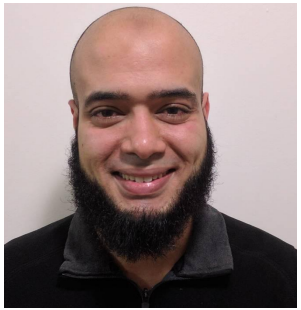
Python Programming

Slicing 2

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)



Slice with a **positive** step

```
4   my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
5
6   sub_list = my_list[1:8 ]    # 1 2 3 4 5 6 7
7   sub_list = my_list[1:8:1]   # 1 2 3 4 5 6 7
8   sub_list = my_list[1:8:2]   # 1 3 5 7
9   sub_list = my_list[1:8:3]   # 1 4 7
10
11  # Missing step: default = 1
12  sub_list = my_list[1:8: ]   # [1, 2, 3, 4, 5, 6, 7]
13
14  # Positive step: Missing end: default is len(seq)
15  sub_list = my_list[1:9:2]   # 1 3 5 7
16  sub_list = my_list[1: :2]   # 1 3 5 7
17
18  # Positive step: Missing start: default is 0
19  sub_list = my_list[0:6:2]   # 0 2 4
20  sub_list = my_list[ :6:2]   # 0 2 4
21
22  sub_list = my_list[0:9:2]   # 0 2 4 6 8
23  sub_list = my_list[ : :2]   # 0 2 4 6 8
24
25  sub_list = my_list[0:9:1]   # [0, 1, 2, 3, 4, 5, 6, 7, 8]
26  sub_list = my_list[ : : ]   # [0, 1, 2, 3, 4, 5, 6, 7, 8]
27
```

Slice with a **negative** step

```
my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]

sub_list = my_list[1:8:1]    # 1 2 3 4 5 6 7

sub_list = my_list[8:1:-1]   # 8 7 6 5 4 3 2: high to low

sub_list = my_list[7:0:-1]   # 7 6 5 4 3 2 1
sub_list = my_list[7:0:-2]   # [7, 5, 3, 1]

sub_list = my_list[2:5:-1]   # [] must be high to low

# Negative step: Missing start: default is len
sub_list = my_list[9:2:-1]       # [8, 7, 6, 5, 4, 3]
sub_list = my_list[ :2:-1]       # [8, 7, 6, 5, 4, 3]

# Negative step: Missing end: default is hmm
# starts from index 0 INCLUSIVE (NOT default)
sub_list = my_list[5: :-1]       # [5, 4, 3, 2, 1, 0]

sub_list = my_list[5:0:-1]       # [5, 4, 3, 2, 1]

sub_list = my_list[::-1]         # reversed list
# [8, 7, 6, 5, 4, 3, 2, 1, 0]

```

Missing defaults - In other words

```
3   my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
4
5   # Positive step: Missing start: iterate from the begin
6   print(my_list[ :5:1])       # [0, 1, 2, 3, 4]
7
8   # Positive step: Missing end: iterate till the end (inclusive)
9   print(my_list[2: :1])       # [2, 3, 4, 5, 6, 7, 8]
10
11  # Negative step: Missing start: iterate from the end
12  print(my_list[ :5:-1])      # [8, 7, 6]
13
14  # Negative end: Missing start: iterate till the begin (inclusive)
15  print(my_list[2: :-1])      # [2, 1, 0]
16
17  # covers from the end till the begin inclusive
18  print(my_list[ : :-1])      # [8, 7, 6, 5, 4, 3, 2, 1, 0]
19
20  # kind of: cover all values in the missing direction
21
22  # practice makes perfect :)
```

Replace and Delete

```
2
3     lst = [1, 2, 3, 4, 5, 6, 7]
4     lst[2] = 100    # 1 2 100 4 5 6 7
5
6     lst[3:6] = [982]    # 1 2 100 982 7
7
8     lst[1:3] = [10, 11, 12, 13]    # 1 10 11 12 13 982 7
9
10    # you need to replace 3 times with LIST OF THREE
11    #lst[1:6:2] = [1]   # ValueError
12    lst[1:6:2] = [-1, -2, -3]    # 1 -1 11 -2 13 -3 7
13    #lst[6:2:-2] = [0]   # ValueError
14
15    lst[3:] = [123]    # 1 -1 11 123
16
17    lst = [1, 2, 3, 4, 5, 6, 7]
18
19    del lst[1:3]          # 1 4 5 6 7
20
21    del lst[1:5:2]       # 1 5 7
22
23
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."