

Write a function to reverse a string in-place ↴ .

Since strings in Python are immutable ↴ , first convert the string into *a list of characters*, do the in-place reversal on that list, and re-join that list into a string before returning it. This isn't technically "in-place" and the list of characters will cost $O(n)$ additional space, but it's a reasonable way to stay within the spirit of the challenge. If you're comfortable coding in a language with mutable strings, that'd be even better!

Breakdown

In general, an "in-place" ↴ algorithm will require swapping elements.

Solution

We swap the first and last characters, then the second and second-to-last characters, and so on until we reach the middle.

```
def reverse(string):  
    string_list = list(string)  
  
    left_pointer = 0  
    right_pointer = len(string_list) - 1  
  
    while left_pointer < right_pointer:  
  
        # swap characters  
        string_list[left_pointer], string_list[right_pointer] = \  
            string_list[right_pointer], string_list[left_pointer]  
  
        # move towards middle  
        left_pointer += 1  
        right_pointer -= 1  
  
    return ''.join(string_list)
```

Complexity

$O(n)$ time and $O(1)$ space.

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.