

Forum:[Hack Forums](#)

Thread: [Reverse array challenge code I wrote](#)

Doc - November 23, 2020, 7:38 am

Posting this here looking for constructive criticism. If anyone's got any tips/advice on things I could've done better/differently, please feel free to say so.

The code challenge was: Prompt: Reverse an array. A classic code interview question.

Difficulty: easy

Languages: no-restrictions

Constraints: The function should take in any array and return that array in reverse. If your language has a native `.reverse()` function cannot use it because this is a test of complexity and not how well you can use a function!

Bonus: Do this without allocating any extra memory. This is called modifying the array "in-place"!

Code # reverse an array challenge

```
# gotta have numpy
import numpy as np

# big daddy string to get things done
def string_to_array_to_reversed_array():

    # for handling user input
    def user_input_function(user_input): # user_input to take in the input when called
        user_input = list(user_input.replace(" ", "")) # user_input is reassigned it's input string as a list,
    while user_input.replace trims any white space out
        return np.array(user_input, order = 'C') # return user_input list as an array in contiguous order
    to array_arg parameter in reverse_array function

    # for reversing the array
    def reverse_array(array_arg): # array_arg takes in the return from user_input_function
        return np.flipud(array_arg) # FLIPS THE ARRAY, DOESN'T REVERSE IT. YOU SEE THE
    .flipud part right?. NOTHING WAS SAID ABOUT FLIPPING. ONLY REVERSING.

    # x assigned to take in final return value and store it for printing
    x = reverse_array(user_input_function(input("Please type in a single word as a string: "))) # get
    user input
    return x # RESULTS ARE IN!

# call "main" function and make shit happen, store in reversed_array for printing.
reversed_array = string_to_array_to_reversed_array()
print(reversed_array)
I dunno if I got the bonus, although I DID get a lotta much needed experience using numpy for
```

certain things, so at least there's that.
Anyway lemme know what you guys think.

Qismon - December 1, 2020, 8:07 am

```
Code def reverse_array(normalArray):  
    reversedArray = normalArray[::-1]  
    return reversedArray
```

```
normalArray = [1, 2, 3, 4, 5]  
reversedArray = reverse_array(normalArray)  
print(reversedArray)
```

vortalbox - December 1, 2020, 8:17 am

(Dec 1, 2020 09:07 AM)Qismon Wrote: [\[CITE\]](#) Code def reverse_array(normalArray):
 reversedArray = normalArray[::-1]
 return reversedArray

```
normalArray = [1, 2, 3, 4, 5]  
reversedArray = reverse_array(normalArray)  
print(reversedArray)
```

I would say using python's list functions would fall under python's built-in reverse.

SecuroServ - December 1, 2020, 8:27 am

I think what they are looking for is, loop over the array, switch first item with last item, second item with second last item etc until you reach the middle.

I know using a library isn't in their constraints, but I'm pretty sure they also mean that your thing isn't allowed because they are looking for complexity like they said

vortalbox - December 1, 2020, 12:32 pm

(Dec 1, 2020 09:27 AM)SecuroServ Wrote: [\[CITE\]](#) I think what they are looking for is, loop over the array, switch first item with last item, second item with second last item etc until you reach the middle.

I know using a library isn't in their constraints, but I'm pretty sure they also mean that your thing isn't allowed because they are looking for complexity like they said

Completely agree. No packages allowed.

Here is my solution

PHP Code

```
<?php  
def reverse($lst):  
    $length = count($lst)  
    $tmp = array();  
    for ($i = 0; $i < $length; $i++):  
        $idx = ($length - 1) - $i;  
        $tmp[$i] = $lst[$idx];  
    return $tmp;
```

LEGION - December 2, 2020, 6:14 am

This is the easiest solution I could think of. Just looping through the original array backwards and adding all the items to a new array.

PHP Code

```
<?php
def reverse(arr):
    rev = []

    for x in range(len(arr), 0, -1):
        rev.append(x)

    return rev
```

vorticalbox - December 2, 2020, 9:16 pm

(Dec 2, 2020 07:14 AM)LEGION Wrote: [\[CITE\]](#) This is the easiest solution I could think of. Just looping through the original array backwards and adding all the items to a new array. Should it not be this?

PHP Code

```
<?php
def reverse(arr):
    rev = []

    for x in range(len(arr), 0, -1):
        rev.append(arr[x])

    return rev
```

Doc - December 8, 2020, 12:39 pm

Honestly it was kinda confusing for me to figure out how to go about doing this, primarily because Python doesn't technically have arrays, it has this weird kind of "lists" (I actually saw a pretty good meme about that recently) that can be a bit confusing to work with. Trying to reverse an array "in-place" (which was what I was originally wanting to do) is something I'd think is easier and more straightforward when using languages that have "normal" arrays.

synchronizing - December 9, 2020, 11:23 am

This will reverse even and odd lists with almost zero memory space used, and quicker than most other solutions since it only iterates to half of the length of the list (though same big O).

PHP Code

```
<?php
def rev(lst):
    for i in range(len(lst) // 2):
        lst[i], lst[len(lst) - i - 1] = lst[len(lst) - i - 1], lst[i]

    return lst

print(rev([1, 2, 3, 4, 5, 6, 7])) # [7, 6, 5, 4, 3, 2, 1]
print(rev([1, 2, 3, 4, 5, 6, 7, 8])) # [8, 7, 6, 5, 4, 3, 2, 1]
```

