

# Python Programming

## Slicing 1

**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)*

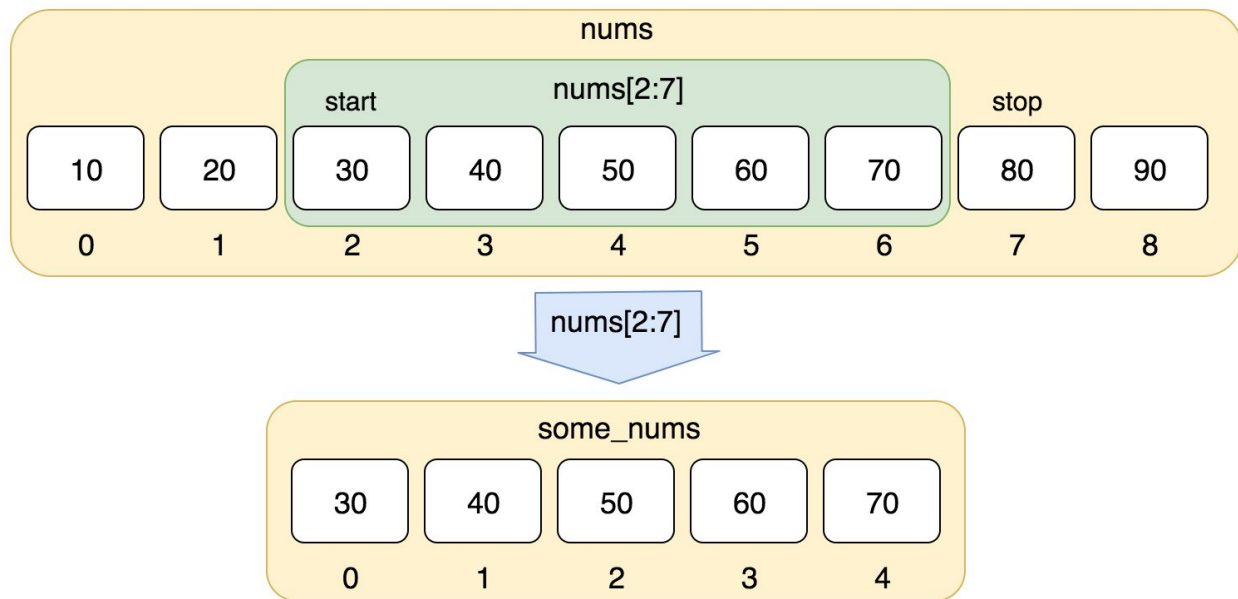


# Recall: Range Function

```
2  # [First index to include, first index to exclude : step]
3  # step can be +ve or -ve
4  # range can be increasing or decreasing based on step
5
6  print(type(range(5)))
7
8  print(list(range(5))) ..... # [0, 1, 2, 3, 4]
9
10 print(list(range(2, 5))) ..... # [2, 3, 4]
11
12 print(list(range(1, 21, 4))) ..... # [1, 5, 9, 13, 17]
13
14 print(list(range(5, 0, -1))) ..... # [5, 4, 3, 2, 1]
15
16 print(list(range(10, 0, -2))) ..... # [10, 8, 6, 4, 2]
17
18 print(list(range(5-1, -1, -1))) ..... # [4, 3, 2, 1, 0]
```

# Slicing

- Slicing is a flexible tool to build new lists out of an existing list.



Slicing: <first index to **include**, first index to **exclude**>

```
3 my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
4
5 # 2 is the start
6 # 6 is end (exclusive): ends actually at 5
7 sub_list = my_list[2:6] ... # 2 3 4 5
8
9 sub_list[0] = 100 ... # my_list is NOT changed
10
11 sub_list = my_list[5:6] ... # 5 a single element
12
13 sub_list = my_list[5:1000] ... # 5 6 7 8
14
15 # syntax: my_list[start : end+1
16
```

# Default limits

```
2 my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
3
4 sub_list = my_list[0:5] # 0 1 2 3 4
5 # If you did not provide start: then 0
6 sub_list = my_list[:5] # 0 1 2 3 4
7
8 # 9 = len(my_list)
9 sub_list = my_list[4:9] # 4 5 6 7 8
10 # similarly: if not end: it is len
11 sub_list = my_list[4:] # 4 5 6 7 8
12
13 # observe:
14 # my_list[4] is the 5th element (index 4)
15 # my_list[4:] is slice from index 4 to last element
16 # my_list[:4] is slice from 0 to 3
17
18 same_values = my_list[:4] + my_list[4:]
19 # 0 1 2 3 4 5 6 7 8
20 print(same_values is my_list) # False
21
22 # both start and end are empty: WHOLE list
23 same_values = my_list[:]
24
25 print(same_values)
```

# 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
```

- Practice well before next time
- Next
  - Missing values for (start, end, step)
  - Negative step
  - Replace and delete

# Slicing and Memory

```
2
3 class Employee:
4     pass
5
6 obj1 = Employee()
7 obj2 = Employee()
8 obj3 = Employee()
9
10 lst1 = [obj1, obj2, obj3]
11 lst2 = lst1[0:2] ... # create a NEW list
12
13 print(lst1 is lst2) ... # False
14 print(lst1[0] is lst2[0]) ... # True
15
16 # List is new - items are just assigned (same memory)
17
18
```

*“Acquire knowledge and impart it to the people.”*

*“Seek knowledge from the Cradle to the Grave.”*