In [ ]:
```
#T12_Tuples,Set,Dict - Practice - 27th Oct
```

In [ ]:
```
#1. Tuple is similar to List except that the objects in tuple are immutable which m
#2. When we do not want to change the data over time, tuple is a preferred data typ
#3. Iterating over the elements of a tuple is faster compared to iterating over a l
```

In [ ]:
```
#Tuple practice done in previous practice
```

In [ ]:
```
#Sets
#1) Unordered & Unindexed collection of items.
#2) Set elements are unique. Duplicate elements are not allowed.
#3) Set elements are immutable (cannot be changed).
#4) Set itself is mutable. We can add or remove items from it.
```

In [15]:
```
myset = {1,2,3,4,5}    #Set of numbers
myset
```

Out[15]:
```
{1, 2, 3, 4, 5}
```

In [16]:
```
len(myset)
```

Out[16]:
```
5
```

In [17]:
```
my_set = {1,1,2,2,3,4,5,5}
my_set         #Duplicate elements are not allowed
```

Out[17]:
```
{1, 2, 3, 4, 5}
```

In [18]:
```
myset1 = {1.79, 2.08, 3.99, 4.56, 5.45}    #Set of float numbers
myset1
```

Out[18]:
```
{1.79, 2.08, 3.99, 4.56, 5.45}
```

In [19]:
```
myset2 = {'Rashmi', 'Saanchi', "Balurkar"}     #Set of strings
myset2
```

Out[19]:
```
{'Balurkar', 'Rashmi', 'Saanchi'}
```

In [20]:
```
myset3 = {10,20, "Rashmi", (11,22,33)}    #Mixed datatypes
myset3
```

Out[20]:
```
{(11, 22, 33), 10, 20, 'Rashmi'}
```

In [21]:
```
myset3 = {10,20, "Rashmi", [11,22,33]}    #Set doesn't allow mutable items like lis
myset3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 myset3 = {10,20, "Rashmi", [11,22,33]}    #Set doesn't allow mutable items l
ike list
      2 myset3

TypeError: unhashable type: 'list'
```

In [22]:
```
myset4 = set()     #Create an empty set
myset4
```

Out[22]:
```
set()
```

In [23]:
```
print(type(myset4))
```

```
<class 'set'>
```

In [24]:
```
my_set1 = set(('one', 'two', 'three', 'four'))
my_set1
```

Out[24]:
```
{'four', 'one', 'three', 'two'}
```

In [ ]:
```
#Loop through a set
```

In [25]:
```
myset = {'one', 'two','three','four','five','six','seven','eight'}
for i in myset:
    print(i)
```

```
five
six
one
seven
eight
two
three
four
```

In [26]:
```
for i in enumerate(myset):
    print(i)
```

```
(0, 'five')
(1, 'six')
(2, 'one')
(3, 'seven')
(4, 'eight')
(5, 'two')
(6, 'three')
(7, 'four')
```

In [27]:
```
#Set Membership
myset
```

Out[27]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [28]:
```
'one' in myset    #Check if 'one' exist in the set
```

Out[28]: True

In [29]:
```
'ten' in myset
```

Out[29]: False

In [30]:
```
if 'three' in myset:
    print('Three is present in the set')
else:
    print('Three is not present in the set')
```

```
Three is present in the set
```

In [31]:
```
if 'eleven' in myset:
    print('Eleven is present in the set')
else:
    print('Eleven is not present in the set')
```

```
Eleven is not present in the set
```

In [32]:
```
#Add & Remove Items
myset
```

Out[32]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [33]:
```
myset.add('NINE')       #Add item to a set using add() method
myset
```

Out[33]: {'NINE', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [34]:
```
myset.update(['TEN','ELEVEN','TWELVE'])     #Add multiple items to a set using list
myset
```

Out[34]:
```
{'ELEVEN',
 'NINE',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [35]:
```
myset.remove('NINE')      #Remove item using remove() method
myset
```

Out[35]:
```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [36]:
```
myset.discard('TEN')     #Remove item using discard() method
myset
```

Out[36]:
```
{'ELEVEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [37]:
```
myset.remove('100')
myset
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[37], line 1
----> 1 myset.remove('100')
      2 myset

KeyError: '100'
```

In [39]:
```
myset.discard('100')    #No error if element not present
myset
```

Out[39]:
```
{'ELEVEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [40]:
```
myset.clear()     #Delete all items in a set
myset
```

Out[40]:
```
set()
```

In [41]:
```
del myset    #Delete the set object
myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[41], line 2
      1 del myset    #Delete the set object
----> 2 myset

NameError: name 'myset' is not defined
```

