

Array diff

<https://www.codewars.com/kata/523f5d21c841566fde000009/python>

```
def array_diff(a, b):  
    stack = []  
  
    for i in a:  
        for x in b:  
            if i != x:  
                stack.append(i)  
    return stack
```

This was my initial code.

The code would iterate once as the return stack was not properly nested, for x in b was unnecessary as you don't really need another variable to iterate through list b

Your current implementation has a logical flaw. It appends the element `i` to the `stack` for every element `x` in `b` that is not equal to `i`. This results in duplicates being added to the `stack` for each comparison with the elements in `b`.

```
def array_diff(a, b):  
    stack = []  
  
    for i in a:  
        if i not in b:  
            stack.append(i)  
    return stack
```

This was the reworked code

single loop and membership test, list comprehension

```
def array_diff(a, b):  
    # Create a set from list b for O(1) average-time complexity  
    b_set = set(b)  
  
    # Use list comprehension to filter out elements in a that are in b  
    result = [item for item in a if item not in b_set]  
  
    return result
```