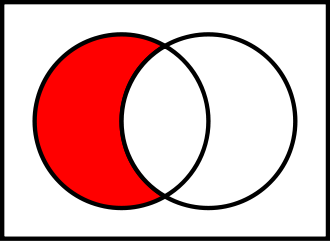
[Discourse](https://www.codewars.com/kata/playing-with-sets-complement/discuss/javascript)

* 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 . . .

**Complement**



Two sets can be "subtracted". The **relative complement** of B in A, denoted by A \ B (or A − B), is the set of all elements that are members of A but not members of B. Note that it is valid to "subtract" members of a set that are not in the set, such as removing the element green from the set {1, 2, 3}; doing so has no effect.

**Examples:**

{1, 2} \ {1, 2} = ∅.

{1, 2, 3, 4} \ {1, 3} = {2, 4}.

**Task**

Create function diff getting 2 sets as arguments and returning a **new Set** as result of relative complement of second set in first.

**Examples:**

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

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

diff(A,B) // -> {1}

diff(B,A) // -> {3}

" May the Code be with you ! "

<https://www.codewars.com/kata/playing-with-sets-complement/javascript>

1. **function** union(s1, s2){
2. *//...*
3. **const** s = **new** Set();
5. **for**(**const** item of s1) {
6. s.add(item);
7. }
8. **for**(**const** item of s2) {
9. s.add(item);
10. }
11. **return** s;
12. }
14. **function** diff(s1, s2){
16. **const** dif = **new** Set();
18. **for**(**const** item of s1) {
19. **if**(!s2.has(item)) {
20. dif.add(item);
21. }
22. }
23. **return** dif;
24. }

27. **var** res = union( **new** Set([2,2,2,2,2,2,2,2]), **new** Set( [2,2,2,2,2,2])  );


31. **for** (let item **in** res.values()) {
32. print(item + " ");
33. }