In [ ]:
```
#Copy Set
```

In [42]:
```
myset = {'one', 'two','three','four','five','six','seven','eight'}
myset
```

Out[42]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [43]:
```
myset1 = myset   #Create a new reference "myset1"
myset1
```

Out[43]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [44]:
```
id(myset), id(myset1)     #ID will be same
```

Out[44]:
```
(2438524408704, 2438524408704)
```

In [45]:
```
my_set = myset.copy()      #Create a copy
my_set
```

Out[45]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [46]:
```
id(my_set)     #ID will be different
```

Out[46]:
```
2438524407136
```

In [47]:
```
myset.add('nine')
myset
```

Out[47]:
```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [48]:
```
myset1    #points to same myset, so this will also be impacted
```

Out[48]:
```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [50]:
```
my_set     #No impact
```

Out[50]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [ ]:
```
#Set Operation
```

In [51]:
```
#Union
A = {1,2,3,4,5}
```

```
B = {4,5,6,7,8}
C = {8,9,10}
```

In [52]:
```
A | B       #Union of A and B (All elements from both sets, no duplicates)
```

Out[52]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [53]:
```
A.union(B)      #Union of A and B
```

Out[53]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [54]:
```
A.union(B, C)    #Union of A, B and C
```

Out[54]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [56]:
```
A.update(B,C)   #Updates the set calling the update()
A
```

Out[56]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [57]:
```
#Intersection
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [58]:
```
A & B     #Intersection of A and B (Common items in both sets)
```

Out[58]:
```
{4, 5}
```

In [59]:
```
A.intersection(B)      #Intersection of A and B
```

Out[59]:
```
{4, 5}
```

In [ ]:
```
#Difference
```

In [60]:
```
A - B     #Difference of sets
```

Out[60]:
```
{1, 2, 3}
```

In [61]:
```
A.difference(B)   #Difference of sets
```

Out[61]:
```
{1, 2, 3}
```

In [62]:
```
B -A
```

Out[62]:
```
{6, 7, 8}
```

In [63]:
```
B.difference(A)
```

Out[63]:
```
{6, 7, 8}
```

In [64]:
```
B.difference_update(A)        #B updated with difference of B & A
B
```

Out[64]: {6, 7, 8}

In [66]:
```
#Symmetric Difference
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [67]:
```
A ^ B       #Symmetric difference (Set of elements in A and B but not in both)
```

Out[67]: {1, 2, 3, 6, 7, 8}

In [68]:
```
A.symmetric_difference(B)      #Symmetric difference of sets
```

Out[68]: {1, 2, 3, 6, 7, 8}

In [69]:
```
A.symmetric_difference_update(B)     #Set A is updated with symmetric difference of
A
```

