In [1]:
```
tup1 = ()
```

In [3]:
```
tup2 = (10,30,60)
```

In [5]:
```
tup3 = (10.77, 30.66, 60.89)
```

In [7]:
```
tup4 =('one', 'two', 'three')
```

In [9]:
```
tup5 = ('Vinay', 25, (50, 100), (150, 90))
```

In [11]:
```
tup6  = (100, 'Vinay', 17.765)
```

In [19]:
```
tup7 = ('Vinay', 25, [50, 100], [150, 90], {'John', 'David'}, (99,22,33))
```

In [21]:
```
len(tup7)
```

Out[21]: 6

In [23]:
```
# Tuple indexing
```

In [25]:
```
tup2[0]
```

Out[25]: 10

In [34]:
```
tup4[0]
```

Out[34]: 'one'

In [37]:
```
tup4[0][0]
```

Out[37]: 'o'

In [44]:
```
tup4[-1]
```

Out[44]: 'three'

In [46]:
```
tup5[-1]
```

Out[46]: (150, 90)

Tuple Slicing

In [53]:
```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seve', 'eight')
```

In [59]:
```
mytuple[2:5]
```

Out[59]: ('three', 'four', 'five')

In [61]:
```
mytuple[:3]
```

Out[61]: ('one', 'two', 'three')

In [67]:
```
mytuple[:2]
```

Out[67]: ('one', 'two')

In [70]:
```
mytuple[-3:]
```

Out[70]: ('six', 'seve', 'eight')

In [75]:
```
mytuple[-2:]
```

Out[75]: ('seve', 'eight')

In [85]:
```
mytuple[-1]
```

Out[85]: 'eight'

In [89]:
```
mytuple[:]
```

Out[89]: ('one', 'two', 'three', 'four', 'five', 'six', 'seve', 'eight')

In [92]:
```
mytuple
```

Out[92]: ('one', 'two', 'three', 'four', 'five', 'six', 'seve', 'eight')

In [98]:
```
del mytuple[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[98], line 1
----> 1 del mytuple[0]

TypeError: 'tuple' object doesn't support item deletion
```

In [104…
```
mytuple[0] =1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[104], line 1
----> 1 mytuple[0] =1

TypeError: 'tuple' object does not support item assignment
```

In [112…
```
del mytuple
```

In [118…
```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
mytuple
```

Out[118… ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [124…
```
for i in mytuple:
    print(i)
```

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

In [130…
```
for i in enumerate(mytuple):
    print(i)
```

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

In [135…
```
mytuple.count('one')
```

Out[135… 1

In [141…
```
'two' in mytuple
```

Out[141… True

In [146…
```
2 in mytuple
```

Out[146… False

In [153…
```
if 'three' in mytuple:
    print ('Found in mytuple')
else:
    print ('not found')
```

```
Found in mytuple
```

In [155…
```
if 'Ten' in mytuple:
    print ('Found in mytuple')
else:
    print ('not found')
```

```
not found
```

Index position

In [165…
```
mytuple
```

Out[165… ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [172…
```
mytuple.index('one')
```

Out[172… 0

In [182…
```
mytuple.index('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[182], line 1
----> 1 mytuple.index('ten')

ValueError: tuple.index(x): x not in tuple
```

Sorting

In [191…
```
mytuple2 = mytuple
```

In [193…
```
sorted(mytuple2,reverse=True)
```

Out[193… ['two', 'three', 'six', 'seven', 'one', 'four', 'five', 'eight']

In [199…
```
sorted(mytuple2)
```

Out[199… ['eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two']

Sets

In [208…
```
myset = {1,2,3,4,5,6}
myset
```

Out[208… {1, 2, 3, 4, 5, 6}

In [211…
```
len(myset)
```

Out[211… 6

In [218…
```
my_set = {1,1,2,2,3,4,5,5}
my_set
```

Out[218…
```
{1, 2, 3, 4, 5}
```

In [224…
```
len(my_set)
```

Out[224…
```
5
```

In [231…
```
myset1 = {1.76, 2.44, 5.66}
myset1
```

Out[231…
```
{1.76, 2.44, 5.66}
```

In [234…
```
len(myset1)
```

Out[234…
```
3
```

In [241…
```
myset3 = set()
print(type(myset3))
```

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

In [250…
```
my_set1 = set((1,2,3,4))
my_set1
```

Out[250…
```
{1, 2, 3, 4}
```

In [252…
```
print(type(my_set1))
```

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

In [258…
```
myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
myset
```

Out[258…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [260…
```
for i in myset:
    print(i)
```

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

In [269…
```
for i in enumerate(myset):
    print(i)
```

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

In [271…
```
myset
```

Out[271…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

Set Membership

In [283…
```
myset
```

Out[283…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [285…
```
'one' in myset
```

Out[285…
```
True
```

In [295…
```
'ten' in myset
```

Out[295…
```
False
```

In [298…
```
if 'two' in myset:
    print('Found in myset')
else:
    print('Not found')
```

```
Found in myset
```

In [305…
```
if 'eleven' in myset:
    print('Found in myset')
else:
    print('Not found')
```

```
Not found
```

Add & Remove Items

In [319…
```
myset
```

Out[319…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [335…
```
myset.add('Nine')
myset
```

Out[335…
```
{'Nine', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [344…
```
myset.update(['TEN','ELEVEN', 'TWELVE'])
myset
```

Out[344…
```
{'ELEVEN',
 'Nine',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [354…
```
myset.remove('two')
```

myset.pop()

In [370…
```
myset.pop(2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[370], line 1
----> 1 myset.pop(2)

TypeError: set.pop() takes no arguments (1 given)
```

In [388…
```
myset.remove('two')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[388], line 1
----> 1 myset.remove('two')

KeyError: 'two'
```

In [389…
```
myset.discard('two')
```

Set Operation

In [403…
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [408…
```
A | B
```

Out[408…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [410…
```
A.union(B,C)
```

Out[410…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [414…
```
A
```

Out[414…
```
{1, 2, 3, 4, 5}
```

In [420…
```
A.update(B,C)
```

In [422…
```
A
```

Out[422…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Intersection

In [432…
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

In [434…
```
A & B
```

Out[434…
```
{4, 5}
```

In [436…
```
A.intersection(B,C)
```

Out[436…
```
set()
```

In [438…
```
A.intersection(B)
```

Out[438…
```
{4, 5}
```

In [443…
```
A - B
```

Out[443…
```
{1, 2, 3}
```

In [445…
```
B - A
```

Out[445…
```
{6, 7, 8}
```

In [452…
```
A.difference(B)
```
Out[452…
```
{1, 2, 3}
```
In [454…
```
B.difference(A)
```
Out[454…
```
{6, 7, 8}
```
In [458…
```
B
```
Out[458…
```
{4, 5, 6, 7, 8}
```
In [460…
```
A
```
Out[460…
```
{1, 2, 3, 4, 5}
```
In [464…
```
B.difference_update(A)
```
In [468…
```
B
```
Out[468…
```
{6, 7, 8}
```

Symmetric Difference

In [480…
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```
In [482…
```
A ^ B
```
Out[482…
```
{1, 2, 3, 6, 7, 8}
```
In [486…
```
A.symmetric_difference(B)
```
Out[486…
```
{1, 2, 3, 6, 7, 8}
```
In [488…
```
A ^ B == A.symmetric_difference(B)
```
Out[488…
```
True
```
In [492…
```
A.symmetric_difference_update(B)
```
In [494…
```
A
```
Out[494…
```
{1, 2, 3, 6, 7, 8}
```

Subset, Superset & Disjoint

In [504…
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```
In [509…
```
B.issubset(A)
```
Out[509…
```
True
```
In [514…
```
A.issuperset(B)
```
Out[514…
```
True
```
In [516…
```
C.isdisjoint(A)
```
Out[516…
```
True
```
In [521…
```
B.isdisjoint(A)
```
Out[521…
```
False
```
In [526…
```
sum(A)
```
Out[526…
```
45
```
In [530…
```
max(A)
```
Out[530…
```
9
```
In [535…
```
min(A)
```
Out[535…
```
1
```
In [542…
```
len(A)
```
Out[542…
```
9
```
In [545…
```
list(enumerate(A))
```
Out[545…
```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```
In [549…
```
D = sorted(A,reverse=True)
D
```

Out[549… [9, 8, 7, 6, 5, 4, 3, 2, 1]

In [555…
```
sorted(D)
```

Out[555… [1, 2, 3, 4, 5, 6, 7, 8, 9]

Dictionary

In [566…
```
mydict = dict()
mydict
```

Out[566… {}

In [568…
```
type(mydict)
```

Out[568… dict

In [570…
```
mydict = {1: 'one', 2:'two', 3:'three'}
mydict
```

Out[570… {1: 'one', 2: 'two', 3: 'three'}

In [576…
```
mydict = dict({1: 'one', 2:'two', 3:'three'})
mydict
```

Out[576… {1: 'one', 2: 'two', 3: 'three'}

In [582…
```
mydict = {'A' :'one', 'B':'two', 'C':'three'}
mydict
```

Out[582… {'A': 'one', 'B': 'two', 'C': 'three'}

In [584…
```
mydict.keys()
```

Out[584… dict_keys(['A', 'B', 'C'])

In [590…
```
mydict.values()
```

Out[590… dict_values(['one', 'two', 'three'])

In [594…
```
mydict.items()
```

Out[594… dict_items([('A', 'one'), ('B', 'two'), ('C', 'three')])

Accessing Items

In [600…
```
mydict ={1:'one', 2:'two', 3:'three', 4:'four'}
mydict
```

Out[600… {1: 'one', 2: 'two', 3: 'three', 4: 'four'}

In [604…
```
mydict[1]
```

Out[604… 'one'

In [609…
```
mydict.get(3)
```

Out[609… 'three'

Add Remove & Change items

In [614…
```
mydict[3]= 'test'
mydict
```

Out[614… {1: 'one', 2: 'two', 3: 'test', 4: 'four'}

In [617…
```
mydict.pop()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[617], line 1
----> 1 mydict.pop()

TypeError: pop expected at least 1 argument, got 0
```

Looping through dictionary

In [623…
```
for i in mydict:
    print('Key:', i, '- value:', mydict[i])
```

```
Key: 1 - value: one
Key: 2 - value: two
Key: 3 - value: test
Key: 4 - value: four
```

In [643…
```
i =0
keys = list(mydict.keys())
while i < len(mydict.keys()):
    print(mydict[keys[i]])
    i = i + 1
```

```
one
two
test
four
```

In [ ]:

In [ ]:

In [ ]: