# **Nazneen Mohammed Ameen**

# **Feb 20, 2020**

# **Programming with Python**

# **Assignment04**

**Lists and Dictionaries**

**INTRODUCTION**

In the module, you learn about list and dictionaries properties. Lists is ordered and changeable. Allows duplicate members whereas dictionaries are unordered sets. But the main difference is that items in dictionaries are accessed via keys and not via their position. A dictionary is an associative array (also known as hashes). Any key of the dictionary is associated (or mapped) to a value. Also, you learn working with files and list, files and dictionaries.

**Lists**

The list a collection which is ordered and changeable. is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type. lists include a lot of built functions.

**Dictionary**

Dictionaries are Python’s implementation of a data structure that is more generally known as an associative array. A dictionary consists of a collection of key-value pairs. Each key-value pair maps the key to its associated value. Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

You can define a dictionary by enclosing a comma-separated list of key-value pairs in curly braces ({}). A colon (:) separates each key from its associated value as shown below.

d = {

<key>: <value>,

<key>: <value>,

.

<key>: <value>

}

**Accessing Values in Dictionary**

To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value. Following is a simple example.

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

print ("dict['Name']: ", dict['Name'])

print ("dict['Age']: ", dict['Age'])

When the above code is executed, it produces the following result.

dict ['Name']: Zara

dict ['Age']: 7

If we attempt to access a data item with a key, which is not a part of the dictionary, we get an error as follows −

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

print ("dict['Alice']: ", dict['Alice'])

When the above code is executed, it produces the following result −

dict['Zara']:

Traceback (most recent call last):

File "test.py", line 4, in <module>

print "dict['Alice']: ", dict['Alice'];

KeyError: 'Alice'

**Updating Dictionary**

You can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown in a simple example given below.

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

dict['Age'] = 8; # update existing entry

dict['School'] = "DPS School" # Add new entry

print ("dict['Age']: ", dict['Age'])

print ("dict['School']: ", dict['School'])

When the above code is executed, it produces the following result.

dict['Age']: 8

dict['School']: DPS School

**Delete Dictionary Elements**

You can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

To explicitly remove an entire dictionary, just use the **del** statement. Following is a simple example. <https://www.tutorialspoint.com/python3/python_dictionary.htm>

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

del dict['Name'] # remove entry with key 'Name'

dict.clear() # remove all entries in dict

del dict # delete entire dictionary

print ("dict['Age']: ", dict['Age'])

print ("dict['School']: ", dict['School'])

This produces the following result.

An exception is raised because after **del dict**, the dictionary does not exist anymore.

dict['Age']:

Traceback (most recent call last):

File "test.py", line 8, in <module>

print "dict['Age']: ", dict['Age'];

TypeError: 'type' object is unsubscriptable

**Built-in Dictionary Functions and Methods**

**Python includes the following dictionary functions as shown in table 5-1**:

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Description** |
| 1 | [cmp(dict1, dict2)](https://www.tutorialspoint.com/python3/dictionary_cmp.htm) No longer available in Python 3. |
| 2 | [len(dict)](https://www.tutorialspoint.com/python3/dictionary_len.htm)  Gives the total length of the dictionary. This would be equal to the number of items in the dictionary. |
| 3 | [str(dict)](https://www.tutorialspoint.com/python3/dictionary_str.htm) Produces a printable string representation of a dictionary |
| 4 | [type(variable)](https://www.tutorialspoint.com/python3/dictionary_type.htm)  Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type. |

