Basic Data structure in Python

1- Tuple

2- List

3-Dictionaries

4- Set

1-Tuples

In [ ]:
```
-Ordered collection of elements
-enclosed in () round braces/paranthesis
-Different kind of elements can be stored
-Once elements are stored you can not change them (unmutateable)
```

tup1 means first method to use tup

In [7]:
```
tup1=(1,"python", True, 2.5)
tup1
```

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

In [8]:
```
# type of tuple
type(tup1)
```

Out[8]:
```
tuple
```

Indexing in tuple (Start from 0)

In [9]:
```
tup1[1]
```

Out[9]:
```
'python'
```

In [10]:
```
tup1[0]
```

Out[10]:
```
1
```

In [13]:
```
tup1[3]
```

Out[13]:
```
2.5
```

In [14]:
```
tup1[2]
```

Out[14]:
```
True
```

In [18]:
```
tup1[1],tup1[2]
```

Out[18]:
```
('python', True)
```

In [20]:
```
tup1[0:4]
```

Out[20]:
```
(1, 'python', True, 2.5)
```

In [21]:
```
tup1[0:2]
```

Out[21]:
```
(1, 'python')
```

In [22]:
```
len(tup1)
```

Out[22]:
```
4
```

tup2 means is 2nd method to use the tup

In [47]:
```
tup2 = (2, "sana", 3.4, False, True)
tup2
```

Out[47]:
```
(2, 'sana', 3.4, False, True)
```

In [75]:
```
# plus called concatenate (to add two tuple or more than two)
tup1+tup2
```

Out[75]:
```
(1, 'python', True, 2.5, 2, 'sana', 3.4, False, True)
```

In [48]:
```
# concatinate +Repition
tup1*2+tup2
```

Out[48]:
```
(1, 'python', True, 2.5, 1, 'python', True, 2.5, 2, 'sana', 3.4, False, True)
```

tup1+tup22 22 means 2 times

In [49]:
```
tup1+tup2*2  ()
```

Out[49]:
```
(1,
 'python',
 True,
 2.5,
 2,
 'sana',
 3.4,
 False,
 True,
 2,
 'sana',
 3.4,
 False,
 True)
```

*tup2*3 means 3 times repition*

In [50]:
```
tup1+tup2*3
```

Out[50]:
```
(1,
 'python',
 True,
 2.5,
 2,
 'sana',
 3.4,
 False,
 True,
 2,
 'sana',
 3.4,
 False,
 True,
 2,
 'sana',
 3.4,
 False,
 True)
```

In [62]:
```
tup3=(70,40,20,60)
tup3
```

Out[62]:
```
(70, 40, 20, 60)
```

minimum number

In [60]:
```
min(tup3)
```

Out[60]:
```
20
```

In [63]:
```
max(tup3)
```

Out[63]:
```
70
```

*tup3*4 means repition 4 times*

In [69]:
```
tup3*4
```

Out[69]:
```
(70, 40, 20, 60, 70, 40, 20, 60, 70, 40, 20, 60, 70, 40, 20, 60)
```

2-List

- Ordered collection of elements
- enclosed in these square braket/braces
- Mutateable, you can change the values

In [89]:
```
list1 = (1, "sana", 4.5, True)
list1
```

Out[89]:
```
(1, 'sana', 4.5, True)
```

In [90]:
```
type(list1)
```

Out[90]:
```
tuple
```

( ) for Tuple, [ ] for list

In [139…
```
list2=[1, "sana", 4.5, False]
list2
```

Out[139]:
```
[1, 'sana', 4.5, False]
```

In [94]:
```
type(list2)
```

Out[94]:
```
list
```

In [96]:
```
len(list2)
```

Out[96]:
```
4
```

In [140…
```
list2[2]
```

Out[140]:
```
4.5
```

In [141…
```
list2[3]
```

Out[141]:
```
False
```

In [142…
```
list2[0:2]
```

Out[142]:
```
[1, 'sana']
```

In [143…
```
list2[0:4]
```

Out[143]:
```
[1, 'sana', 4.5, False]
```

In [145…
```
list3=[3,5,"sana", "like", True, False]
list3b
```

Out[145]:
```
[3, 5, 'sana', 'like', True, False]
```

In [146…
```
list3*4
```

Out[146]:
```
[3,
 5,
 'sana',
 'like',
 True,
 False,
 3,
 5,
 'sana',
 'like',
 True,
 False,
 3,
 5,
 'sana',
 'like',
 True,
 False,
 3,
 5,
 'sana',
 'like',
 True,
 False]
```

Reverse is function, ( ) after function

In [150…
```
list2.reverse()
list2
```

Out[150]:
```
[False, 4.5, 'sana', 1]
```

In [151…
```
list3.reverse()
list3
```

Out[151]:
```
[False, True, 'like', 'sana', 5, 3]
```

Append means to add some thing to the list,

In [172…
```
list3.append("university")
list3
```

Out[172]:
```
[False, True, 'like', 'sana', 5, 3, 'university']
```

clicking on appendmake more addition

In [176…
```
list3.append("university")
list3
```

Out[176]:
```
[False,
 True,
 'like',
 'sana',
 5,
 3,
 'university',
 'university',
 'university',
 'university',
 'university']
```

if there is loats of university here by clicking and you want to remove some of them

