PYTHON DATA STRUCTURE

1 . LIST

In [595…
```
list=[]
list
```

Out[595…
```
[]
```

In [596…
```
print(type(list))
```

```
<class 'list'>
```

In [599…
```
list1=[10,20,30]
list1 #list is in intiger number
```

Out[599…
```
[10, 20, 30]
```

In [600…
```
list2=[10.73,30.66,60.76]
list2 #list of intiger
```

Out[600…
```
[10.73, 30.66, 60.76]
```

In [601…
```
list3=['one','two','three']
list3 # list of strings
```

Out[601…
```
['one', 'two', 'three']
```

In [605…
```
list4=['Asif',25,[54,86],[150,90]]
list4 # nested lists
```

Out[605…
```
['Asif', 25, [54, 86], [150, 90]]
```

In [607…
```
list5=[100,'Asif',17.59]
list5 # list of mixed datatype
```

Out[607…
```
[100, 'Asif', 17.59]
```

In [609…
```
list6=['Asif',25,[50,100],[150,90],{'John','David'}]
list6
```

Out[609…
```
['Asif', 25, [50, 100], [150, 90], {'David', 'John'}]
```

In [611…
```
len(list6)
```

Out[611…
```
5
```

In [613…
```
list2
```

Out[613…
```
[10.73, 30.66, 60.76]
```

list indexing

In [616…
```
list2[0]
```

Out[616…
```
10.73
```

In [618…
```
list3[0]
```

Out[618…
```
'one'
```

In [620…
```
list3[0][0]
```

Out[620…
```
'o'
```

In [622…
```
list3[-1]
```

Out[622…
```
'three'
```

In [624…
```
list4[-1]
```

Out[624…
```
[150, 90]
```

list slicing

In [627…
```
mylist=['one' , 'two' , 'three' , 'four' , 'five' , 'six' , 'seven' , 'eight']
mylist
```

Out[627…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [629…
```
mylist[0:3]
```

Out[629…
```
['one', 'two', 'three']
```

In [631…
```
mylist[2:5]
```

Out[631…
```
['three', 'four', 'five']
```

In [633…
```
mylist[:3]
```

Out[633…
```
['one', 'two', 'three']
```

In [635…
```
mylist[:2]
```

Out[635…
```
['one', 'two']
```

In [637…
```
mylist[-3:]
```

Out[637…
```
['six', 'seven', 'eight']
```

In [639…
```
mylist[-2:]
```

Out[639… ['seven', 'eight']

In [641…
```
mylist[-1]
```

Out[641… 'eight'

In [643…
```
mylist[:]
```

Out[643… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

Add,Remove & Changes items

In [646…
```
mylist
```

Out[646… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [648…
```
mylist.append('nine')
mylist
```

Out[648… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [650…
```
mylist.remove('nine')
mylist
```

Out[650… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [652…
```
mylist.insert(1,'ten')
mylist
```

Out[652… ['one', 'ten', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [654…
```
mylist.insert(1,'ONE')
mylist # add items at the index location
```

Out[654… ['one', 'ONE', 'ten', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [656…
```
mylist.remove('ONE')
mylist # Remmove items "ONE"
```

Out[656… ['one', 'ten', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [658…
```
mylist.pop()
mylist
```

Out[658… ['one', 'ten', 'two', 'three', 'four', 'five', 'six', 'seven']

In [660…
```
mylist.pop(8)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[660], line 1
----> 1 mylist.pop(8)

IndexError: pop index out of range
```

In [662…
```
del mylist[7]
mylist
```

Out[662…
```
['one', 'ten', 'two', 'three', 'four', 'five', 'six']
```

In [664…
```
mylist[0]=1
mylist[1]=2
mylist[2]=3
mylist
```

Out[664…
```
[1, 2, 3, 'three', 'four', 'five', 'six']
```

In [666…
```
mylist.clear()
mylist # empty list
```

Out[666…
```
[]
```

In [668…
```
del mylist #delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[668], line 2
      1 del mylist #delete the whole list
----> 2 mylist

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

copylist

In [671…
```
mylist=['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
mylist
```

Out[671…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [673…
```
mylist1=mylist
```

In [675…
```
id(mylist),id(mylist1)
```

Out[675…
```
(2412055469760, 2412055469760)
```

In [677…
```
mylist2 = mylist.copy()
```

In [679…
```
id(mylist2)
```

Out[679…
```
2412055439040
```

In [681…
```
mylist[0]=1
```

In [683…
```
mylist
```

Out[683…
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [685…
```
mylist1
```

Out[685… [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [687…
```
mylist2
```

Out[687… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

Join lists

In [690…
```
list1=['one','two','three','four']
list1
```

Out[690… ['one', 'two', 'three', 'four']

In [692…
```
list2=['five','six','seven','eight']
list2
```

Out[692… ['five', 'six', 'seven', 'eight']

In [694…
```
list1.extend(list2)
list1
```

Out[694… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

list membership

In [697…
```
list1
```

Out[697… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [699…
```
'one' in list1
```

Out[699… True

In [701…
```
'ten' in list1
```

Out[701… False

In [703…
```
if 'three' in list1:
    print('three is present in the list')
else:
    print('three is not present in the list')
```

three is present in the list

In [705…
```
if 'eleven' in list1:
    print('eleven is present in the list')
else:
    print('eleven is not present in the list')
```

eleven is not present in the list

REVERSE and sort list

In [708…
```
list1
```

Out[708…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [710…
```
list1.reverse()
list1
```

Out[710…
```
['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

In [712…
```
list1=list1[::-1]
list1
```

Out[712…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [714…
```
mylist3=[9,5,2,99,12,88,34]
mylist3.sort()
```

In [716…
```
mylist3
```

Out[716…
```
[2, 5, 9, 12, 34, 88, 99]
```

In [718…
```
mylist3=[9,5,2,99,12,88,34]
mylist3.sort(reverse=True)
mylist3
```

Out[718…
```
[99, 88, 34, 12, 9, 5, 2]
```

In [724…
```
mylist4=[88,65,33,21,11,98]
sorted(mylist4)
```

Out[724…
```
[11, 21, 33, 65, 88, 98]
```

In [726…
```
mylist4
```

Out[726…
```
[88, 65, 33, 21, 11, 98]
```

loop through a list

In [729…
```
list1
```

Out[729…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [731…
```
for i in list1:
    print(i)
```

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

In [733…
```
for i in enumerate (list1):
    print(i)
```

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

In [735…
```
list10=['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
list10
```

Out[735…
```
['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [737…
```
list10.count('one')
```

Out[737…
```
3
```

In [739…
```
list10.count('two')
```

Out[739…
```
2
```

All / Any

In [746…
```
l1=[1,2,3,4,0]
l1
```

Out[746…
```
[1, 2, 3, 4, 0]
```

In [748…
```
all(l1)
```

Out[748…
```
False
```

In [750…
```
any(l1)
```

Out[750…
```
True
```

In [754…
```
l2=[1,2,3,4,True,False]
```

In [756…
```
all(l2)
```

Out[756…
```
False
```

In [758…
```
any(l2)
```

Out[758…
```
True
```

In [ ]:

2 . TUPLE

tup1=() # empty tuple

In [5]:
```
tup2=(10,20,30)
tup2 #tuple of intiger number
```

Out[5]:
```
(10, 20, 30)
```

In [7]:
```
tup3=(10.5,23.5,38.5)
tup3 #tuple of float number
```

Out[7]:
```
(10.5, 23.5, 38.5)
```

In [9]:
```
tup4=('one','two','three')
tup4 # tuple of float number
```

Out[9]:
```
('one', 'two', 'three')
```

In [11]:
```
tup5=('Asif',24,(50,100),(150,90))
tup5 #nested tuple
```

Out[11]:
```
('Asif', 24, (50, 100), (150, 90))
```

In [13]:
```
tup6=(200,'Asif',17.98)
tup6 # tuple with mixed datatype
```

Out[13]:
```
(200, 'Asif', 17.98)
```

In [15]:
```
tup7=('Asif',35,[50,100],[150,90],{'John','David'},(99,22,33))
tup7
```

Out[15]:
```
('Asif', 35, [50, 100], [150, 90], {'David', 'John'}, (99, 22, 33))
```

In [17]:
```
len(tup7)
```

Out[17]:
```
6
```

Tuple indexing

In [20]:
```
tup2[0]
```

Out[20]:
```
10
```

In [22]:
```
tup4[0]
```

Out[22]:
```
'one'
```

In [24]:
```
tup4[0][0]
```

Out[24]:
```
'o'
```

In [26]:
```
tup4[-1]
```

Out[26]:
```
'three'
```

In [28]:
```
tup5[-1]
```

Out[28]:
```
(150, 90)
```

Tuple slicing

In [31]:
```
mytuple=('one','two','three','four','five','six','seven','eight')
mytuple
```

Out[31]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [33]:
```
mytuple[0:3]
```

Out[33]:
```
('one', 'two', 'three')
```

In [35]:
```
mytuple[2:5]
```

Out[35]:
```
('three', 'four', 'five')
```

In [37]:
```
mytuple[:3]
```

Out[37]:
```
('one', 'two', 'three')
```

In [39]:
```
mytuple[:2]
```

Out[39]:
```
('one', 'two')
```

In [41]:
```
mytuple[-3:]
```

Out[41]:
```
('six', 'seven', 'eight')
```

In [43]:
```
mytuple[-2:]
```

Out[43]:
```
('seven', 'eight')
```

In [45]:
```
mytuple[-1]
```

Out[45]:
```
'eight'
```

In [47]:
```
mytuple[:]
```

Out[47]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [49]:
```
mytuple
```

Out[49]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

Remove & Change items

In [52]:
```
mytuple
```

Out[52]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [54]:
```
del mytuple[0] #tuple are immutable which means we can't DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[54], line 1
----> 1 del mytuple[0]

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

In [ ]:
```
mytuple[0]=1 #tuple are immutable which means we can't CHANGE tuple items
```

In [ ]:
```
del mytuple #Deleting entire tuple object is possible
```

Loop through a tuple

In [59]:
```
mytuple
```

Out[59]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [61]:
```
for i in mytuple:
    print(i)
```

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

In [63]:
```
for i in enumerate (mytuple):
    print(i)
```

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

Tuple Membership

In [66]:
```
mytuple
```

Out[66]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [68]:
```
'one' in mytuple # check if 'one'exists in the list
```

Out[68]: True

In [70]:
```
'ten' in mytuple #check if 'ten' exists in the list
```

Out[70]: False

In [72]:
```
if 'three' in mytuple: # check if 'three' exists in the tuple
    print('Three is present in the tuple.')
else:
    print('Three is not present in the tuple.')
```

Three is present in the tuple.

In [74]:
```
if 'eleven' in mytuple:# check if 'eleven' exists in the tuple
    print('Eleven is present in the mytuple.')
else:
    print('Eleven is not present in mytuple.')
```

Eleven is not present in mytuple.

Index position

In [77]:
```
mytuple
```

Out[77]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

In [79]:
```
mytuple.index('one') # index of first element equal to'one'
```

Out[79]: 0

In [81]:
```
mytuple.index('five') #index of first element equal to 'five'
```

Out[81]: 4

In [83]:
```
mytuple1=('one','two','three','four','one','one','two','three')
```

In [85]:
```
mytuple1
```

Out[85]: ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')

In [87]:
```
mytuple1.index('three')# index 1st element equal to one
```

Out[87]: 2

In [89]:
```
mytuple1.index('three')# if repeatation element in tuple,index always print the
```

