

Python Programming

Sorting and Reversing Methods

Mostafa S. Ibrahim

Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada

Bachelor / Msc from Cairo University - Egypt

Ex-(Software Engineer / ICPC World Finalist)



Sort Method

- Algorithm: is a step-by-step procedure for calculations
 - We already tried several algorithmic (computational) problems in loops section
- Sort algorithm: Order the items. By default from small to large
- **In-place algorithm:** It doesn't create new memory. It modifies the given one
 - Minor memory creation may occur

```
1
2 lst = [5, 7, 2]
3 # NO new list: in-place - memory efficient
4 lst.sort() ..... # [2 5 7]
5
6 lst.sort(reverse=True) ..... # [7 5 2]
7
8 # common mistake:
9 lst = lst.sort()
10 # lst now is NONE!
11
```

Sorted function

- Returns a sorted list of the specified **iterable** object.
 - Work on list and others

```
12  lst = [5, 7, 2]
13  lst_sorted_cpy = sorted(lst) ... # sorted copy
14  # lst = NO CHANGE
15  # lst_sorted_cpy [2 5 7]
16
17  my_str = 'zacb'
18  new_lst = sorted(my_str) ... # LIST! ['a', 'b', 'c', 'z']
19  new_lst = sorted(my_str, reverse=True)
20  # new_lst = ['z', 'c', 'b', 'a']
21
22  print(new_lst)
23  # common mistake
24  sorted = sorted(my_str)
25  # now sorted become a variable. You can't call the function
26  # TypeError: 'list' object is not callable
27  #sorted = sorted(my_str)
```

Reverse Method Reversed Function

```
1
2 my_list = [1, 2, 3, 4]
3
4 my_list.reverse() ... # 4 3 2 1
5
6 my_list += ['Hi']
7
8 new_lst = reversed(my_list)
9 print(new_lst) ... # list_reverseiterator
10
11 new_lst_rev1 = list(reversed(my_list))
12 print(new_lst_rev1) ... # list_reverseiterator
13 # ['Hi', 1, 2, 3, 4]
14
15 new_lst_rev2 = my_list.copy()
16 new_lst_rev2.reverse()
17 print(new_lst_rev2)
18 # ['Hi', 1, 2, 3, 4]
19
```

Iterate in a reversed order

```
2 lst = [7, 8, 9, 'Hi']
3
4 for pos in range(len(lst)): # C++/Java style
5     print(lst[len(lst) - pos - 1], end=' ')
6 print()
7
8 # Python: range(start, end, step)
9 for idx in range(len(lst) - 1, -1, -1):
10     print(lst[idx], end=' ')
11 print()
12
13 # Better
14 for item in reversed(lst): # NO copy is created
15     print(item, end=' ')
16 print()
17
18 for pos, item in reversed(list(enumerate(lst))):
19     print(pos, item, end=' - ')
20 # 3 Hi - 2 9 - 1 8 - 0 7
21
22 # be careful:
23 # list(iterable) => makes copy: more memory / slower
24
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”