Out[69]: {1, 2, 3, 6, 7, 8}

In [70]:
```
#Subset, Superset & Disjoint
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

In [71]:
```
B.issubset(A)       #All elements of B are in A ?
```

Out[71]: True

In [72]:
```
A.issuperset(B)     #All elements of B are in A ?
```

Out[72]: True

In [73]:
```
C.isdisjoint(A)    #No common elements between Set A and Set C
```

Out[73]: True

In [74]:
```
B.isdisjoint(A)
```

Out[74]: False

In [75]:
```
#Other Built in Functions
A
```

Out[75]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [76]:
```
sum(A)
```

Out[76]: 45

In [77]:
```
max(A)
```

Out[77]: 9

In [78]:
```
min(A)
```

Out[78]: 1

In [79]:
```
len(A)
```

Out[79]: 9

In [80]:
```
list(enumerate(A))
```

Out[80]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]

In [81]:
```
D = sorted(A, reverse=True)
D
```

Out[81]: [9, 8, 7, 6, 5, 4, 3, 2, 1]

In [82]:
```
sorted(D)
```

Out[82]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [ ]:
```
#Dictionary
'''Dictionary is a mutable data type in Python.
A python dictionary is a collection of key and value pairs separated by a colon (:)
Keys must be unique in a dictionary, duplicate values are allowed.'''
```

In [83]:
```
mydict = dict()     #Empty dictionary
mydict
```

Out[83]: {}

In [84]:
```
mydict = {}    #Empty dictionary
mydict
```

Out[84]: {}

In [87]:
```
mydict = {1:'one',2:'two',3:'three'}      #dictionary with integer keys
mydict
```

Out[87]: {1: 'one', 2: 'two', 3: 'three'}

In [88]:
```
mydict = dict({1:'one',2:'two',3:'three'})  #Create dictionary with dict()
mydict
```

Out[88]: {1: 'one', 2: 'two', 3: 'three'}

In [90]:
```
mydict = {'A':'one','B':'two','C':'three'}      #dictionary with character keys
mydict
```

Out[90]: {'A': 'one', 'B': 'two', 'C': 'three'}

In [94]:
```
mydict = {'1':'one','A':'two','3':'three'}      #dictionary with mixed keys
mydict
```

Out[94]:
```
{'1': 'one', 'A': 'two', '3': 'three'}
```

In [95]:
```
mydict.keys()     #Return dictionary keys
```

Out[95]:
```
dict_keys(['1', 'A', '3'])
```

In [96]:
```
mydict.values()     #Return dictionary values
```

Out[96]:
```
dict_values(['one', 'two', 'three'])
```

In [97]:
```
mydict.items     #Access each key value pair
```

Out[97]:
```
<function dict.items()>
```

In [98]:
```
mydict.items()     #Access each key value pair
```

Out[98]:
```
dict_items([('1', 'one'), ('A', 'two'), ('3', 'three')])
```

In [99]:
```
mydict = {1:'one', 2:'two','A':['Rashmi','Saanchi','Balurkar']}    #Dictionary with
mydict
```

Out[99]:
```
{1: 'one', 2: 'two', 'A': ['Rashmi', 'Saanchi', 'Balurkar']}
```

In [100…
```
mydict = {1:'one', 2:'two','A':['Rashmi','Saanchi','Balurkar'], 'B':['Bat', 'Cat',
mydict
```

Out[100…
```
{1: 'one',
 2: 'two',
 'A': ['Rashmi', 'Saanchi', 'Balurkar'],
 'B': ['Bat', 'Cat', 'Hat']}
```

In [106…
```
mydict = {1:'one', 2:'two','A':{'Name':'Rashmi','Age':36}, 'B':('Bat', 'Cat', 'Hat'
mydict
```

Out[106…
```
{1: 'one',
 2: 'two',
 'A': {'Name': 'Rashmi', 'Age': 36},
 'B': ('Bat', 'Cat', 'Hat')}
```

In [107…
```
keys = {'a', 'b', 'c', 'd'}
mydict3 = dict.fromkeys(keys)    #Create  a dictionary from a sequence of keys
mydict3
```

Out[107…
```
{'d': None, 'c': None, 'b': None, 'a': None}
```

In [108…
```
keys = {'a', 'b', 'c', 'd'}
value = [10,20,30]
mydict3 = dict.fromkeys(keys,value)    #Create  adictionary from a sequence of keys
mydict3
```

Out[108…
```
{'d': [10, 20, 30], 'c': [10, 20, 30], 'b': [10, 20, 30], 'a': [10, 20, 30]}
```

In [109…
```
value.append(40)
mydict3
```

Out[109…
```
{'d': [10, 20, 30, 40],
 'c': [10, 20, 30, 40],
 'b': [10, 20, 30, 40],
 'a': [10, 20, 30, 40]}
```

In [ ]:
```
#Accessing Items
```

In [110…
```
mydict = {1:'one',2:'two',3:'three',4:'four'}
mydict
```

Out[110…
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [111…
```
mydict[1]      #Access item using key
```

Out[111…
```
'one'
```

In [112…
```
mydict.get(1)      #Access item using get()
```

Out[112…
```
'one'
```

In [113…
```
mydict1 = {'Name' : 'Rashmi', 'ID': 2698890, 'DOB' : 1988, 'job':'Developer'}
mydict1
```

Out[113…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, ' job': 'Developer'}
```

In [114…
```
mydict1['Name']
```

Out[114…
```
'Rashmi'
```

In [117…
```
mydict1.get('job')     #Access item using get()
```

In [ ]:
```
#Add, Remove and Change items
```

In [129…
```
mydict1 = {'Name':'Rashmi','ID': 2698890, 'DOB' : 1988, 'Address':'Thane'}
mydict1
```

Out[129…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Thane'}
```

In [130…
```
mydict1['DOB'] = 1985       #Changing Dictionary Items
```

In [131…
```
mydict1['Address'] = 'Mumbai'
mydict1
```

Out[131…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1985, 'Address': 'Mumbai'}
```

In [132…
```
dict1 = {'DOB' : 1983}
mydict1.update(dict1)
```

```
mydict1
```

Out[132…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1983, 'Address': 'Mumbai'}
```

In [133…
```
mydict1['Job'] = 'Software Engineer'      #Adding items in the dictionary
mydict1
```

Out[133…
```
{'Name': 'Rashmi',
 'ID': 2698890,
 'DOB': 1983,
 'Address': 'Mumbai',
 'Job': 'Software Engineer'}
```

In [134…
```
mydict1.pop('Job')     #Remove items using pop()
```

Out[134…
```
'Software Engineer'
```

In [135…
```
mydict1
```

Out[135…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1983, 'Address': 'Mumbai'}
```

