

Lecture 3 _ working with Dictionary

Hi pythonistas, hope you are doing well in terms of pythoning. I loved your immediate and overwhelming response for the lecture 1 (strings) and lecture 2 (lists). My name is Akshansh (+91 8384891269, akshanshofficial@gmail.com) and I am here again to discuss one more data type – Dictionary in python.

Let's Get started -

We all have seen dictionary right? What does it contain? Words with their meaning like – beautiful : your girlfriends (I know you are blushing :-))
_ handsome : me

to be realistic, in dictionary you will find words with their meaning like -
job : task
girlfriend : lover etc.

Python is easy right? Dictionary is same in python. You will have something and there will be specific meaning of it. That something is called “key” and it's meaning will be called “value”.

TBR- Dictionary in python is always written in braces {}

Syntax of Dictionary-

Dictionary_name = { key1 : value1, key2 : value 2, key3 : value 3, }

Example of dictionaries -

```
phones = { "samsung" : "Galaxy s10",  
           "apple" : "Iphone 11 pro",  
           "huawei" : "mate 30",  
           "one plus" : "one plus 7 pro" }
```

```
couples = { "alex" : "rihana", "alfonso" : "rihana", "jacob" : "rihana",  
            "john" : "rihana" }
```

Observation – either rihana is a fake Id on facebook or she is the only girl in college in mechanical engineering branch (-_^)

```
family_age= { “Alex” : 25, “mom” : 38, “dad” : 42, “brother” : 21, “sister” : 18 }
```

Printing our very first dictionary-

Suppose we have a dictionary, named group members and country

```
grp_country = { “bob” : “USA” , “darcy” : “england” , “Akshansh” : “India” , “carter” : “france” }
```

```
>>>print(grp_country)
```

```
O/P = { 'Akshansh': 'India', 'bob': 'USA', 'carter': 'India', 'darcy': 'england' }
```

Observations- order of dictionary is not the same as we wrote, it is printed in alphabattical order. Akshansh>Bob>Carter>Darcy

Dictionary Methods to work with dictionary -

1) Accessing values in dictionary -

In dictionary keys and values are connected with colon (:), one key must have a value otherwise it will generate an error.

Suppose we have a dictionary of players and sports

```
players= { “ronaldo” : “football” , “messi” : “football” , “kohli” : “cricket” , “conor mcgregor” : “UFC” , “girl” : “hearts” , “you” : “your life” }
```

printing the whole dictionary-

```
>>>print(players) or you can just write players if you are working in jupyter
```

```
O/P = { 'conor mcgregor': 'UFC', 'girl': 'hearts', 'kohli': 'cricket', 'messi': 'football',
```

```
'ronaldo': 'football'  
'you' : 'your life' }
```

You can access values by using keys:

```
>>>print(players["girl"])  
O/P = 'hearts'
```

Observations – when you have to access a particular value, in python, you must use [] square brackets.

```
>>>print(players["you"])  
O/P = 'your life'
```

2) – Adding new things in a dictionary

See there is a dictionary of friends and town-

```
friends- {"Alex" : "New York", "Robert" : "Manila" , "Jayant" : "Delhi" ,  
"Umar" : "Karachi" , "Kim" : "korea" , "Lily" : "Mexico" }
```

(try to print this dictionary by using print(friends))

now I got one more friend and his name is disturbance and it lives in me. I am adding this to my dictionary -

```
>>>friends["disturbance"] = "me"
```

Explanation – we are in friends (dictionary) and we are adding disturbance (it is a string that is why it is in double quotes). We are adding particular thing so [] are used. And the value of disturbance is me (again this is string that is why it is in double quotes)

now printing dictionary

```
>>>print(friends)
```

```
O/P= {'Alex': 'New York',  
'Jayant': 'Delhi',  
'Kim': 'Korea',  
'Lily': 'Mexico',  
'Robert': 'Manila',  
'Umar': 'Karachi',  
'disturbance': 'me'}
```

TBR – you can add anything in dictionary, it may be integers like 1,3,4,7 or floats like 1.2,4.6

Try yourself – add another friend in dictionary friends named Stark who lives in Paris

3)- Printing an empty dictionary

here is how you can print an empty dict

suppose there is a dictionary-

```
>>>money_I_have = {}  
>>>print(money_I_have)  
O/P = {}
```

there is no output because, I don't money ha ha ha

4) – updating the value or changing the value

suppose your girlfriend name is Alisha and then she dumped you and her new boyfriend is John. List of your girlfriend's boyfriend

```
>>>boyfriend = {"alisha" : "you"}  
#now she has dumped you and her boyfriend is John  
#accessing alisha in dictionary by boyfriend["alisha"] and then we will  
assign a new boyfriend (with = sign) to alisha  
>>>boyfriend["alisha"] = "john"  
O/P = {'alisha' : 'john'} (now she has a new boyfriend)
```

#don't cry boy, let me sympathize you.... **"She deserved better than you ha ha ha"**

5) – removing key values

now you got angry and decided to remove john from alisha's life

what will you do, contact a killer ? No no no, python is not that much cruel you can simply delete alisha, no alisha no john. Liked my masterplan??

```
>>> boyfriend = {"alisha" : "john"}
>>> del boyfriend["alisha"]
#now you have deleted alisha from the dictionary boyfriend
O/P = {} (this is an empty list, now you and john are single again. Ha ha
ha )
```

Observation – if you delete key, values attached to it automatically gets deleted permanently

6) – know your dictionary by get() method

suppose you have a dictionary of the things and stuff you and your family own

```
things= { "bike" : "me" , "car" : "dad", "scooty" : "mom" , "makeup kit" :
"sister" }
```

you want to get car from you home and you will be using get() method

things.get("car") means you are looking for things, dot (.) means you are within things dictionary, get("car") means you want to get car. Now we are printing them all

```
>>> print(things.get("car"))
O/P = 'dad'
#output shows that you have to contact dad
```

let's try to get something that you family doesn't own

```
>>> print(things.get("jet aircraft"))
>>> O/P =
```

See, there is nothing in output. That means no one owns that thing in your family. But it looks weird. Let me change the code

```
>>>print(things.get("jet aircraft", "no one has that"))  
O/P = 'no one has that'
```

```
>>>print(things.get("bike", "no one has that"))  
>>>O/P = 'me'
```

#answer is me, because I've bike. If there is no bike in dictionary, output would have been 'no one has that'

Hope you got it.

7) know your dictionaries keys and values

```
things= { "bike" : "me" , "car" : "dad", "scooty" : "mom" , "makeup kit" :  
"sister" }
```

```
>>>print(things.keys())  
this will print all the keys in dictionaries
```

```
>>>print(things.values())  
this will print all the values in dictionaries
```

8) clear() method

this is will clear all things in dictionary

```
d ={'a' : 2, 'b' : 4, 'c' : 5, 'e' : 7}
```

```
>>>print(d.clear())
```

```
O/P = { }
```

#it has been cleared and there is no things in dictionary

```
>>>print(d)
```

```
O/P = { } (see there is nothing in the dict)
```

9) – know the complete dictionary by items()

suppose a dictionary -

```
dictionary1 = {"samsung" : "s10" , "apple" : "iphone", "huawei" :  
"mate30"}
```

```
>>>print(dictionary1.items())
```

```
O/P = {"samsung" : "s10" , "apple" : "iphone", "huawei" : "mate30"}
```

All the things in dictionary have been printed here.

10) pop() method

suppose there is a dict of friends and their age -

```
firends = { "alex" : 23, "john" : 21, "kaalu" : 25, "goof" : 28}
```

now kaalu has broken the friendship due to some reasons (sssh his girlfriend told him to stay away from you guys :^)

now you have to remove kaalu from your friends, pop method will help you in this.

```
>>>print(firends.pop("kaalu"))
```

```
O/P = 'alex' : 23, 'john' : 21, 'goof' : 28
```

see pop method has removed kaalu from your friend circle

11)- update() method

since kaalu has gone, your circle has become

```
friends = { "alex" : 23, "john" : 21, "goof" : 28}
```

you met jacob and his friends

```
jacob_frnds = { "jacob" : 24, "richa" : 21, "jasmine" : 23}
```

you and jacob has become friends, and by hook or crook you came to know richa and jasmine too. (yohoooooooo! Mission successful) Now you have to add them in your circle. Here you want to update your circle. Let python do it for you.

```
>>> friends.update(jacob_frnds)
#this is insert jacob's friends in your friends dictionary
```

```
>>> print(friends)
O/P = { 'alex': 23, 'john': 21, 'goof': 28, 'jacob': 24, 'richa': 24,
'jasmine': 23}
```

#see your friend circle has become larger and friendly of course because boys tend to be more friendly in front of girls \m/(-_-)\m/

-here you can update the values of the keys too -

```
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d1.update(b=200, d=400)
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}
```

another example -

```
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d1.update([('b', 200), ('d', 400)])
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}
```

This is all from my end today for the Dictionaries in python. Hope you guys find it useful. Till the next lecture notes, bye bye Namaste!

Keep pythoning, pythonista !