Dictionary

->Dictionary is a mutable data type in Python. ->A Python dictionary is a collection of key and pairs separated by a colon (:) and enclosed in curly braces {}. ->Keys must be unique in a dictionary. Duplicate values are allowed.

Create Dictionary

```
In [8]:  md = dict() #empty dictionary
         md

Out[8]:  {}

In [10]: md = {}
         md

Out[10]: {}

In [12]: md = {8:'eight', 9:'nine', 10:'ten'} #dictionary with integer keys
         md

Out[12]: {8: 'eight', 9: 'nine', 10: 'ten'}

In [14]: md = dict({8:'eight', 9:'nine', 10:'ten'}) #create dictionary using dict()
         md

Out[14]: {8: 'eight', 9: 'nine', 10: 'ten'}

In [16]: md = {'A':'one', 'B':'two','C':'three'} #dictionary with character keys
         3md

Out[16]: {'A': 'one', 'B': 'two', 'C': 'three'}

In [18]: md = {1:'A',2:'B',3:'C'} #dictionary with mixed keys
         md

Out[18]: {1: 'A', 2: 'B', 3: 'C'}

In [20]: md.keys()    #Returns dictionary keys

Out[20]: dict_keys([1, 2, 3])

In [22]: md.values() #Returns dictionary values

Out[22]: dict_values(['A', 'B', 'C'])

In [24]: md.items() #Access each key-value pair within a dictionary
```

```
Out[24]: dict_items([(1, 'A'), (2, 'B'), (3, 'C')])
```

```
In [30]: #passing a list and tupel as value in dictionary
         md1 = {1:'one', 2:'two',3:'three','A':['abc','Sun'],'B':('bat','cat')}
         md1
```

```
Out[30]: {1: 'one', 2: 'two', 3: 'three', 'A': ['abc', 'Sun'], 'B': ('bat', 'cat')}
```

```
In [32]: md2 = {1:'one', 2:'two','A':{'name':'Rani', 'Age': 20},'B':('bat','cat')}
         md2
```

```
Out[32]: {1: 'one', 2: 'two', 'A': {'name': 'Rani', 'Age': 20}, 'B': ('bat', 'cat')}
```

```
In [36]: keys = {2,3,4,5}
         md3 = dict.fromkeys(keys) #Create a dictionary from a sequence of keys
         md3
```

```
Out[36]: {2: None, 3: None, 4: None, 5: None}
```

```
In [44]: keys = {'a','d','b','c'}
         value = 50
         md4 = dict.fromkeys(keys,value)
         md4
```

```
Out[44]: {'a': 50, 'd': 50, 'b': 50, 'c': 50}
```

```
In [58]: keys = {'a','d','b','c'}
         value = [40,80,60]
         md5 = dict.fromkeys(keys,value)
         md5
```

```
Out[58]: {'a': [40, 80, 60], 'd': [40, 80, 60], 'b': [40, 80, 60], 'c': [40, 80, 60]}
```

```
In [60]: value.append(10)
         md5
```

```
Out[60]: {'a': [40, 80, 60, 10],
          'd': [40, 80, 60, 10],
          'b': [40, 80, 60, 10],
          'c': [40, 80, 60, 10]}
```

Accessing Items

```
In [71]: md
```

```
Out[71]: {1: 'A', 2: 'B', 3: 'C'}
```

```
In [73]: md[1] #Access item using key
```

```
Out[73]: 'A'
```

In [75]:
```
md.get(1) #Access item using get() method
```
Out[75]:
```
'A'
```

In [77]:
```
mydict1 = {'Name':'abc', 'ID': 324, 'DOB': 1999, 'job': 'Analyst'}
mydict1
```
Out[77]:
```
{'Name': 'abc', 'ID': 324, 'DOB': 1999, 'job': 'Analyst'}
```

In [79]:
```
mydict1['Name']
```
Out[79]:
```
'abc'
```

In [83]:
```
mydict1.get('DOB')
```
Out[83]:
```
1999
```

Add, Remove and Change Items

In [86]:
```
mydict1
```
Out[86]:
```
{'Name': 'abc', 'ID': 324, 'DOB': 1999, 'job': 'Analyst'}
```

In [88]:
```
mydict1['DOB'] = 1996
mydict1['job'] = 'Scientist'
mydict1
```
Out[88]:
```
{'Name': 'abc', 'ID': 324, 'DOB': 1996, 'job': 'Scientist'}
```

In [92]:
```
dict1 = {'ID' : 987}
mydict1.update(dict1)
mydict1
```
Out[92]:
```
{'Name': 'abc', 'ID': 987, 'DOB': 1996, 'job': 'Scientist'}
```

In [94]:
```
mydict1['Address'] = 'Delhi'
mydict1
```
Out[94]:
```
{'Name': 'abc', 'ID': 987, 'DOB': 1996, 'job': 'Scientist', 'Address': 'Delhi'}
```

In [99]:
```
mydict1.pop('job') #Removing items in the dictionary using pop method
```
Out[99]:
```
'Scientist'
```

In [101…
```
mydict1
```
Out[101…
```
{'Name': 'abc', 'ID': 987, 'DOB': 1996, 'Address': 'Delhi'}
```

In [105…
```
mydict1.popitem()
mydict1
```

```
Out[105…  {'Name': 'abc', 'ID': 987}
```

```
In [107…  mydict1['Job'] = 'Analyst'
          mydict1['Address'] = 'Hyderabad'
          mydict1
```

```
Out[107…  {'Name': 'abc', 'ID': 987, 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

```
In [109…  del(mydict1['ID']) #removing item using del method
          mydict1
```

```
Out[109…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

```
In [133…  m = mydict1.copy()
```

```
In [135…  m
```

```
Out[135…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

```
In [137…  mydict1
```

```
Out[137…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

```
In [141…  m.clear()
```

```
In [143…  m
```

```
Out[143…  {}
```

```
In [145…  del m #Delete the dictionary object
```

```
In [147…  m
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[147], line 1
----> 1 m

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

Copy Dictionary

```
In [152…  mydict1
```

```
Out[152…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

```
In [156…  m = mydict1 #Create a new reference "mydict"
```

```
In [158…  id(m), id(mydict1)
```

Out[158…

```
(1356060892928, 1356060892928)
```

In [160…

```
mydict2 = m.copy()
```

In [162…

```
mydict2
```

Out[162…

```
{'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

In [165…

```
id(mydict2)
```

Out[165…

```
1356095149056
```

In [167…

```
mydict1['Address'] = 'Mumbai'
mydict1
```

Out[167…

```
{'Name': 'abc', 'Job': 'Analyst', 'Address': 'Mumbai'}
```

In [171…

```
m #m  will also be impacted as it is pointing to the same dictionary
```

Out[171…

```
{'Name': 'abc', 'Job': 'Analyst', 'Address': 'Mumbai'}
```

In [173…

```
mydict2
```

Out[173…

```
{'Name': 'abc', 'Job': 'Analyst', 'Address': 'Hyderabad'}
```

Looping through a dictionary

In [176…

```
mydict1
```

Out[176…

```
{'Name': 'abc', 'Job': 'Analyst', 'Address': 'Mumbai'}
```

In [182…

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

```
Name : abc
Job : Analyst
Address : Mumbai
```

In [184…

```
for i in mydict1:
    print(mydict1[i])
```

```
abc
Analyst
Mumbai
```

Dictionary Membership

In [187…

```
mydict1
```

```
Out[187…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Mumbai'}
```

```
In [189…  'Job' in mydict1 #Test if a key is in a dictionary or not.
```

```
Out[189…  True
```

```
In [193…  'Mumbai' in mydict1 #Membership test can be only done for keys.
```

```
Out[193…  False
```

```
In [195…  'Name' in mydict1
```

```
Out[195…  True
```

```
In [197…  'ID' in mydict1
```

```
Out[197…  False
```

All/Any

- The all() method returns:
- . True - If all all keys of the dictionary are True
- . False - If all all keys of the dictionary are False
- The any() method returns
- . True if any key of the dictionary is True.
- . If not, any() returns False.

```
In [202…  mydict1
```

```
Out[202…  {'Name': 'abc', 'Job': 'Analyst', 'Address': 'Mumbai'}
```

```
In [204…  all(mydict1)
```

```
Out[204…  True
```

```
In [206…  any(mydict1)
```

```
Out[206…  True
```

```
In [210…  r = {0,True}
          r
```

```
Out[210…  {0, True}
```

```
In [212…  all(r)
```

```
Out[212…  False
```

In [214…

```
any(r)
```

Out[214…

```
True
```

In [ ]:

```
```