In [136…
```
mydict1.popitem()      #A random item is removed
```

Out[136…
```
('Address', 'Mumbai')
```

In [137…
```
mydict1
```

Out[137…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1983}
```

In [138…
```
del[mydict1['ID']]      #Remove item using del method
```

In [139…
```
mydict1
```

Out[139…
```
{'Name': 'Rashmi', 'DOB': 1983}
```

In [140…
```
mydict1.clear()     #Delete all items using clear()
mydict1
```

Out[140…
```
{}
```

In [141…
```
del mydict1     #Delete the object
```

In [142…
```
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[142], line 1
----> 1 mydict1

NameError: name 'mydict1' is not defined
```

In [ ]:
```
#Copy Dictionary
```

In [143…
```
mydict = {'Name':'Rashmi','ID': 2698890, 'DOB' : 1988, 'Address':'Thane'}
mydict
```

Out[143…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Thane'}
```

In [144…
```
mydict1 = mydict     #Create a new reference
```

In [145…
```
id(mydict), id(mydict1)     #Address of both will be same
```

Out[145…
```
(2438514238208, 2438514238208)
```

In [146…
```
mydict2 = mydict.copy()     #Create a copy
```

In [147…
```
id(mydict2)                 #Address is different
```

Out[147…
```
2438522983872
```

In [148…
```
mydict['Address'] = 'Mumbai'
mydict
```

Out[148…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Mumbai'}
```

In [149…
```
mydict1     #Also impacted as pointing to same
```

Out[149…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Mumbai'}
```

In [150…
```
mydict2     #Not impacted as its the copy
```

Out[150…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Thane'}
```

In [ ]:
```
#Loop through Dictionary
```

In [151…
```
mydict1 = {'Name':'Rashmi','ID': 2698890, 'DOB' : 1988, 'Address':'Thane'}
mydict1
```

Out[151…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Address': 'Thane'}
```

In [152…
```
for i in mydict1:
    print(i, ':', mydict1[i])     #Key and value pair
```

```
Name : Rashmi
ID : 2698890
DOB : 1988
Address : Thane
```

In [153…
```
for i in mydict1:
    print(i)
```

```
Name
ID
DOB
Address
```

In [154…
```
for i in mydict1:
    print(mydict1[i])     #Dictionary items
```

```
Rashmi
2698890
1988
Thane
```

In [ ]:
```
#Dictionary membership
```

In [155…
```
mydict1 = {'Name':'Rashmi','ID': 2698890, 'DOB' : 1988, 'Job':'Developer'}
mydict1
```

Out[155…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Job': 'Developer'}
```

In [156…
```
'Name' in mydict1      #Check key present ?
```

Out[156…
```
True
```

In [157…
```
'Rashmi' in mydict1     #Test can be done only for keys
```

Out[157…
```
False
```

In [158…
```
'ID' in mydict1
```

Out[158…
```
True
```

In [159…
```
'Address' in mydict1
```

Out[159…
```
False
```

In [164…
```
mydict2 = mydict1.fromkeys('ID','10605120')     #Tried myself
```

In [165…
```
mydict2
```

Out[165…
```
{'I': '10605120', 'D': '10605120'}
```

In [ ]:
```
# All / Any
'''The all() method returns:
True - If all all keys of the dictionary are true
False - If any key of the dictionary is false
The any() function returns True if any key of the dictionary is True. If not, any()
```

In [166…
```
mydict1 = {'Name':'Rashmi','ID': 2698890, 'DOB' : 1988, 'Job':'Developer'}
mydict1
```

Out[166…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Job': 'Developer'}
```

In [167…
```
all(mydict1)            #Check if any 0
```

Out[167…
```
True
```

In [168…
```
mydict1['Marks'] = 0
mydict1
```

Out[168…
```
{'Name': 'Rashmi', 'ID': 2698890, 'DOB': 1988, 'Job': 'Developer', 'Marks': 0}
```

In [169…
```
all(mydict1)
```

Out[169…
```
True
```

In [170…
```
mydict1['History'] = 'False'
mydict1
```

Out[170…
```
{'Name': 'Rashmi',
 'ID': 2698890,
 'DOB': 1988,
 'Job': 'Developer',
 'Marks': 0,
 'History': 'False'}
```

In [171…
```
all(mydict1)
```

Out[171…
```
True
```

In [172…
```
any(mydict1)
```

Out[172…
```
True
```

In [ ]:
```

```