Data Structures of Python

x=5

lst1=[6,7,'8.9','Python','Prgramming',3.6]

print(lst1)   # [6, 7, '8.9', 'Python', 'Prgramming', 3.6]

* Data structures are used to store a collection of data
* There are 4 builtin data structures

List 🡪 []

Tuple🡪()

Dictonary-🡪{} with key value pair  
Set 🡪 {}

LISTS:

my\_list=[]

my\_list1=[3,2,4,"python",5.6]

print(bool(my\_list))  # False

print(bool(my\_list1)) # True

“hello”-🡪 0 1 2 3 4

-5 -4 -3 -2 -1

my\_list=[3,2.4,"python",5.6]

my\_list=[3,2.4,"python",5.6]

print(my\_list[0]) #3

print(my\_list[1]) #2.4

print(type(my\_list)) # <class 'list'>

my\_list=[3,2,4,"python",5]

print(my\_list[2],my\_list[-3])  # 4  4

print(my\_list[3][1])  # y

my\_list=[3,2.4,"python",5.6]

print(my\_list)        #[3, 2.4, 'python', 5.6]

print(my\_list[:])      # [3, 2.4, 'python', 5.6]

print(my\_list[0:])     # [3, 2.4, 'python', 5.6]

print(my\_list[1:4])   # [2.4, 'python', 5.6]

print(my\_list[:4])    # [3, 2.4, 'python', 5.6]

####################################

Lists are mutable

Strings are immutable

# my\_list=[3,4,5,7,8,5,9,10]

# print(dir(my\_list))

# 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']

my\_list.index(5) 🡪 2

(5,2) -🡪 6 (7th value)

Count(10) -🡪

Count(5) -🡪 3

-🡪 clear() (operation) no taking output for this

🡪 print(my\_list)

* Count\_5=my\_list.cont(5)

My\_new\_list=my\_list

My\_one\_list=my\_list.copy()

Id(my\_new\_list,my\_list) -🡪 Are same

Id(my\_one\_list,my\_list) -🡪 are different

Append:

My\_list

my\_list.append(56)🡪 Last

3, ---- Insert 5,

my\_list.insert(1,45) ----------🡪 3,1,45

4. Extend

My\_list=[2,5,6,6,5,7,8]

My\_new\_list=[5,6]

My\_list.append(my\_new\_list)

[2,5,6,6,5,7,8,[5,6]]

Extend:

My\_list.extend(my\_new\_list

[2,5,6,6,5,7,8,5,6]

Pop/remove:

My\_list=[5,6,7,8,4,3,2]

My\_list.remove(10)

Error: Not in the list

My\_list.remove(4) 🡪 5,6,7,8,3,2]

my\_list.pop() 🡪 [5,6,7,8,3] 🡪Modifying the data

my\_list.pop(0) 🡪 [6,7,8,3---🡪 Last Index

5,6,7,8,3

My\_list.pop(0) 🡪 Index 0

5,6,7,8,3,2

print(my\_list.pop ()) 🡪2

print(my\_list) -🡪 5,6,7,8,3

Reverse/Sort:

5,6,7,8,3,2

My\_list.reverse()

Print(my\_list)

My.list.sort()🡪 Ascneding order

My\_list.reverse() 🡪 Descending order after

My\_list.sort(reverse=true) 🡪 Above the same.