

Forum:[Hack Forums](#)

Thread: [Simple Python Question](#)

?Solace - September 25, 2014, 12:02 pm

Hello HF,

I have a simple question about a simple operation in python.

I'm attempting to pass a function a list (variable a), and exclude all numbers that are less than my second variable which is given by the user ®.

```
Code def threshold(a,x):  
    for i, val in enumerate(a):  
        if a[i] < x:  
            a.remove(a[i]);  
    return a;
```

```
r = int(raw_input("Enter a number to take the threshold of:"));  
print threshold([0,1,2,3,4,5,6],r);
```

Could anybody explain to me why my code doesn't work? (I don't have any formal learning with python, I'm taking Java questions from an old course and replicating them to teach myself.)

Thanks :)

tyguy - September 25, 2014, 12:11 pm

could be youre returning a table or some shit idk

K-On! - September 25, 2014, 12:25 pm

Use `del a[i]` instead of `a.remove(a[i])`

`.remove()` removes an item based on the value you give it, not by the index.

MoonCakeRuler - September 25, 2014, 12:42 pm

(Sep 25, 2014 12:25 PM)K-On! Wrote: [\[CITE\]](#) Use `del a[i]` instead of `a.remove(a[i])`

`.remove()` removes an item based on the value you give it, not by the index.

He passed `a[i]` to `remove()`, so it still counts as a value. D:

To the OP, your code doesn't work because you're enumerating and removing items from the list at the same time. If we take an example argument of a list `[0,1,2,3,4]` and the second argument as `2`, your iterator condition removes `0` from the list. BUT, then your list shifts by one position and becomes `[1,2,3,4]`. However, the iterator also increases the index and so, you don't get what you need as the `1` is now in `0`th position while the index is shifted to `1`.

A better way to do this is a list comprehension:

```
Code return [i for i in a if i >= x]
```