In [180…
```
list3.remove("university")
list3
```

Out[180]:
```
[False, True, 'like', 'sana', 5, 3, 'university']
```

In [215…
```
list4=["sana",3,5,5,3,3,3,3,3,False]
list4
```

Out[215]:
```
['sana', 3, 5, 5, 3, 3, 3, 3, 3, False]
```

In [219…
```
list4.count("sana")
```

Out[219]:
```
1
```

In [222…
```
list4.count(3)
```

Out[222]:
```
6
```

In [233…
```
list4.extend("False")
list4
```

Out[233]:
```
['F', 'a', 'l', 's', 'e']
```

In [241…
```
list5=[20, 30, 35, 50, 35, 12, 10 ,50]
list5
```

Out[241]:
```
[20, 30, 35, 50, 35, 12, 10, 50]
```

In [242…
```
len(list5)
```

Out[242]:
```
8
```

Sorting a list it means to arrange in ascending order

In [246…
```
list5.sort()
list5
```

Out[246]:
```
[10, 12, 20, 30, 35, 35, 50, 50]
```

In [247…
```
list5*3
```

Out[247]:
```
[10,
 12,
 20,
 30,
 35,
 35,
 50,
 50,
 10,
 12,
 20,
 30,
 35,
 35,
 50,
 50,
 10,
 12,
 20,
 30,
 35,
 35,
 50,
 50]
```

In [249…
```
list5+list2
```

Out[249]:
```
[10, 12, 20, 30, 35, 35, 50, 50, False, 4.5, 'sana', 1]
```

- Dictionaries

- An unordered collection of elements
- Key and Value
- Curly Baraket/braces{ }
- Mutateable/Change the value

In [260…
```
# foods and their prices
d1={"samosa":30, "pokoray":100, "Raita":50, "salad":50, "Chicken Rolls":40}
d1
```

Out[260]:
```
{'samosa': 30, 'pokoray': 100, 'Raita': 50, 'salad': 50, 'Chicken Rolls': 40}
```

In [261…
```
type(d1)
```

Out[261]:
```
dict
```

In [267…
```
# Extract Data
keys=d1.keys()
keys
```

Out[267]:
```
dict_keys(['samosa', 'pokoray', 'Raita', 'salad', 'Chicken Rolls'])
```

In [268…
```
# Extract Values
values=d1.values()
```

```
values
```

Out[268]:
```
dict_values([30, 100, 50, 50, 40])
```

In [276…
```
# Adding More first method
d1["roti"]=10
d1
```

Out[276]:
```
{'samosa': 30,
 'pokoray': 100,
 'Raita': 50,
 'salad': 50,
 'Chicken Rolls': 40,
 'roti': 10}
```

In [278…
```
#  update the values
d1["roti"]=15
d1
```

Out[278]:
```
{'samosa': 30,
 'pokoray': 100,
 'Raita': 50,
 'salad': 50,
 'Chicken Rolls': 40,
 'roti': 15}
```

In [279…
```
# d2
d2={"dates":50, "Chocolates":200, "Saviya":1000, }
d2
```

Out[279]:
```
{'dates': 50, 'Chocolates': 200, 'Saviya': 1000}
```

In [282…
```
# Concatinate
d1.update(d2)
d1
```

Out[282]:
```
{'samosa': 30,
 'pokoray': 100,
 'Raita': 50,
 'salad': 50,
 'Chicken Rolls': 40,
 'roti': 15,
 'dates': 50,
 'Chocolates': 200,
 'Saviya': 1000}
```

Sets

- Unordered and unindexed
- curly braces{ }
- No duplicate
- Not to add booleans(T/F)
- add string, float, integers

In [329…
```
s1={1,2.2,5.3, "Lahore", "sana", True}
s1
```

Out[329]:
```
{1, 2.2, 5.3, 'Lahore', 'sana'}
```

In [330…
```
s1.add("ammar1")
s1
```

Out[330]:
```
{1, 2.2, 5.3, 'Lahore', 'ammar1', 'sana'}
```

In [331…
```
s1.remove("ammar1")
s1
```

Out[331]:
```
{1, 2.2, 5.3, 'Lahore', 'sana'}
```

In [332…
```
# Add again ammar1 then it will not add, bcz no duplicate
s1.add("ammar1")
s1
```

Out[332]:
```
{1, 2.2, 5.3, 'Lahore', 'ammar1', 'sana'}
```

In [334…
```
s2={1,2,3}
s3={3,4,5}
s2.union(s3)
s2
```

Out[334]:
```
{1, 2, 3}
```

In [ ]: