LIST

In [365…
```
#Creation of list
```

In [366…
```
list = []
```

In [367…
```
print(type(list))
```

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

In [368…
```
list1 = [10, 20, 30, 40, 50, 60, 70]
```

In [369…
```
len(list1)
```

Out[369… 7

In [370…
```
#Indexing
```

In [371…
```
list1[3]
```

Out[371… 40

In [372…
```
list1[2]
```

Out[372… 30

In [373…
```
list1[-1]
```

Out[373… 70

In [374…
```
#Slicing
```

In [375…
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine'
```

In [376…
```
mylist[0:6]
```

Out[376… ['one', 'two', 'three', 'four', 'five', 'six']

In [377…
```
mylist[:5]
```

Out[377… ['one', 'two', 'three', 'four', 'five']

In [378…
```
mylist[7:]
```

Out[378… ['eight', 'nine']

In [379…
```
mylist[:-1]
```

Out[379… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [380…
```
mylist[-1:]
```

Out[380… ['nine']

In [381…
```
mylist[:]
```

Out[381… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [382…
```
#Append
```

In [383…
```
mylist.append('ten')
```

In [384…
```
mylist
```

Out[384… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [385…
```
mylist.append('one')
```

In [386…
```
mylist
```

Out[386…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [387…
```
#Count
```

In [388…
```
mylist.count('one')
```

Out[388… 2

In [389…
```
mylist
```

Out[389…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [390…
```
#Copy
```

In [391…
```
mylist1 = mylist.copy()
```

In [392…
```
mylist1
```

Out[392…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [393…
```
mylist
```

Out[393…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [394…
```
#Append
```

In [395…
```
mylist1
```

Out[395…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [396…
```
mylist1.append('one')
```

In [397…
```
mylist1
```

Out[397…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one',
 'one']
```

In [398…
```
#Pop
```

In [399…
```
mylist1.pop()
```

Out[399…
```
'one'
```

In [400…
```
#Remove
```

In [401…
```
mylist1.remove('one')
```

In [402…
```
mylist1
```

Out[402…
```
['two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten', 'one']
```

In [403…
```
mylist
```

Out[403…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [404…
```
#Clear
```

In [405…
```
mylist.clear()
```

In [406…
```
mylist
```

Out[406…
```
[]
```

In [407…
```
#Indexing
```

In [408…
```
mylist1
```

Out[408…
```
['two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten', 'one']
```

In [409…
```
mylist1.index('nine')
```

Out[409…
```
7
```

In [410…
```
#Extend
```

In [411…
```
mylist = [2,3,4,5,6,7,8,9]
```

In [412…
```
mylist
```

Out[412…
```
[2, 3, 4, 5, 6, 7, 8, 9]
```

In [413…
```
mylist.extend(mylist1)
```

In [414…
```
mylist
```

Out[414…
```
[2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [415…
```
#Insert in list
```

In [416…
```
mylist1.insert(0, 'one')
```

In [417…
```
mylist1
```

Out[417…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'one']
```

In [418…
```
mylist1.insert(10, '11')
```

In [419…
```
mylist1
```

Out[419…
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 '11',
 'one']
```

In [420…
```
mylist2 = [100, 200, 3, 80, 94, 12, 56, 45, 33, 70]
mylist2
```

Out[420… [100, 200, 3, 80, 94, 12, 56, 45, 33, 70]

In [421…
```
#List membership
```

In [422…
```
list = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [423…
```
'nine' in list
```

Out[423… True

In [424…
```
'ten' in list
```

Out[424… False

In [425…
```
if 'seven' in list:
    print('Seven is present in the list')
else:
    print('Seven is not present in the list')
```

Seven is present in the list

In [426…
```
if 'Ten' in list:
    print('Ten is present in the list')
else:
    print('Ten is not present in the list')
```

Ten is not present in the list

In [427…
```
# Reverse & Sort in List
```

In [428…
```
mylist2.reverse()
```

In [429…
```
mylist2
```

Out[429… [70, 33, 45, 56, 12, 94, 80, 3, 200, 100]

In [430…
```
mylist2.sort()
mylist2
```

Out[430… [3, 12, 33, 45, 56, 70, 80, 94, 100, 200]

In [431…
```
mylist2.sort(reverse = True)
mylist2
```

Out[431… [200, 100, 94, 80, 70, 56, 45, 33, 12, 3]

In [432…
```
mylist2.sort(reverse = False)
mylist2
```

Out[432… [3, 12, 33, 45, 56, 70, 80, 94, 100, 200]

In [433…
```
mylist1.sort()
mylist1
```

Out[433…
```
['11',
 'eight',
 'five',
 'four',
 'nine',
 'one',
 'one',
 'seven',
 'six',
 'ten',
 'three',
 'two']
```

