Dictionary – Unit 5

chapter 11 (as per text book – think python)

Bhavika Vaghela

Asst.Prof.

PICS

- A **dictionary** is like a list, but more general. In a list, the indices have to be integers; in a dictionary they can be (almost) any type.
- In python data of tuple, list or set are accessing by using index value but in dictionary it access by using its key value.
- Key must be unique no duplicate key is allowed in dictionary.
- It declare using {} bracket.
- Its have paired data it means key and its value.

Syntax :

<name of dictionary> = {key : value1, key2:value2, key3:value3…}

You can also declare it like

Name of dictionary = { key : value,
key : value,
key : value…..}

You can also declare empty dictionary like : dict_name = {}
and later on you can append data into it. By passing its key value.

- The function **dict**() creates a new dictionary with no items. Because dict is the name of a built-in function, you should avoid using it as a variable name.

```
>>> numdict=dict() # creating empty dictionary using dict() function
>>> numdict   # printing dictionary numdict.
{}     # output is empty dictionary
```

You can add element like below

```
>>> numdict['one']=['1']
>>> numdict['two']=['2']
>>> numdict
{'one': ['1'], 'two': ['2']} #but here you can see [] bracket it take as value
>>> numdict['three']=3 # here you can also assign value direct.
>>> numdict
{'one': ['1'],
'two': ['2'], 'three': 3} # check the output on you machine also.
```

Here one, two, three is known as key and 1,2,3 is its value respectively

- **Note –** Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.
- The dictionary is the data type in python which can simulate the real-life data arrangement where some specific value exists for some particular key.
- In other words, we can say that a dictionary is the collection of key-value pairs where the value can be any python object whereas the keys are the immutable python object, i.e., Numbers, string or tuple.
- Python dictionary is an unordered collection of items. While other compound data types have only value as an element, a dictionary has a key: value pair.
- **So key is mandatory to access any data from dictionary. Without it you cannot access the data of dictionary.**
- **Key and value separated by ":".**
- **Key may be any integer or string or mix.**

One more example (creating dictionary for English to Spanish)

```
>>> engtosp={'one':'uno','two':'dos','three':'tres'}
>>> engtosp
{'one': 'uno', 'two': 'dos', 'three': 'tres'}
>>> engtosp['four']='cuatro'
>>> engtosp['five']='cinco'
>>> engtosp['six']='seis'
>>> engtosp
{'one': 'uno', 'two': 'dos', 'three': 'tres', 'four': 'cuatro', 'five': 'cinco', 'six': 'seis'}
>>> engtosp['Six']='seis‘ #here it not gives an error as key is case sensitive
>>> engtosp
{'one': 'uno', 'two': 'dos', 'three': 'tres', 'four': 'cuatro', 'five': 'cinco', 'six': 'seis',
'Six': 'seis'}
```

Here adding element into dictionary

So first key is six and second one is Six that’s why it not gives error .

All of you please try it on your machine with other examples also.

Accessing element from dictionary

- you can access element of dictionary using two way .

1) By passing key value inside [] bracket like

```
>>> engtosp['three']
'tres'
>>> print(engtosp['three'])
Tres
```

2) By passing key value using get() method like

```
>>> engtosp.get('two')
'dos'
>>> print(engtosp.get('three'))
tres
```

Change element in dictionary

- You can change element of dictionary by passing its key value.
- Dictionary are mutable. We can add new items or change the value of existing items using assignment operator.
- **If the key is already present, value gets updated, else a new key: value pair is added to the dictionary.**

```
>>> engtosp['Six']='its not duplicate key'
>>> engtosp['seven']='seven‘
```

****so no error if key is not exists**.

Delete or remove element form the dictionary

- We can remove a particular item in a dictionary by using the method **pop()**. This method removes as item with the provided key and returns the value.
- The method, **popitem()** can be used to remove and return an arbitrary item (key, value) form the dictionary. All the items can be removed at once using the **clear()** method.
- We can also use the **del** keyword to remove individual items or the entire dictionary itself.

**** please refer next slide for the remove operations and try it on your machine also**

```
# create a dictionary
squares = {1:1, 2:4, 3:9, 4:16, 5:25}
# remove a particular item
print(squares.pop(4)) #output will be 16
# remove an arbitrary item
print(squares.popitem())
print(squares)# Output: {2: 4, 3: 9, 5: 25}
# delete a particular item
del squares[5]
print(squares) # Output: {2: 4, 3: 9}
# remove all items
squares.clear()
print(squares) # Output: {}
# delete the dictionary itself
del squares
```

Iterating Dictionary using for loop

 # for loop to print all the keys of a dictionary

```
Employee = {"Name": "John", "Age": 29, "salary":25000,"Company":"GOOGLE"}
for x in Employee:
    print(x);
```

Output :

```
Name
Company
salary
Age
```

#for loop to print all the values of the dictionary

```
Employee = {"Name": "John", "Age": 29, "salary":25000,"Company":"GOOGLE"}
for x in Employee:
    print(Employee[x]);
```

Output :

29
GOOGLE
John
25000

for loop to print the values of the dictionary by using values() method.

```
Employee = {"Name": "John", "Age": 29, "salary":25000,"Company":"GOOGLE"}
for x in Employee.values():
    print(x);
```

Output:

```
GOOGLE
25000
John
29
```

for loop to print the items of the dictionary by using items() method.

```
Employee = {"Name": "John", "Age": 29, "salary":25000,"Company":"GOOGLE"}
for x in Employee.items():
    print(x);
```

Output:

```
('Name', 'John')
('Age', 29)
 ('salary', 25000)
('Company', 'GOOGLE')
```

You can also access like this using for loop

```
Employee = {"Name": "John", "Age": 29, "Salary":25000,"Company":"GOOGLE","Name
":"Johnn"}
for x,y in Employee.items():
    print(x,y)
```

Output:

```
Salary 25000
Company GOOGLE
Name Johnn
Age 29
```

**** please try it on your machine also**

Built-in Dictionary methods

SN	Method	Description
1	dic.clear()	It is used to delete all the items of the dictionary.
2	dict.copy()	It returns a shallow copy of the dictionary.
3	dict.fromkeys(iterable, value = None, /)	Create a new dictionary from the iterable with the values equal to value.
4	dict.get(key, default = "None")	It is used to get the value specified for the passed key.
5	dict.has_key(key)	It returns true if the dictionary contains the specified key.
6	dict.items()	It returns all the key-value pairs as a tuple.
7	dict.keys()	It returns all the keys of the dictionary.
8	dict.setdefault(key,default= "None")	It is used to set the key to the default value if the key is not specified in the dictionary
9	dict.update(dict2)	It updates the dictionary by adding the key-value pair of dict2 to this dictionary.
10	dict.values()	It returns all the values of the dictionary.
11	len()	

Built-in Dictionary functions

SN	Function	Description
1	cmp(dict1, dict2)	It compares the items of both the dictionary and returns true if the first dictionary values are greater than the second dictionary, otherwise it returns false.
2	len(dict)	It is used to calculate the length of the dictionary.
3	str(dict)	It converts the dictionary into the printable string representation.
4	type(variable)	It is used to print the type of the passed variable.

`len(d)` returns the number of items in d.

`d.keys()` returns a list containing the keys in d.

`d.values()` returns a list containing the values in d.

`k in d` returns True if key k is in d.

`d[k]` returns the item in d with key k.

`d.get(k, v)` returns `d[k]` if k is in d, and v otherwise.

`d[k] = v` associates the value v with the key `k in d`. If there is already a value associated with k, that value is replaced.

`del d[k]` removes the key k from d.

`for k in d` iterates over the keys in d.

Develop below listed program

- Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x). Sample Dictionary . Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25} where value of n is enter by user.
- Create a dictionary which shows the occurrence/ frequency of each character present in string. [hint : if user enter “wel come to parul university” than your dictionary contain {a:1,b:0,c:1,d:0,e:3, f:0……} like this.
- As python dictionary take any object as its value so create a dictionary which shows the use of it. [hint : take employee id as key and details of employee as data] e.g. {'1001': ['bhavika', 30, 'kalali'], '1002': ['amit', 25, 'manjalpur']} and also try to access only name or age from the data.
- WAP program for encode string which is entered by user like if user enter **WEL COME** than it must be print like **JRY PBZR**, for that create your dictionary like below {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u', 'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b', 'p':'c', 'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k', 'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S', 'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A', 'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I', 'W':'J', 'X':'K', 'Y':'L', 'Z':'M'}

- Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary :
  dic1={1:10, 2:20}
  dic2={3:30, 4:40}
  dic3={5:50,6:60}
  Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
- Change the above program like add the keys and its value in third dictionary like output is {4:40,6:60…} you need to add both key and its value.
- Write a Python program to combine two dictionary adding values for commo keys. d1={'a':100,'b':200,'c':300}, d2={'a':300,'b':200,'d':400} **Sample output**: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
- Write a menu driven program which shows the use of each method for remove and deletion data from dictionary.
- Write a Python program to create a dictionary of keys x, y, and z where each key has as value a list from 11-20, 21-30, and 31-40 respectively. Access the fifth value of each key from the dictionary. {'x': [11, 12, 13, 14, 15, 16, 17, 18, 19],
  'y': [21, 22, 23, 24, 25, 26, 27, 28, 29], 'z': [31, 32, 33, 34, 35, 36, 37, 38, 39]} then output is
  15
  25
  35
- https://www.w3resource.com/python-exercises/dictionary/