Basic Data Structure In Python

1- Tuple

2- List

3- Dict

4- Set

Tuple

- ordered collection of elements

- enclosed with round braces ()

- Tuples are immutable means ones the values are entered, you cannot change the values

In [1]:
```
#Tuples can store different data types

#Basic Example
tup1 = (1, "Python", True, 2.5,1, "Python", True, 2.5)
tup1
```

Out[1]:
```
(1, 'Python', True, 2.5, 1, 'Python', True, 2.5)
```

In [2]:
```
#Advance Example
tup2 = (1, "Python", True, 2.5,("Python", True, 2.5), ["Python", True, 2.5], {"name": "
tup2
```

Out[2]:
```
(1,
 'Python',
 True,
 2.5,
 ('Python', True, 2.5),
 ['Python', True, 2.5],
 {'name': 'Mustafa'})
```

In [3]:
```
tup1[1]
```

Out[3]:
```
'Python'
```

In [4]:
```
tup2[1:6]
```

Out[4]:
```
('Python', True, 2.5, ('Python', True, 2.5), ['Python', True, 2.5])
```

In [5]:
```
len(tup2)
```

Out[5]:
```
7
```

In [6]:
```
#Concatination
tup1 + tup2
```

Out[6]:
```
(1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 ('Python', True, 2.5),
 ['Python', True, 2.5],
 {'name': 'Mustafa'})
```

In [7]:
```
#Repetition
tup1*3
```

Out[7]:
```
(1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5,
 1,
 'Python',
 True,
 2.5)
```

In [8]:
```
tup3 = (10,20,30,40,50,60)
tup3
```

Out[8]:
```
(10, 20, 30, 40, 50, 60)
```

In [9]:
```
min(tup3)
```

Out[9]:
```
10
```

In [10]:
```
max(tup3)
```

Out[10]:
```
60
```

In [11]:
```
type(tup2)
```

Out[11]:
```
tuple
```

In [12]:
```
tup1
```

Out[12]:
```
(1, 'Python', True, 2.5, 1, 'Python', True, 2.5)
```

In [13]:
```
tup1.index('Python')
```

Out[13]:
```
1
```

In [14]:
```
#Count method will return the count how many time the given value appares in the dataTy
tup1.count(2.5)
```

Out[14]:
```
2
```

List

- ordered collection of elements

- enclosed with square braces []

- List are mutable means ones the values are entered, you can change the values

In [15]:
```
#Simple Example
list1 = [1, 'Python', 2.5, "Mustafa",False, 66]
list1
```

Out[15]:
```
[1, 'Python', 2.5, 'Mustafa', False, 66]
```

In [16]:
```
#Advance Example
adList = [1,"Mustafa", True, 9.0, (True, 33, 'Muhammad'),[True, 33, 'Muhammad'],{'name'
adList
```

Out[16]:
```
[1,
 'Mustafa',
 True,
 9.0,
 (True, 33, 'Muhammad'),
```

```
 [True, 33, 'Muhammad'],
 {'name': 'Mustafa'}]
```

In [17]:
```
list2 = [55, True, 33, 'Muhammad']
list2
```

Out[17]:
```
[55, True, 33, 'Muhammad']
```

In [18]:
```
type(list1)
```

Out[18]:
```
list
```

In [19]:
```
len(list1)
```

Out[19]:
```
6
```

In [20]:
```
list1[2]
```

Out[20]:
```
2.5
```

In [21]:
```
#Concatination
list1 + list2
```

Out[21]:
```
[1, 'Python', 2.5, 'Mustafa', False, 66, 55, True, 33, 'Muhammad']
```

In [22]:
```
list1 * 2
```

Out[22]:
```
[1,
 'Python',
 2.5,
 'Mustafa',
 False,
 66,
 1,
 'Python',
 2.5,
 'Mustafa',
 False,
 66]
```

In [23]:
```
list1.reverse()
list1
```

Out[23]:
```
[66, False, 'Mustafa', 2.5, 'Python', 1]
```

In [24]:
```
list1.append("Hello")
list1
```

Out[24]:
```
[66, False, 'Mustafa', 2.5, 'Python', 1, 'Hello']
```

In [25]:
```
list1.count("Hello")
```

Out[25]:
```
1
```

In [26]:
```
x = list1.copy()
x
```

Out[26]:
```
[66, False, 'Mustafa', 2.5, 'Python', 1, 'Hello']
```

In [27]:
```
list1.clear()
list1
```

Out[27]:
```
[]
```

In [28]:
```
list1 = x.copy()
list1
```

Out[28]:
```
[66, False, 'Mustafa', 2.5, 'Python', 1, 'Hello']
```

In [29]:
```
list1.extend(x)
list1
```

Out[29]:
```
[66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello',
 66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello']
```

