

Dictionary
Methods/Functions

len() :

This function returns the length of Dictionary means, it returns the total number of Key-Value pairs present in the dictionary.

for example:

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print("Length of dictionary A is :", len(A))  
print("Length of dictionary B is :", len(B))
```

OUTPUT

```
Length of dictionary A is : 3  
Length of dictionary B is : 4
```

clear() :

This function simply removes all items from the dictionary. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
A.clear()  
print("Length of dictionary A is :", len(A))  
print("Length of dictionary B is :", len(B))
```

OUTPUT

```
Length of dictionary A is : 0  
Length of dictionary B is : 4
```

NOTE : This function only remove element/items from the dictionary. It does not delete the dictionary

get() :

This function simply return the value of the required Key. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
```

```
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}
```

```
print(A.get(2))
```

```
print(B.get('C'))
```

```
print(A.get(4)) #This key is not available in dictionary A so return NONE
```

```
print(A.get(4),"Key Not Found") #We can specify our message to display if key not found
```

OUTPUT

Two

Cat

None

Key Not Found

NOTE: If key is not available in dictionary then this method will return None or customized message as shown above

items() :

This function returns all the key value pair of dictionary in the form of list of tuples. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print(A.items())  
print(B.items())
```

OUTPUT

```
dict_items([(1, 'One'), (2, 'Two'), (3, 'Three')])  
dict_items([('A', 'Apple'), ('B', 'Bat'), ('C', 'Cat'), ('D', 'Doll')])
```

keys() :

This function return all the keys of dictionary in the form of List. for example :

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print(A.keys())  
print(B.keys())
```

OUTPUT:

```
dict_keys([1, 2, 3])  
dict_keys(['A', 'B', 'C', 'D'])
```

values():

This function return all the values of dictionary in the form of List. for example :

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print(A.values())  
print(B.values())
```

OUTPUT:

```
dict_values(['One', 'Two', 'Three'])  
dict_values(['Apple', 'Bat', 'Cat', 'Doll'])
```

update():

This function merge the key value pair of two dictionary into one. In simple words it simply join/merge the two dictionary. It merge the keys and values of one dictionary into other and overwrites the values of the same key. for example:

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
A.update(B)  
print(A)
```

OUTPUT

```
{1: 'Amit', 2: 'Sunil', 3: 'Three', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
```

#It over writes the values of same keys and add the values of different keys

pop():

This function not only delete the element of required key but also return the deleted value.

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
```

```
a=B.pop(2) #It returns the element of Key - 2
```

```
print(a)
```

```
print(B)
```

OUTPUT

```
Sunil {1: 'Amit', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
```

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
```

```
a=B.pop(6) #It returns the element of Key - 6 print(a)
```

```
print(B)
```

OUTPUT

```
Suman {1: 'Amit', 2: 'Sunil', 5: 'Lata', 7: 'Ravi'}
```

Dictionary

Built-in Dictionary Functions

S.No.	Function & Description
1	<u>len(dict)</u> Gives the total length of the dictionary. It is equal to the number of items in the dictionary. dict = {'Name': 'Aman', 'Age': 37}; print ("Length : %d" % len (dict)) OUTPUT ->2
2	<u>str(dict)</u> Return a printable string representation of a dictionary
3	<u>type(variable)</u> If variable is dictionary, then it would return a dictionary type.

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
1	<code>dict()</code> - creates dictionary <code>x = dict(name = "Aman", age = 37, country = "India")</code> Here x is created as dictionary
2	<code>keys()</code> - returns all the available keys <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>print(x.keys())</code> OUTPUT-> <code>dict_keys(['country', 'age', 'name'])</code>
3	<code>values()</code> - returns all the available values <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>print(x.values())</code> OUTPUT-> <code>dict_values(['India', 37, 'Aman'])</code>

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
4	<code>items()</code> - return the list with all dictionary keys with values. <code>x = dict(name = "Aman", age = 37, country = "India") print(x.items())</code> OUTPUT-> <code>dict_items([('country', 'India'), ('age', 37), ('name', 'Aman')])</code>
5	<code>update()</code> -used to change the values of a key and add new keys <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>d1 = dict(age= 39) x.update(d1,state="Rajasthan") print(x)</code> OUTPUT- <code>{'country': 'India', 'age': 39,'name': 'Aman','state': 'Rajasthan'}</code>

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
6	del -used to remove key <pre>x = dict(name = "Aman", age = 37, country = "India") del x['age'] print(x) OUTPUT->{'country': 'India', 'name': 'Aman'}</pre> del x -> will remove complete dictionary
7	fromkeys() – is used to create dictionary from keys <pre>keys = {'a', 'e', 'i', 'o', 'u' } value ="Vowel" vowels = dict.fromkeys(keys, value) print(vowels) OUTPUT-> {'i': 'Vowel', 'u': 'Vowel', 'e': 'Vowel', 'a': 'Vowel', 'o': 'Vowel'}</pre>

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
8	<code>copy()</code> - returns a shallow copy of the dictionary. <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>y=x.copy()</code> <code>print(y)</code> <code>print(id(x))</code> <code>print(id(y))</code> OUTPUT - >{'country': 'India', 'age': 37, 'name': 'Aman'} 33047872 33047440
9	<code>popitem()</code> – removes last item from dictionary <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>x.popitem()</code> <code>print(x)</code> OUTPUT-> {'age': 37, 'name': 'Aman'}

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
10	<code>setdefault()</code> method returns the value of the item with the specified key. If the key does not exist, insert the key, with the specified value. <pre>x = dict(name = "Aman", country = "India") y=x.setdefault('age',39) print(y) OUTPUT-> 39</pre>
11	<code>max()</code> – returns key having maximum value <pre>Tv = {'a':100, 'b':1292, 'c' : 88} Keymax = max(Tv, key=Tv.get) print(Keymax) OUTPUT-> b</pre>
12	<code>min()</code>- returns key having minimum value

Dictionary

Built-in Dictionary Methods

S.No.	Method & Description
13	sorted- sort by key or value <pre>dict1 = {'b':100, 'a':12, 'c' : 88} y = sorted(dict1.items(),key=lambda x: x[1],reverse=True) print(y)</pre> OUTPUT-> [('b', 100), ('c', 88), ('a', 12)]