Out[89]: 2

Sortinng

In [92]:
```
mytuple2=(78,89,2,35,789,10,40)
mytuple2
```

Out[92]:
```
(78, 89, 2, 35, 789, 10, 40)
```

In [94]:
```
sorted(mytuple2)
```

Out[94]:
```
[2, 10, 35, 40, 78, 89, 789]
```

In [96]:
```
sorted(mytuple2,reverse=True)
```

Out[96]:
```
[789, 89, 78, 40, 35, 10, 2]
```

Tupple count

In [99]:
```
mytuple1
```

Out[99]:
```
('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

In [101…
```
mytuple1.count('one')
```

Out[101…
```
3
```

In [103…
```
mytuple1.count('two')
```

Out[103…
```
2
```

In [105…
```
mytuple1.count('three')
```

Out[105…
```
2
```

In [107…
```
mytuple.count('four')
```

Out[107…
```
1
```

In [ ]:
```

```

3.SET

In [772…
```
myset={1,2,3,4,5}
myset
```

Out[772…
```
{1, 2, 3, 4, 5}
```

In [774…
```
len(myset)
```

Out[774…
```
5
```

In [776…
```
my_set={1,1,2,2,3,4,5,5}
my_set
```

Out[776…
```
{1, 2, 3, 4, 5}
```

In [782…
```
myset1={1.79,2.8,3.99,4.56,5.45}
myset1
```

Out[782…
```
{1.79, 2.8, 3.99, 4.56, 5.45}
```

In [784…
```
myset2={'Asif','John','Tyrion'}
myset2
```

Out[784…
```
{'Asif', 'John', 'Tyrion'}
```

In [786…
```
myset3={10,20,"Hola",[11,22,32]}
myset3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[786], line 1
----> 1 myset3={10,20,"Hola",[11,22,32]}
      2 myset3

TypeError: unhashable type: 'list'
```

In [790…
```
myset4=set()
print(type(myset4))
```

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

In [794…
```
myset5=set(('one','two','three','four'))
myset5
```

Out[794…
```
{'four', 'one', 'three', 'two'}
```

In [912…
```
myset={'one','two','three','four','six','seven','eight'}
myset
```

Out[912…
```
{'eight', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [913…
```
for i in myset:
    print(i)
```

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

In [916…
```
for i in enumerate (myset):
    print(i)
```

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

set membership

In [919…
```
myset
```

Out[919…
```
{'eight', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [921…
```
'one' in myset
```

Out[921…
```
True
```

In [923…
```
'ten' in myset
```

Out[923…
```
False
```

In [925…
```
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 [927…
```
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
```

add & remove items

In [930…
```
myset
```

Out[930…
```
{'eight', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [932…
```
myset.add('NINE')
myset
```

Out[932…
```
{'NINE', 'eight', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [934…
```
myset.update(['TEN','ELEVEN','TWELVE'])
myset
```

Out[934…
```
{'ELEVEN',
 'NINE',
 'TEN',
 'TWELVE',
 'eight',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [936…
```
myset.remove('NINE')
myset
```

Out[936…
```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [938…
```
myset.discard('ten')
myset
```

Out[938…
```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [940…
```
myset.clear()
```

In [942…
```
myset
```

Out[942…
```
set()
```

In [944…
```
del myset
```

In [946…
```
myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[946], line 1
----> 1 myset

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

Copy set

In [949…
```
myset= {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
```

In [951…
```
myset
```

Out[951…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [953…
```
myset1=myset
```

In [955…
```
myset1
```

Out[955…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [957…
```
id(myset1) , id(myset)
```

Out[957…
```
(2412061004896, 2412061004896)
```

In [959…
```
myset2=myset.copy()
```

In [961…
```
myset2
```

Out[961…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [963…
```
id(myset2)
```

Out[963…
```
2412061010272
```

In [965…
```
myset.add('nine')
myset
```

Out[965…
```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [967…
```
myset1
```

Out[967…
```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

In [969…
```
myset2
```

Out[969…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

set operation

In [974…
```
a = {1,2,3,4,5}
b = {4,5,6,7,8}
c = {8,9,10}
```

In [976…
```
a|b
```

Out[976…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [978…
```
a.union(b)
```

Out[978…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [980…
```
a.union(b,c)
```

Out[980…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [982…
```
a.update(b,c)
```

In [984…
```
a
```

Out[984… {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [986…
```
a = {1,2,3,4,5}
b = {4,5,6,7,8}
```