In [434…
```
mylist1.sort(reverse = True)
mylist1
```

Out[434…
```
['two',
 'three',
 'ten',
 'six',
 'seven',
 'one',
 'one',
 'nine',
 'four',
 'five',
 'eight',
 '11']
```

In [435…
```
# Loop through a list
```

In [436…
```
for i in mylist1:
    print(i)
```

```
two
three
ten
six
seven
one
one
nine
four
five
eight
11
```

In [437…
```
for i in enumerate(mylist1):
    print(i)
```

```
(0, 'two')
(1, 'three')
(2, 'ten')
(3, 'six')
(4, 'seven')
(5, 'one')
(6, 'one')
(7, 'nine')
(8, 'four')
(9, 'five')
(10, 'eight')
(11, '11')
```

In [438…
```
# All / Any
```

In [439…
```
L1 = [9,8,7,6,5,4]
all(L1)
```

Out[439… True

In [440…
```
any(L1)
```

Out[440… True

In [441…
```
L2 = [1 , 2 , 3 , 4, True]
all(L2)
```

Out[441… True

In [442…
```
L2 = [1 , 2 , 3 , 4 , False]
all(L2)
```

Out[442… False

In [443…
```
L3 = [1 , 2 , 3 , 4, True]
any(L3)
```

Out[443… True

In [444…
```
L4 = [False]
any(L4)
```

Out[444… False

In [445…
```
L5 = [1, 2 , 3, 4, False]
any(L5)
```

Out[445… True

TUPLE

In [446…
```
#Tuple creation
```

In [447…
```
tup1 = ()
tup1
```

Out[447… ()

In [448…
```
tup2 = (10, 20.00, 'three', (1,8),(5,4))
tup2
```

Out[448… (10, 20.0, 'three', (1, 8), (5, 4))

In [449…
```
len(tup2)
```

Out[449… 5

In [450…
```
#Tuple indexing
```

In [451…
```
tup2[3]
```

Out[451… (1, 8)

In [452…
```
tup2[1]
```

Out[452… 20.0

In [453…
```
tup2[2]
```

Out[453… 'three'

In [454…
```
#Tuple slicing
```

In [455…
```
tup2[0:3]
```

Out[455… (10, 20.0, 'three')

In [456…
```
tup2[:3]
```

Out[456… (10, 20.0, 'three')

In [457…
```
tup2[-3:]
```

Out[457… ('three', (1, 8), (5, 4))

In [458…
```
tup2[:]
```

Out[458… (10, 20.0, 'three', (1, 8), (5, 4))

In [459…
```
# Remove, Delete and Change items in tuple
```

In [460…
```
tup = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', '
tup
```

Out[460… ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')

In [461…
```
del tup[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[461], line 1
----> 1 del tup[0]

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

In [ ]:
```
tup[0] = 1
```

In [462…
```
del tup
```

In [463…
```
#Loop through a tuple
```

In [464…
```
for i in tup:
    print(i)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[464], line 1
----> 1 for i in tup:
      2     print(i)

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

In [472…
```
for i in enumerate(tup):
    print(i)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[472], line 1
----> 1 for i in enumerate(tup):
      2     print(i)

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

In [473…
```
#Count
```

In [474…
```
mytup = ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

In [475…
```
mytup.count('one')
```

Out[475… 3

In [476…
```
mytup.count('two')
```

Out[476… 2

In [477…
```
mytup.count('four')
```

Out[477… 1

In [478…
```
#Tuple membership
```

In [479…
```
mytup = ('one', 'two', 'three', 'four', 'five', 'six', 'seven')
```

In [480…
```
'one' in mytup
```

Out[480… True

In [481…
```
'nine' in mytup
```

Out[481… False

In [482…
```
if 'seven' in mytup:
    print('Seven is present in the tuple')
else:
    print('Seven is not present in the tuple')
```

```
Seven is present in the tuple
```

In [483…
```
if 'Nine' in mytup:
    print('Nine is present in the tuple')
else:
    print('Nine is not present in the tuple')
```

```
Nine is not present in the tuple
```

In [484…
```
#Index position
```

In [485…
```
tup = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', '
tup
```

Out[485… ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')

In [486…
```
tup.index('seven')
```

Out[486… 6

In [487…
```
tup[6]
```

Out[487… 'seven'

In [488…
```
#Sorting
```

In [489…
```
tup = (9, 72, 81, 18, 27, 54, 63, 36, 45, 90)
tup
```

Out[489… (9, 72, 81, 18, 27, 54, 63, 36, 45, 90)

In [490…
```
sorted(tup)
```

Out[490… [9, 18, 27, 36, 45, 54, 63, 72, 81, 90]

In [491…
```
sorted(tup, reverse = True)
```

Out[491… [90, 81, 72, 63, 54, 45, 36, 27, 18, 9]

SET

In [492…
```
#Set Creation
```

In [493…
```
myset = {1,2,3,4,5}
myset
```

Out[493…
```
{1, 2, 3, 4, 5}
```

In [494…
```
len(myset)
```

Out[494…
```
5
```

In [495…
```
myset1 = {'Asif' , 'John' , 'Tyrion'}
myset1
```

Out[495…
```
{'Asif', 'John', 'Tyrion'}
```

In [496…
```
#Loop through a Set
```

In [497…
```
myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
for i in myset:
    print(i)
```

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

In [498…
```
for i in enumerate(myset):
   print(i)
```

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

In [499…
```
#Set Membership
```

In [500…
```
myset
```

Out[500…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [501…
```
'one' in myset
```

Out[501…
```
True
```

In [502…
```
'ten' in myset
```

Out[502…
```
False
```

In [503…
```
if 'seven' in myset:
    print('Seven is present in the set')
```

```
else:
    print('Seven is not present in the set')
```

```
Seven is present in the set
```

In [504…
```
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 [505…
```
#Add, Update, Discard, Clear, Delete & Remove Items
```

In [506…
```
myset
```

Out[506…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [507…
```
myset.add('NINE')
myset
```

Out[507…
```
{'NINE', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [508…
```
myset.update(['TEN' , 'ELEVEN' , 'TWELVE'])
myset
```

Out[508…
```
{'ELEVEN',
 'NINE',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [509…
```
myset.remove('NINE')
myset
```

Out[509…
```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [510…
```
myset.discard('TEN')
myset
```

Out[510…
```
{'ELEVEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

In [511…
```
myset.clear()
myset
```

Out[511…
```
set()
```

In [512…
```
del myset
myset
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[512], line 2
      1 del myset
----> 2 myset

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

In [513…
```
#Copy Set
```

In [514…
```
myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}
myset
```

Out[514…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [515…
```
myset1 = myset
myset1
```

Out[515…
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [516…
```
#Set Operation
```

In [517…
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [518…
```
A.union(B)
```

Out[518…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [519…
```
A.union(B, C)
```

Out[519…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [520…
```
A | B
```

Out[520…
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [521…
```
A.intersection(B)
```
Out[521… {4, 5}

In [522…
```
A & B
```
Out[522… {4, 5}

In [523…
```
A.difference(B)
```
Out[523… {1, 2, 3}

In [524…
```
A - B
```
Out[524… {1, 2, 3}

In [525…
```
B - A
```
Out[525… {6, 7, 8}

In [526…
```
A.symmetric_difference(B)
```
Out[526… {1, 2, 3, 6, 7, 8}

In [527…
```
A
```
Out[527… {1, 2, 3, 4, 5}

In [528…
```
B
```
Out[528… {4, 5, 6, 7, 8}

In [529…
```
A ^ B
```
Out[529… {1, 2, 3, 6, 7, 8}

In [530…
```
#Subset , Superset & Disjoint
```

In [531…
```
A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

In [532…
```
B.issubset(A)
```
Out[532… True

In [533…
```
A.issuperset(B)
```
Out[533… True

In [534…
```
C.isdisjoint(A)
```
Out[534… True

In [535…
```
#Other Built-in functions
```

In [536…
```
A
```
Out[536… {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [537…
```
sum(A)
```
Out[537… 45

In [538…
```
max(A)
```
Out[538… 9

In [539…
```
min(A)
```
Out[539… 1

In [540…
```
len(A)
```
Out[540… 9

In [541…
```
list(enumerate(A))
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[541], line 1
----> 1 list(enumerate(A))

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

In [542…
```
D= sorted(A,reverse=True)
D
```
Out[542… [9, 8, 7, 6, 5, 4, 3, 2, 1]

In [543…
```
sorted(D)
```
Out[543… [1, 2, 3, 4, 5, 6, 7, 8, 9]

DICTIONARY

In [544…
```
#Create Dictionary
```

In [545…
```
mydict = dict()
mydict
```
Out[545… {}

In [546…
```
mydict = {}
mydict
```
Out[546… {}

In [547…
```
mydict = {1:'one' , 2:'two' , 3:'three'}
```

In [548…
```
mydict.keys()
```

Out[548…
```
dict_keys([1, 2, 3])
```

In [549…
```
mydict.values()
```

Out[549…
```
dict_values(['one', 'two', 'three'])
```

In [550…
```
mydict.items()
```

Out[550…
```
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

In [551…
```
# Create a dictionary from a sequence of keys
```

In [552…
```
keys = {'a' , 'b' , 'c' , 'd'}
mydict3 = dict.fromkeys(keys)
mydict3
```

Out[552…
```
{'d': None, 'a': None, 'b': None, 'c': None}
```

In [553…
```
keys = {'a' , 'b' , 'c' , 'd'}
value = 10
mydict3 = dict.fromkeys(keys , value)
mydict3
```

Out[553…
```
{'d': 10, 'a': 10, 'b': 10, 'c': 10}
```

In [554…
```
keys = {'a' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict3 = dict.fromkeys(keys , value)
mydict3
```

Out[554…
```
{'d': [10, 20, 30], 'a': [10, 20, 30], 'b': [10, 20, 30], 'c': [10, 20, 30]}
```

In [555…
```
value.append(40) #append fuction is used to add another value into dict
mydict3
```

Out[555…
```
{'d': [10, 20, 30, 40],
 'a': [10, 20, 30, 40],
 'b': [10, 20, 30, 40],
 'c': [10, 20, 30, 40]}
```

In [556…
```
#Accessing Items
```

In [557…
```
mydict = {1:'one' , 2:'two' , 3:'three' , 4:'four'}
mydict
```

Out[557…
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

In [558…
```
mydict[1]
```

Out[558…
```
'one'
```

In [559…
```
# Access item using get() method
```

In [560…
```
mydict.get(1)
```

Out[560… 'one'

In [561…
```
mydict1 = {'Name':'Asif' , 'ID': 74123 , 'DOB': 1991 , 'job' :'Analyst'}
mydict1
```

Out[561… {'Name': 'Asif', 'ID': 74123, 'DOB': 1991, 'job': 'Analyst'}

In [562…
```
mydict1['Name']
```

Out[562… 'Asif'

In [563…
```
# Change, Update, Add, Remove, Delete, Pop, Popitem & Clear Items
```

In [564…
```
mydict1 = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Address' : 'Hilsinki'}
mydict1
```

Out[564… {'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}

In [565…
```
mydict1['DOB'] = 1992
mydict1['Address'] = Delhi
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[565], line 2
      1 mydict1['DOB'] = 1992
----> 2 mydict1['Address'] = Delhi
      3 mydict1

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

In [566…
```
dict1 = {'DOB':1995}
mydict1.update(dict1)
mydict1
```

Out[566… {'Name': 'Asif', 'ID': 12345, 'DOB': 1995, 'Address': 'Hilsinki'}

In [567…
```
mydict1['Job'] = Analyst
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[567], line 1
----> 1 mydict1['Job'] = Analyst
      2 mydict1

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

In [568…
```
mydict1.pop('Job')
mydict1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[568], line 1
----> 1 mydict1.pop('Job')
      2 mydict1

KeyError: 'Job'
```

In [569…
```
mydict1.popitem()
```

Out[569…
```
('Address', 'Hilsinki')
```

In [570…
```
mydict1
```

Out[570…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1995}
```

In [571…
```
del[mydict1['ID']]
mydict1
```

Out[571…
```
{'Name': 'Asif', 'DOB': 1995}
```

In [572…
```
mydict1.clear()
mydict1
```

Out[572…
```
{}
```

In [573…
```
del mydict1
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[573], line 2
      1 del mydict1
----> 2 mydict1

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

In [574…
```
#Copy Dictionary
```

In [575…
```
mydict = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Address' : 'Hilsinki'}
mydict
```

Out[575…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

In [576…
```
mydict1 = mydict
mydict1
```

Out[576…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

In [577…
```
mydict2 = mydict.copy()
mydict2
```

Out[577…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

In [578…
```
# Loop through a Dictionary
```

In [579…
```
mydict1 = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Address' : 'Hilsinki' ,
mydict1
```

Out[579…
```
{'Name': 'Asif',
 'ID': 12345,
 'DOB': 1991,
 'Address': 'Hilsinki',
 'Job': 'Analyst'}
```

In [580…
```
for i in mydict1:
    print(i , ':' , mydict1[i])
```

```
Name : Asif
ID : 12345
DOB : 1991
Address : Hilsinki
Job : Analyst
```

In [581…
```
for i in mydict1:
    print(mydict1[i])
```

```
Asif
12345
1991
Hilsinki
Analyst
```

In [582…
```
#Dictionary Membership
```

In [583…
```
mydict1 = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Job': 'Analyst'}
mydict1
```

Out[583…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Job': 'Analyst'}
```

In [584…
```
'Name' in mydict1
```

Out[584…
```
True
```

In [585…
```
'Asif' in mydict1
```

Out[585…
```
False
```

In [586…
```
'ID' in mydict1
```

Out[586…
```
True
```

In [587…
```
'Address' in mydict1
```

Out[587…
```
False
```

In [588…
```
#All / Any
```

In [589…
```
mydict1 = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Job': 'Analyst'}
mydict1
```

Out[589…
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Job': 'Analyst'}
```

In [590…
```
all(mydict1)
```

Out[590…
```
True
```

In [592…
```
any(mydict1)
```

Out[592…
```
True
```