24/10/25 TUPLE

In [286…
```
t=()
```

In [288…
```
type(t)
```

Out[288…
```
tuple
```

In [290…
```
tt=(1,2,3,4,5)
tt
```

Out[290…
```
(1, 2, 3, 4, 5)
```

In [292…
```
type(tt)
```

Out[292…
```
tuple
```

In [294…
```
# tuple allows multiple datatypes
t1=(1,2.3,"anjali",True,10+20j)
t1
```

Out[294…
```
(1, 2.3, 'anjali', True, (10+20j))
```

In [296…
```
#allows duplicates
t2=(10,20,"anjali",True,10,"python","anjali",30,100,30,50)
t2
```

Out[296…
```
(10, 20, 'anjali', True, 10, 'python', 'anjali', 30, 100, 30, 50)
```

In [298…
```
t2[0]
```

Out[298…
```
10
```

In [300…
```
t2[4]
```

Out[300…
```
10
```

In [302…
```
t2[2]
```

Out[302…
```
'anjali'
```

In [304…
```
t2[:]
```

Out[304…
```
(10, 20, 'anjali', True, 10, 'python', 'anjali', 30, 100, 30, 50)
```

In [306…
```
t2[::]
```

Out[306…
```
(10, 20, 'anjali', True, 10, 'python', 'anjali', 30, 100, 30, 50)
```

In [308…
```
t2[::2]
```

```
Out[308…  (10, 'anjali', 10, 'anjali', 100, 50)
```

```
In [310…  t2[::-1]
```

```
Out[310…  (50, 30, 100, 30, 'anjali', 'python', 10, True, 'anjali', 20, 10)
```

```
In [312…  t2
```

```
Out[312…  (10, 20, 'anjali', True, 10, 'python', 'anjali', 30, 100, 30, 50)
```

```
In [314…  #count fuction
          t2.count(10)
```

```
Out[314…  2
```

```
In [316…  t2.count("anjali")
```

```
Out[316…  2
```

```
In [318…  t2.count(500)
```

```
Out[318…  0
```

```
In [320…  t2.count(100)
```

```
Out[320…  1
```

```
In [322…  #index fuction
          print(t2)
          t2.index(10)
```

```
         (10, 20, 'anjali', True, 10, 'python', 'anjali', 30, 100, 30, 50)
Out[322…  0
```

```
In [324…  t2.index("anjali")
```

```
Out[324…  2
```

```
In [326…  t2.index(100)
```

```
Out[326…  8
```

```
In [328…  t2.index(100)
```

```
Out[328…  8
```

```
In [330…  t3=(20,30,10,200,400,150,100)
          sorted(t3)
```

```
Out[330…  [10, 20, 30, 100, 150, 200, 400]
```

```
In [332…  sorted(t3,reverse=True)
```

Out[332…

```
[400, 200, 150, 100, 30, 20, 10]
```

SET

In [335…

```
s={}
type(s)
```

Out[335…

```
dict
```

In [337…

```
#creating a empty set
s1=set()
type(s1)
```

Out[337…

```
set
```

In [339…

```
s2={10,20,30,40,50}
s2
```

Out[339…

```
{10, 20, 30, 40, 50}
```

In [341…

```
#different datatypes are allowed
s3={10,20,"anjali",True,5.0,2.56,10+20j}
s3
```

Out[341…

```
{(10+20j), 10, 2.56, 20, 5.0, True, 'anjali'}
```

In [343…

```
#does not allow duplicates
s4={10,30,28,59,20,"1","anjali",2.5,30,"anjali","python"}
s4
```

Out[343…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python'}
```

In [345…

```
#does not allow indexing and slicing
s4[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[345], line 2
      1 #does not allow indexing and slicing
----> 2 s4[0]

TypeError: 'set' object is not subscriptable
```

In [347…

```
s4[4]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[347], line 1
----> 1 s4[4]

TypeError: 'set' object is not subscriptable
```

In [349…

```
s4[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[349], line 1
----> 1 s4[:]

TypeError: 'set' object is not subscriptable
```

In [351…

```
s4[::]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[351], line 1
----> 1 s4[::]

TypeError: 'set' object is not subscriptable
```

In [353…

```
#add function
s4.add("set")
s4
```

Out[353…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python', 'set'}
```

In [355…

```
#copy function
s5=s4.copy()
s5
```

Out[355…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python', 'set'}
```

In [357…

```
#clear function
s4.clear()
s4
```

Out[357…

```
set()
```

In [359…

```
s5
```

Out[359…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python', 'set'}
```

In [361…

```
#pop function
s5.pop()
```

Out[361…

```
'set'
```

In [363…

```
s5
```

Out[363…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python'}
```

In [365…

```
#remove function
s5.remove("anjali")
s5
```

Out[365…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'python'}
```

In [367…

```
s5.remove(500)
s5
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[367], line 1
----> 1 s5.remove(500)
      2 s5

KeyError: 500
```

In [369…

```
s5.add("anjali")
s5
```

Out[369…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'anjali', 'python'}
```

In [371…

```
#discord fuction
s5.discard("anjali")
s5
```

Out[371…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'python'}
```

In [373…

```
s5.discard(500)
s5
```

Out[373…

```
{'1', 10, 2.5, 20, 28, 30, 59, 'python'}
```

SET OPERATIONS

In [376…

```
#union
a={1,2,3,4,5}
b={4,5,6,7,8}
c={8,9,10}
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [378…

```
a.union(b)
```

Out[378…

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [380…

```
a|b
```

Out[380…

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [382…

```
b.union(c)
```

Out[382…

```
{4, 5, 6, 7, 8, 9, 10}
```

In [384…
```
b|c
```
Out[384…
```
{4, 5, 6, 7, 8, 9, 10}
```
In [386…
```
c.union(a)
```
Out[386…
```
{1, 2, 3, 4, 5, 8, 9, 10}
```
In [388…
```
c|a
```
Out[388…
```
{1, 2, 3, 4, 5, 8, 9, 10}
```
In [390…
```
a.union(b,c)
```
Out[390…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```
In [392…
```
a|b|c
```
Out[392…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```
In [394…
```
#intersection
print(a)
print(b)
print(c)
```
```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```
In [396…
```
a.intersection(b)
```
Out[396…
```
{4, 5}
```
In [398…
```
a&b
```
Out[398…
```
{4, 5}
```
In [400…
```
b.intersection(c)
```
Out[400…
```
{8}
```
In [402…
```
b&c
```
Out[402…
```
{8}
```
In [404…
```
c.intersection(a)
```
Out[404…
```
set()
```
In [406…
```
a.intersection(b,c)
```
Out[406…
```
set()
```

In [408…

```
a&b&c
```

Out[408…

```
set()
```

In [410…

```
b.intersection(a)
```

Out[410…

```
{4, 5}
```

In [412…

```
b&a
```

Out[412…

```
{4, 5}
```

In [414…

```
#Difference
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [416…

```
a.difference(b)
```

Out[416…

```
{1, 2, 3}
```

In [418…

```
a-b
```

Out[418…

```
{1, 2, 3}
```

In [420…

```
b.difference(a)
```

Out[420…

```
{6, 7, 8}
```

In [422…

```
b-a
```

Out[422…

```
{6, 7, 8}
```

In [424…

```
a.difference(b,c)
```

Out[424…

```
{1, 2, 3}
```

In [426…

```
c.difference(a,b)
```

Out[426…

```
{9, 10}
```

In [428…

```
c-a-b
```

Out[428…

```
{9, 10}
```

27/10/25

symmetric difference

In [431…

```
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [433…

```
a.symmetric_difference(b)
```

Out[433…

```
{1, 2, 3, 6, 7, 8}
```

In [435…

```
a^b
```

Out[435…

```
{1, 2, 3, 6, 7, 8}
```

In [437…

```
a.symmetric_difference(c)
```

Out[437…

```
{1, 2, 3, 4, 5, 8, 9, 10}
```

In [439…

```
a^c
```

Out[439…

```
{1, 2, 3, 4, 5, 8, 9, 10}
```

In [441…

```
b.symmetric_difference(a)
```

Out[441…

```
{1, 2, 3, 6, 7, 8}
```

In [443…

```
b.symmetric_difference(c)
```

Out[443…

```
{4, 5, 6, 7, 9, 10}
```

In [445…

```
b^c
```

Out[445…

```
{4, 5, 6, 7, 9, 10}
```

In [447…

```
c.symmetric_difference(b)
```

Out[447…

```
{4, 5, 6, 7, 9, 10}
```

In [449…

```
#symmetric difference update
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [451…

```
a.symmetric_difference_update(b)
a
```

Out[451…

```
{1, 2, 3, 6, 7, 8}
```

In [453…

```
print(a)
```

```
{1, 2, 3, 6, 7, 8}
```

In [455…

```
b.symmetric_difference_update(c)
b
```

Out[455…

```
{4, 5, 6, 7, 9, 10}
```

In [457…

```
a.symmetric_difference_update(c)
a
```

Out[457…

```
{1, 2, 3, 6, 7, 9, 10}
```

In [459…

```
c.symmetric_difference_update(a)
c
```

Out[459…

```
{1, 2, 3, 6, 7, 8}
```

In [461…

```
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 6, 7, 9, 10}
{4, 5, 6, 7, 9, 10}
{1, 2, 3, 6, 7, 8}
```

In [463…

```
#superset,subset,and disjoint set
s1={1,2,3,4,5,6,7,8,9}
s2={4,5,6,7,8}
s3={10,20,30,40}
print(s1)
print(s2)
print(s3)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
{4, 5, 6, 7, 8}
{40, 10, 20, 30}
```

In [465…

```
s1.issuperset(s2)
```

Out[465…

```
True
```

In [467…

```
s2.issuperset(s1)
```

Out[467…

```
False
```

In [469…

```
s2.issubset(s1)
```

Out[469…

```
True
```

In [471…

```
s2.issubset(s3)
```

Out[471…

```
False
```

In [473…

```
s1.issubset(s2)
```

Out[473…

```
False
```

In [475…

```
s3.isdisjoint(s1)
```

Out[475…

```
True
```

In [477…

```
s3.isdisjoint(s2)
```

Out[477…

```
True
```

DICTIONARY

In [480…

```
d={}
type(d)
```

Out[480…

```
dict
```

In [482…

```
#creating a dictionary
d1={"name":"anjali","age":21,"city":"hyderabed","course":"DS WITH AI","pin":502310}
d1
```

Out[482…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'course': 'DS WITH AI',
 'pin': 502310}
```

In [484…

```
#to get keys
d1.keys()
```

Out[484…

```
dict_keys(['name', 'age', 'city', 'course', 'pin'])
```

In [486…

```
#to get values
d1.values()
```

Out[486…

```
dict_values(['anjali', 21, 'hyderabed', 'DS WITH AI', 502310])
```

In [488…

```
#to get items
d1.items()
```

Out[488…

```
dict_items([('name', 'anjali'), ('age', 21), ('city', 'hyderabed'), ('course', 'DS
WITH AI'), ('pin', 502310)])
```

In [490…

```
#update fuction
d1.update({"values":[10,20,30,40]})
d1
```

Out[490…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'course': 'DS WITH AI',
 'pin': 502310,
 'values': [10, 20, 30, 40]}
```

In [492…

```
#fromkeys fuction
keys={'a','b','c','d'}
d2=dict.fromkeys(keys)
d2
```

Out[492…

```
{'a': None, 'd': None, 'c': None, 'b': None}
```

In [494…

```
value=10
d3=dict.fromkeys(keys,value)
d3
```

Out[494…

```
{'a': 10, 'd': 10, 'c': 10, 'b': 10}
```

In [496…

```
values=[10,20,30,40]
d4=dict.fromkeys(keys,values)
d4
```

Out[496…

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

In [498…

```
d1
```

Out[498…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'course': 'DS WITH AI',
 'pin': 502310,
 'values': [10, 20, 30, 40]}
```

In [500…

```
#copy fuction
d5=d1.copy()
d5
```

Out[500…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'course': 'DS WITH AI',
 'pin': 502310,
 'values': [10, 20, 30, 40]}
```

In [502…

```
#clear fuction
d5.clear()
d5
```

Out[502…

```
{}
```

In [504…

```
d1
```

Out[504…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'course': 'DS WITH AI',
 'pin': 502310,
 'values': [10, 20, 30, 40]}
```

In [506…

```
#get fuction
d1.get("name")
```

Out[506…

```
'anjali'
```

In [508…

```
d1.get("values")
```

Out[508…

```
[10, 20, 30, 40]
```

In [510…

```
#pop fuction
d1.pop("course")
```

Out[510…

```
'DS WITH AI'
```

In [512…

```
d1
```

Out[512…

```
{'name': 'anjali',
 'age': 21,
 'city': 'hyderabed',
 'pin': 502310,
 'values': [10, 20, 30, 40]}
```

In [514…

```
#popitem fuction
d1.popitem()
```

Out[514…

```
('values', [10, 20, 30, 40])
```

In [516…

```
print(d1)
```

```
{'name': 'anjali', 'age': 21, 'city': 'hyderabed', 'pin': 502310}
```

In [518…

```
d1['name']
```

Out[518…

```
'anjali'
```

In [520…

```
d1['name']="chinni"
d1
```

Out[520…

```
{'name': 'chinni', 'age': 21, 'city': 'hyderabed', 'pin': 502310}
```

In [522…

```
for i in d1:
    print(i)
```

```
name
age
city
pin
```

In [524…

```
for i in enumerate(d1):
    print(i)
```

```
(0, 'name')
(1, 'age')
(2, 'city')
(3, 'pin')
```

RANGE

In [527…

```
range(10)
```

Out[527…

```
range(0, 10)
```

In [529…

```
range(10,20)
```

Out[529…

```
range(10, 20)
```

In [531…

```
range(10,100,5)
```

Out[531…

```
range(10, 100, 5)
```

In [533…

```
list(range(10))
```

Out[533…

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [535…

```
list(range(10,20))
```

Out[535…

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [537…

```
list(range(10,100,5))
```

Out[537…

```
[10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]
```

In [539…

```
list(range(10,100,10))
```

Out[539…

```
[10, 20, 30, 40, 50, 60, 70, 80, 90]
```

In [541…

```
list(range(10))
```

Out[541…

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [543…

```
range(20)
```

Out[543…

```
range(0, 20)
```

In [545…

```
list(range(20))
```

```
Out[545…  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
In [547…  set(range(10))
```

```
Out[547…  {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In [549…  set(range(10,20))
```

```
Out[549…  {10, 11, 12, 13, 14, 15, 16, 17, 18, 19}
```

```
In [551…  set(range(10,100,10))
```

```
Out[551…  {10, 20, 30, 40, 50, 60, 70, 80, 90}
```

```
In [553…  tuple(range(10))
```

```
Out[553…  (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

```
In [555…  tuple(range(10,20))
```

```
Out[555…  (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)
```

```
In [557…  tuple(range(10,100,5))
```

```
Out[557…  (10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95)
```

```
In [559…  dict(range(10))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[559], line 1
----> 1 dict(range(10))

TypeError: cannot convert dictionary update sequence element #0 to a sequence
```

```
In [561…  list(range(1,10))
```

```
Out[561…  [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [563…  list(range(101,105))
```

```
Out[563…  [101, 102, 103, 104]
```

```
In [565…  list(range(101,102))
```

```
Out[565…  [101]
```

```
In [ ]:
```