In [988…
```
a&b
```

Out[988… {4, 5}

In [990…
```
a.intersection(b)
```

Out[990… {4, 5}

In [992…
```
a.intersection_update(b)
```

In [994…
```
a
```

Out[994… {4, 5}

In [996…
```
a = {1,2,3,4,5}
b = {4,5,6,7,8}
```

In [998…
```
a-b
```

Out[998… {1, 2, 3}

In [100…
```
a.difference(b)
```

Out[100… {1, 2, 3}

In [100…
```
b.difference(a)
```

Out[100… {6, 7, 8}

In [100…
```
b.difference_update(a)
```

In [100…
```
b
```

Out[100… {6, 7, 8}

In [100…
```
a = {1,2,3,4,5}
b = {4,5,6,7,8}
```

In [101…
```
a^b
```

Out[101… {1, 2, 3, 6, 7, 8}

In [101…
```
a.symmetric_difference(b)
```

Out[101… {1, 2, 3, 6, 7, 8}

In [101…
```
a.symmetric_difference_update(b)
```

In [101…
```
a
```

Out[101… {1, 2, 3, 6, 7, 8}

In [102…
```
a = {1,2,3,4,5,6,7,8,9}
b = {3,4,5,6,7,8}
c = {10,20,30,40}
```

In [102…
```
b.issubset(a)
```

Out[102… True

In [102…
```
b.issuperset(a)
```

Out[102… False

In [103…
```
a.issuperset(b)
```

Out[103… True

In [103…
```
a.isdisjoint(b)
```

Out[103… False

In [104…
```
a.isdisjoint(c)
```

Out[104… True

In [104…
```
b.isdisjoint(c)
```

Out[104… True

other built in function

In [107…
```
a
```

Out[107… {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [107…
```
sum(a)
```

Out[107… 45

In [107…
```
max(a)
```

Out[107… 9

In [108…
```
min(a)
```

Out[108… 1

In [108…
```
list(enumerate(a))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[1082], line 1
----> 1 list(enumerate(a))

TypeError: 'list' object is not callable
```

In [108…
```
d= sorted(a,reverse=True)
d
```

Out[108…
```
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [108…
```
sorted(a)
```

Out[108…
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [109…
```
sorted(d)
```

Out[109…
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

4.Dictionary

In [114…
```
d=dict()
d
```

Out[114…
```
{}
```

In [114…
```
print(type(d))
```

```
<class 'dict'>
```

In [115…
```
d={}
d
```

Out[115…
```
{}
```

In [115…
```
d={1:'one',2:'two',3:'three'}
d
```

Out[115…
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [115…
```
d=dict({1:'one',2:'two',3:'three'})
```

In [115…
```
d
```

Out[115…
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [116…
```
d1={'A':'one' , 'B':'two' , 'C':'three'}

d1
```

Out[116…
```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

In [116…
```
d2={1:'one' , 'A':'two' , 3:'three'}
d2
```

Out[116… {1: 'one', 'A': 'two', 3: 'three'}

In [116…
```
d2.keys()
```

Out[116… dict_keys([1, 'A', 3])

In [116…
```
d2.values()
```

Out[116… dict_values(['one', 'two', 'three'])

In [116…
```
d2.items()
```

Out[116… dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])

In [117…
```
d4={1:'one' , 2:'two' , 'A':['asif' , 'john' , 'Maria'], 'B':('Bat','cat' ,'hat'
d4
```

Out[117…
```
{1: 'one',
 2: 'two',
 'A': ['asif', 'john', 'Maria'],
 'B': ('Bat', 'cat', 'hat')}
```

In [117…
```
keys={'a','b','c','d'}
d5=dict.fromkeys(keys)
d5
```

Out[117… {'d': None, 'b': None, 'c': None, 'a': None}

In [117…
```
keys={'a','b','c','d'}
values=10
d6=dict.fromkeys(keys,values)
d6
```

Out[117… {'d': 10, 'b': 10, 'c': 10, 'a': 10}

In [117…
```
keys={'a','b','c','d'}
value=[10,20,30]
d6=dict.fromkeys(keys,value)
d6
```