In [30]:
```
list1.count("Hello")
```

Out[30]:
```
2
```

In [31]:
```
list1.index('Mustafa')
```

Out[31]:
```
2
```

In [32]:
```
list1.insert(0,'newVal')
list1
```

Out[32]:
```
['newVal',
 66,
```

```
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello',
 66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello']
```

In [33]:
```
list1.pop()
list1
```

Out[33]:
```
['newVal',
 66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello',
 66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1]
```

In [34]:
```
list1.remove('newVal')
list1
```

Out[34]:
```
[66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1,
 'Hello',
 66,
 False,
 'Mustafa',
 2.5,
 'Python',
 1]
```

In [35]:
```
numList = [22,331,1,32,44,12,45,1,2,4,66,4,111,53,25,6,4,2,67,32,2,6,3,4,2165,45]
numList
```

Out[35]:
```
[22,
 331,
 1,
 32,
 44,
 12,
```

```
 45,
 1,
 2,
 4,
 66,
 4,
 111,
 53,
 25,
 6,
 4,
 2,
 67,
 32,
 2,
 6,
 3,
 4,
 2165,
 45]
```

In [36]:
```
numList.sort()
numList
```

Out[36]:
```
[1,
 1,
 2,
 2,
 2,
 3,
 4,
 4,
 4,
 4,
 6,
 6,
 12,
 22,
 25,
 32,
 32,
 44,
 45,
 45,
 53,
 66,
 67,
 111,
 331,
 2165]
```

Dict

- unordered collection of elements

- enclosed with curly braces {}

- Dict are mutable means ones the values are entered, you can change the values

- Dict are consist of key value pairs

In [37]:
```
dict1 = {"Name": "Mustafa", 'car': 'bmw'}
dict1
```

Out[37]:
```
{'Name': 'Mustafa', 'car': 'bmw'}
```

In [38]:
```
#advance Dict
dict2 = {"Fruits_Prices":{'apple': 20, 'banana': 20}, "List_Of_Shops":[22,3,4,5,3,2,2]}
dict2
```

Out[38]:
```
{'Fruits_Prices': {'apple': 20, 'banana': 20},
 'List_Of_Shops': [22, 3, 4, 5, 3, 2, 2]}
```

In [39]:
```
y =  dict1.copy()
y
```

Out[39]:
```
{'Name': 'Mustafa', 'car': 'bmw'}
```

In [40]:
```
dict1.clear()
dict1
```

Out[40]:
```
{}
```

In [41]:
```
dict1 = y.copy()
dict1
```

Out[41]:
```
{'Name': 'Mustafa', 'car': 'bmw'}
```

In [42]:
```
dict1.get('Name')
```

Out[42]:
```
'Mustafa'
```

In [43]:
```
dict1.items()
```

Out[43]:
```
dict_items([('Name', 'Mustafa'), ('car', 'bmw')])
```

In [44]:
```
dict1.keys()
```

Out[44]:
```
dict_keys(['Name', 'car'])
```

In [45]:
```
dict1.pop('car')
dict1
```

Out[45]:
```
{'Name': 'Mustafa'}
```

In [46]:
```
dict1 = y.copy()
```

In [47]:
```
dict1.popitem()
```

Out[47]:
```
('car', 'bmw')
```

In [48]:
```
dict1 = y.copy()
```

In [49]:
```
dict1.update({'Room': 1})
dict1
```

Out[49]:
```
{'Name': 'Mustafa', 'car': 'bmw', 'Room': 1}
```

In [50]:
```
dict1.values()
```

Out[50]:
```
dict_values(['Mustafa', 'bmw', 1])
```

In [51]:
```
dict1.setdefault("Name","Muhammad")
dict1
```

Out[51]:
```
{'Name': 'Mustafa', 'car': 'bmw', 'Room': 1}
```

Sets

- unordered collection of elements

- enclosed with curly braces {}

- unindexed

- no duplicates allowed

In [52]:
```
set1 = {11,'Muhammad', "Mustafa", 33.33}
set1
```

Out[52]:
```
{11, 33.33, 'Muhammad', 'Mustafa'}
```

In [53]:
```
set1.add('newValue')
set1
```

Out[53]:
```
{11, 33.33, 'Muhammad', 'Mustafa', 'newValue'}
```

In [54]:
```
set2 = set1.copy()
set2
```

Out[54]:
```
{11, 33.33, 'Muhammad', 'Mustafa', 'newValue'}
```

In [55]:
```
set1.clear()
set1
```

Out[55]:
```
set()
```

In [56]:
```
set1 = set2.copy()
set1
```

Out[56]:
```
{11, 33.33, 'Muhammad', 'Mustafa', 'newValue'}
```

In [57]:
```
abc_Set = {'a','b','c'}
abcd_Set = {'a','b','d'}
```

In [58]:
```
#checks the 2 same length Sets and return the difference of the set

deff_set = abc_Set.difference(abcd_Set)
deff_set
```

Out[58]:
```
{'c'}
```

In [59]:
```
print(abc_Set.difference_update(abcd_Set))
```

```
None
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: