Dictionary {}

- Mutable : Dictionary is a mutable data structure in python.
- Duplicates : key must be unique,you can have duplicates in values.
- Order : Dictionary follows insertion order.
- Heterogeneous : A dictionary can store different types of keys and values, like integer,string,lists,or even another dictionary.
- we have use to key and values pair to store values in dictionary.
- keys in dictionary acts like index values.
- A python dictionary is a collection of key and value pairsseprated by colon(:) & enclosed in curly braces {}.

Create Dictionary

In [2]:
```
d = dict() # empty dictionary
d
```

Out[2]: {}

In [3]:
```
d = dict() # empty dictionary
d
```

Out[3]: {}

In [4]:
```
d = {1:'one',2:'two',3:'three',4:'four',5:'five'} # dictionary with integer keys
d
```

Out[4]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

In [5]:
```
d = dict({1:'one',2:'two',3:'three',4:'four',5:'five'}) # create dictionaty usin
d
```

Out[5]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

In [6]:
```
d = {1:'one','B':'two',3:'three','C':'four',5:'five'} # dictionary with mixed ke
d
```

Out[6]: {1: 'one', 'B': 'two', 3: 'three', 'C': 'four', 5: 'five'}

In [7]:
```
d = {1:'one','B':'two',3:'three','C':'four',5:'five'} # dictionary with mixed ke
d
```

Out[7]: {1: 'one', 'B': 'two', 3: 'three', 'C': 'four', 5: 'five'}

In [6]:
```
d = {'A':'one','B':'two','c':['sumayya','Indian','global']}
d # dictionary with list.
```

Out[6]: {'A': 'one', 'B': 'two', 'c': ['sumayya', 'Indian', 'global']}

In [1]:
```
d = {'A':'one','B':'two','C':['sumayya','Indian','globel'], 'D':('bat','cat','ha
d # dictionary with list and tuple.
```

Out[1]:
```
{'A': 'one',
 'B': 'two',
 'C': ['sumayya', 'Indian', 'globel'],
 'D': ('bat', 'cat', 'hat')}
```

In [7]:
```
d = {'A':'one','B':'two','C':{'sumayya','Indian','globel'}, 'D':('bat','cat','ha
d   # dictionay with nested data
```

Out[7]:
```
{'A': 'one',
 'B': 'two',
 'C': {'Indian', 'globel', 'sumayya'},
 'D': ('bat', 'cat', 'hat')}
```

In [9]:
```
d['A']  #access item usimg key
```

Out[9]:
```
'one'
```

In [10]:
```
d[0]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[10], line 1
----> 1 d[0]

KeyError: 0
```

In [11]:
```
d[0:2]  # Dictonary are unorderd,so indexing & slicing are not allowed.
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 d[0:2]

TypeError: unhashable type: 'slice'
```

Add,Remove & Change Items

In [13]:
```
d1={1:'sumayya',2:'taskeen',3:'hyderabad',4:'india'}
d1
```

Out[13]:
```
{1: 'sumayya', 2: 'taskeen', 3: 'hyderabad', 4: 'india'}
```

In [14]:
```
d1[1]='sum'  #changing item
d1
```

Out[14]:
```
{1: 'sum', 2: 'taskeen', 3: 'hyderabad', 4: 'india'}
```

In [17]:
```
d1[5]='asia'  #adding item
d1
```

Out[17]:
```
{1: 'sum', 2: 'taskeen', 3: 'hyderabad', 4: 'india', 5: 'asia'}
```

In [18]:
```
d1.pop(1)  # removing items in dictionary using POP method
d1
```

Out[18]:
```
{2: 'taskeen', 3: 'hyderabad', 4: 'india', 5: 'asia'}
```

Dictionary Methods

Keys and values

In [19]:
```
d = dict({1:'one',2:'two',3:'three',4:'four',5:'five'})
d
```

Out[19]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

In [20]:
```
d.keys()
```

Out[20]:
```
dict_keys([1, 2, 3, 4, 5])
```

In [21]:
```
d.values()
```

Out[21]:
```
dict_values(['one', 'two', 'three', 'four', 'five'])
```

item

In [23]:
```
d.items()  # Access each key-value pair within a dictionary
```

Out[23]:
```
dict_items([(1, 'one'), (2, 'two'), (3, 'three'), (4, 'four'), (5, 'five')])
```

In [26]:
```
#Fromkeys :-
keys={1,2,3,4,5}# create a dictonary from a sequanse of keys
d=dict.fromkeys(keys)
d
```

Out[26]:
```
{1: None, 2: None, 3: None, 4: None, 5: None}
```

In [27]:
```
fruits={1,2,3,4,5}
d1=dict.fromkeys(fruits)
d1
```

Out[27]:
```
{1: None, 2: None, 3: None, 4: None, 5: None}
```

In [30]:
```
fruits={1:'apple',2:'banana',3:'mango',4:'orange',5:'grapes'}
values=fruits
d1=dict.fromkeys(fruits,values)
d1
```

Out[30]:
```
{1: {1: 'apple', 2: 'banana', 3: 'mango', 4: 'orange', 5: 'grapes'},
 2: {1: 'apple', 2: 'banana', 3: 'mango', 4: 'orange', 5: 'grapes'},
 3: {1: 'apple', 2: 'banana', 3: 'mango', 4: 'orange', 5: 'grapes'},
 4: {1: 'apple', 2: 'banana', 3: 'mango', 4: 'orange', 5: 'grapes'},
 5: {1: 'apple', 2: 'banana', 3: 'mango', 4: 'orange', 5: 'grapes'}}
```

In [34]:
```
colors={1,2,3,4,5}
colours_v='red','blue','black'
d2=dict.fromkeys(colors,colours_v)
d2
```

Out[34]:
```
{1: ('red', 'blue', 'black'),
 2: ('red', 'blue', 'black'),
 3: ('red', 'blue', 'black'),
 4: ('red', 'blue', 'black'),
 5: ('red', 'blue', 'black')}
```

In [35]:
```
colors={1,2,3,4,5}
colours_v='red'
d2=dict.fromkeys(colors,colours_v)
d2
```

Out[35]:
```
{1: 'red', 2: 'red', 3: 'red', 4: 'red', 5: 'red'}
```

Get :-

In [36]:
```
d = {1:'one',2:'two',3:'three',4:'four',5:'five'}
d
```

Out[36]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

In [37]:
```
d.get(1) # access value using get
```

Out[37]:
```
'one'
```

In [38]:
```
# update
d = {1:'one',2:'two',3:'three',4:'four',5:'five'}
d
```

Out[38]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

In [39]:
```
d1={1:'onee'}  #update d with update method
d.update(d1)
d
```

Out[39]:
```
{1: 'onee', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

In [40]:
```
#popitem
d.popitem()  # A random item is removed
```

Out[40]:
```
(5, 'five')
```

In [41]:
```
#Clear :-
d.clear()  # Delete all items of dictionary using clear method
```

In [42]:
```
d
```

Out[42]:
```
{}
```

In [56]:
```
dict1={1:'sumayya',2:'taskeen',3:'hyderabad',4:'india'}
dict1
```

Out[56]:
```
{1: 'sumayya', 2: 'taskeen', 3: 'hyderabad', 4: 'india'}
```

In [58]:
```
#copy
dict2=dict1 # Create a new refrance dict2
```

In [60]:
```
id(dict1),id(dict2) # both address are same
```

Out[60]:
```
(2177219429248, 2177219429248)
```

In [63]:
```
dict2=dict1.copy() # create a coy of dictionary
dict2
```

Out[63]:
```
{1: 'sumayya', 2: 'taskeen', 3: 'hyderabad', 4: 'india'}
```

In [65]:
```
s={1:'d',2:'a',3:'f',4:'s',1:'x'}   #Python dictionary, duplicate keys are not a
s
```

Out[65]:
```
{1: 'x', 2: 'a', 3: 'f', 4: 's'}
```

In [66]:
```
s={1:'d',2:'a',3:'f',4:'s',5:'d'}   #Python dictionary, duplicate keys are not a
s
```

Out[66]:
```
{1: 'd', 2: 'a', 3: 'f', 4: 's', 5: 'd'}
```

In [67]:
```
s1=s
```

In [68]:
```
s1
```

Out[68]:
```
{1: 'd', 2: 'a', 3: 'f', 4: 's', 5: 'd'}
```

In [70]:
```
#loop
for i in s1:  #for printing keys
    print(i)
```

```
1
2
3
4
5
```

In [71]:
```
for i in s1:
    print(i,':',s[i])  #for pinting key values
```

```
1 : d
2 : a
3 : f
4 : s
5 : d
```

In [72]:
```
for i in s1:
    print(s[i])  #for printing values
```

```
d
a
f
s
d
```

In [74]:
```
#dictionary membership
s1
```

Out[74]:
```
{1: 'd', 2: 'a', 3: 'f', 4: 's', 5: 'd'}
```

In [75]:
```
'd' in s1
```

Out[75]:
```
False
```

In [76]:
```
1 in s1  #membership is dine for only keys
```

Out[76]:
```
True
```

#All / Any - The all() method returns : * True - If all keys of the dictionary is True. * False - If any key of the dictionary is False. - Any function returns the True if any key of the dictionary is True. If not,any() returns False.

In [77]:
```
all(s1)
```

Out[77]:
```
True
```

In [78]:
```
any(s1)
```

Out[78]:
```
True
```

In [83]:
```
f={1:'h',0:'k'}
```

In [84]:
```
all(f)
```

Out[84]:
```
False
```

In [85]:
```
any(f)
```

Out[85]:
```
True
```

In [86]:
```
f={False:'h',0:'k'}
```

In [87]:
```
any(f)
```

Out[87]:
```
False
```

In [ ]:
```

```