Out[117… {'d': [10, 20, 30], 'b': [10, 20, 30], 'c': [10, 20, 30], 'a': [10, 20, 30]}

In [118…
```
value.append(40)
d6
```

Out[118…
```
{'d': [10, 20, 30, 40],
 'b': [10, 20, 30, 40],
 'c': [10, 20, 30, 40],
 'a': [10, 20, 30, 40]}
```

Accessing items

In [120…
```
d1
```

Out[120… {'A': 'one', 'B': 'two', 'C': 'three'}

In [120…
```
d1['A']
```

Out[120…
```
'one'
```

In [120…
```
d1.get('A')
```

Out[120…
```
'one'
```

In [121…
```
d7= {'Name':'Asif' , 'ID': 74123 , 'DOB': 1991 , 'job' :'Analyst'}
d7
```

Out[121…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}
```

In [121…
```
d7['Name']
```

Out[121…
```
'Asif'
```

In [121…
```
d7.get('job')
```

Out[121…
```
'Analyst'
```

add,remove & change items

In [122…
```
d7
```

Out[122…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}
```

In [122…
```
d7['DOB']=1992
d7['job']='developer'
d7
```

Out[122…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1992, 'job': 'developer'}
```

In [123…
```
d8={'DOB':1995}
d7.update(d8)
d7
```

Out[123…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1995, 'job': 'developer'}
```

In [123…
```
d7['Address']='hydrabad'
d7
```

Out[123…
```
{'Name': 'Asif',
 'ID': 74123,
 'DOB': 1995,
 'job': 'developer',
 'Address': 'hydrabad'}
```

In [123…
```
d7.pop('job')
d7
```

Out[123…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1995, 'Address': 'hydrabad'}
```

In [123…
```
del[d7['Address']]
```

```
d7
```

Out[123…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1995}
```

In [124…
```
d7.clear()
d7
```

Out[124…
```
{}
```

In [124…
```
del d7
d7
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1247], line 1
----> 1 del d7
      2 d7

NameError: name 'd7' is not defined
```

copy dictionary

In [125…
```
d7= {'Name':'Asif' , 'ID': 74123 , 'DOB': 1991 , 'job' :'Analyst'}
d7
```

Out[125…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}
```

In [125…
```
d8=d7
```

In [125…
```
id(d7),id(d8)
```

Out[125…
```
(2412054465600, 2412054465600)
```

In [125…
```
d9=d7.copy()
```

In [126…
```
d9
```

Out[126…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}
```

In [126…
```
id(d9)
```

Out[126…
```
2412060487040
```

In [126…
```
d7['job']='developer'
d7
```

Out[126…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
```

In [126…
```
d8
```

Out[126…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
```

In [126…
```
d9
```

Out[126… {'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}

loop through dictionary

In [127…
```
d8={'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
d8
```

Out[127… {'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}

In [127…
```
for i in d8:
    print(d8)
```

```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
```

In [128…
```
for i in d8:
    print(d8[i])
```

```
Asif
74123
1991
developer
```

In [128…
```
for i in d8:
    print(i,':',d8[i])
```

```
Name : Asif
ID : 74123
DOB : 1991
job : developer
```

Dictionary membership

In [129…
```
d7
```

Out[129… {'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}

In [129…
```
'Name' in d7
```

Out[129… True

In [129…
```
'Asif' in d7
```

Out[129… False

In [129…
```
'ID' in d7
```

Out[129… True

In [130…
```
'Address' in d7
```

Out[130… False

Any & All

In [130…
```
d7
```

Out[130…
```
{'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'developer'}
```

In [132…
```
d7[0]='mark'
d7
```

Out[132…
```
{'Name': 'Asif',
 'ID': 74123,
 'DOB': 1991,
 'job': 'developer',
 'mark': 0,
 '0': 'mark',
 0: 'mark'}
```

In [132…
```
any(d7)
```

Out[132…
```
True
```

In [132…
```
all(d7)
```

Out[132…
```
False
```

In [ ]:

4 type of data structure is completed.

In [ ]: