TUPLES

LISTS

Dictionaries

Sets

1- Tuples

In [ ]:
```
# ordered collection of elements.
# it contains multiple elements in a single variable.
# elements are seperated by commas and enclosed within round brackets ().
# they are unchangeable, you cannot modify its contents - you can't add, remove,
# or change elements in a tuple after it has been created.
# to add something in tuple, convert the type to list, make changes in it and then
# again convert it to tuple.
# methods:- Count, Index, length
```

In [9]:
```
tup1= ("adnan", "hyder", 55, 60, 11.5)
tup1
```

Out[9]:
```
('adnan', 'hyder', 55, 60, 11.5)
```

In [84]:
```
```

Out[84]:
```
tuple
```

In [7]:
```
tup2= ("girl", "khan", 88, 32, 44)
tup2
```

Out[7]:
```
('girl', 'khan', 88, 32, 44)
```

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

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

In [24]:
```
print(tup2[2:4])
```

```
(88, 32)
```

In [10]:
```
#if there is nothing before :, then it is 0.
print(tup1[:2])
```

```
('adnan', 'hyder')
```

In [81]:
```
#if there is nothing after :, then it is the length of tuple.
print(tup1[2:])
```
```
(55, 60, 11.5)
```

In [25]:
```
type(tup2)
```
Out[25]:
```
tuple
```

In [26]:
```
len(tup2)
```
Out[26]:
```
5
```

In [27]:
```
tup2.count("girl")
```
Out[27]:
```
1
```

In [28]:
```
tup3= tup1 + tup2
tup3
```
Out[28]:
```
('adnan', 'hyder', 55, 60, 11.5, 'girl', 'khan', 88, 32, 44)
```

In [29]:
```
tup3[-1]
```
Out[29]:
```
44
```

In [31]:
```
if 11.5 in tup3:
    print("yes")
else:
    print("No")
```
```
yes
```

2- Lists

In [ ]:
```
# ordered collection of elements.
# it contains multiple elements in a single variable.
# elements are seperated by commas and enclosed within squared brackets [].
# they are changeable, you can modify its contents - you can add, remove, or
# change elements in a list after it has been created.
# Methods:- Sort, reverse, Index, Count, Copy, Insert, Extend, Concatenate.
```

In [59]:
```
list1= [2,3,55,7, "adnan"]
list1
```
Out[59]:
```
[2, 3, 55, 7, 'adnan']
```

In [33]:
```
len(list1)
```
Out[33]:
```
5
```

In [34]:
```
list1.count("adnan")
```
Out[34]:
```
1
```

```
In [35]: list1[4]

Out[35]: 'adnan'

In [36]: list2= ["khan", 32, 11, "hyder"]
         list2

Out[36]: ['khan', 32, 11, 'hyder']

In [37]: list2[3]

Out[37]: 'hyder'

In [38]: list2[1:3]

Out[38]: [32, 11]

In [39]: list2[:3]

Out[39]: ['khan', 32, 11]

In [40]: list2[0:]

Out[40]: ['khan', 32, 11, 'hyder']

In [46]: list1

Out[46]: ['khan', 'khan', 'khan', 2, 3, 55, 7, 'adnan']

In [48]: list1.insert(1, "khan") #mention that on which number it is to be placed
         list1

Out[48]: [2, 'khan', 3, 55, 7, 'adnan']

In [49]: list2

Out[49]: ['khan', 32, 11, 'hyder']

In [50]: list2.append(55)
         list2

Out[50]: ['khan', 32, 11, 'hyder', 55]

In [51]: list1

Out[51]: [2, 'khan', 3, 55, 7, 'adnan']

In [52]: type(list1)

Out[52]: list

In [53]: type(list1)
```

```
Out[53]: list

In [60]: list1.append(55)
         list1

Out[60]: [2, 3, 55, 7, 'adnan', 55]

In [64]: list1.append(66)
         list1

Out[64]: [2, 3, 55, 7, 'adnan', 55, 66, 66, 66]

In [65]: list1

Out[65]: [2, 3, 55, 7, 'adnan', 55, 66, 66, 66]

In [35]: list1.index(66)
         list1

Out[35]: [2, 3, 7, 66]

In [36]: list1.index(66)
         list1

Out[36]: [2, 3, 7, 66]
```

3- Dictionaries

```
In [ ]: # ordered collection of data items.
        # they store multiple items in a singe variable.
        # they are key-value pairs separated by commas and
        # enclosed with curly bracket {}
        # update, clear, pop, del,

In [37]: dic1= {"samosa":50, "pakore":70, "besan":40}
         dic1

Out[37]: {'samosa': 50, 'pakore': 70, 'besan': 40}

In [38]: len(dic1)

Out[38]: 3

In [47]: #to update the contents of the dictionary, we must have to use these [] brackets
         dic1["samosa"] = 70
         dic1

Out[47]: {'samosa': 70, 'pakore': 70, 'besan': 40}

In [48]: dic1

Out[48]: {'samosa': 70, 'pakore': 70, 'besan': 40}
```

In [53]:
```
dic2= {"mama":10, "papa":20, "api":30, "baji":40}
```

In [54]:
```
dic2
```

Out[54]:
```
{'mama': 10, 'papa': 20, 'api': 30, 'baji': 40}
```

In [55]:
```
dic1.update(dic2)
dic1
```

Out[55]:
```
{'samosa': 70,
 'pakore': 70,
 'besan': 40,
 'mama': 10,
 'papa': 20,
 'api': 30,
 'baji': 40}
```

In [57]:
```
keys= dic1.keys()
keys
```

Out[57]:
```
dict_keys(['samosa', 'pakore', 'besan', 'mama', 'papa', 'api', 'baji'])
```

In [58]:
```
values= dic1.values()
values
```

Out[58]:
```
dict_values([70, 70, 40, 10, 20, 30, 40])
```

In [59]:
```
keys= dic2.values()
keys
```

Out[59]:
```
dict_values([10, 20, 30, 40])
```

In [60]:
```
values= dic1.keys()
keys
```

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

In [2]:
```
dic5= {
    "hyder": 50,
    "ali": 30,
    "hassan": 45,
    "umair": 55,
    "khan": 65
}
dic5
```

Out[2]:
```
{'hyder': 50, 'ali': 30, 'hassan': 45, 'umair': 55, 'khan': 65}
```

In [27]:
```
dic5["umair"]
```

Out[27]:
```
55
```

In [31]:
```
dic6= {
    666: "hyder",
    888: "khan",
    900: "saqib",
    909: "jani"
}
dic6
```

Out[31]:
```
{666: 'hyder', 888: 'khan', 900: 'saqib', 909: 'jani'}
```

In [32]:
```
dic6[909]
```

Out[32]:
```
'jani'
```

4- Sets

In [ ]:
```
# collection of unordered & well defined objects.
# separated by commas and enlosed within curly brackets {}.
# unchangeable.
# do not contain duplicate item.
# Methods:- union, intersection, update, difference, isdisjoin, issuperset, issubset,
# add, remove, discard,delete, clear, pop
```

In [1]:
```
s= {2, 4, 2, 6}
s
```

Out[1]:
```
{2, 4, 6}
```

In [2]:
```
type(s)
```

Out[2]:
```
set
```

In [3]:
```
set = {"adnan", 55, False, "khan", "sher"}
set
```

Out[3]:
```
{55, False, 'adnan', 'khan', 'sher'}
```

In [4]:
```
set
```

Out[4]:
```
{55, False, 'adnan', 'khan', 'sher'}
```

In [7]:
```
s1 = {4, 5, 6, 7}
s1
```

Out[7]:
```
{4, 5, 6, 7}
```

In [8]:
```
s2 = {1, 2, 3, 4}
s2
```

Out[8]:
```
{1, 2, 3, 4}
```

In [9]:
```
#AUB
s1.union(s2)
```

Out[9]:
```
{1, 2, 3, 4, 5, 6, 7}
```

In [56]:
```
#A∩B
s1.intersection(s2)
```
Out[56]:
```
{4}
```

In [13]:
```
#A-B
s2.difference(s1)
```
Out[13]:
```
{1, 2, 3}
```

In [58]:
```
s1.update(s2)
s1
```
Out[58]:
```
{1, 2, 3, 4, 5, 6, 7}
```

In [16]:
```
s3 = {1,2,3,4,5}
s3
```
Out[16]:
```
{1, 2, 3, 4, 5}
```

In [17]:
```
s4 = {1,2,3,4}
s4
```
Out[17]:
```
{1, 2, 3, 4}
```

In [20]:
```
#If therer is even one element same in both sets, it will be false, if no same element
s4.isdisjoint(s3)
```
Out[20]:
```
False
```

In [22]:
```
#if all the elements of set2 are available in set3 then true, otherwise false
s4.issubset(s3)
```
Out[22]:
```
True
```

In [23]:
```
#the different value between both sets will be printed.
s3.symmetric_difference(s4)
```
Out[23]:
```
{5}
```