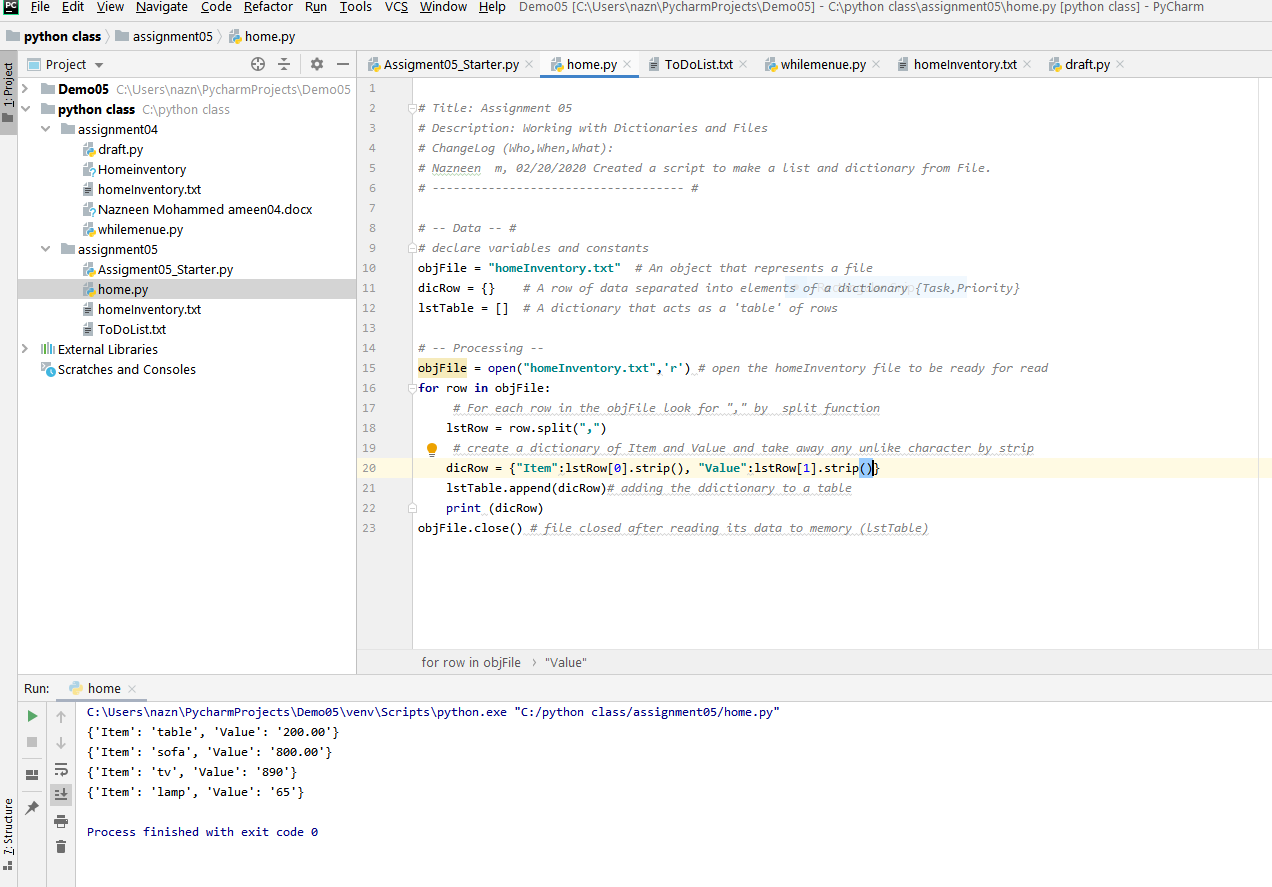
**Table 5-1 shows dictionary function**

**Python includes the following dictionary methods as shown in the table 5-2:**

|  |  |
| --- | --- |
| **Sr.No.** | **Method & Description** |
| 1 | [dict.clear()](https://www.tutorialspoint.com/python3/dictionary_clear.htm) Removes all elements of dictionary *dict* |
| 2 | [dict.copy()](https://www.tutorialspoint.com/python3/dictionary_copy.htm) Returns a shallow copy of dictionary *dict* |
| 3 | [dict.fromkeys()](https://www.tutorialspoint.com/python3/dictionary_fromkeys.htm) Create a new dictionary with keys from seq and values *set* to *value*. |
| 4 | [dict.get(key, default=None)](https://www.tutorialspoint.com/python3/dictionary_get.htm) For *key* key, returns value or default if key not in dictionary |
| 5 | [dict.has\_key(key)](https://www.tutorialspoint.com/python3/dictionary_has_key.htm) Removed, use the *in* operation instead. |
| 6 | [dict.items()](https://www.tutorialspoint.com/python3/dictionary_items.htm) Returns a list of *dict*'s (key, value) tuple pairs |
| 7 | [dict.keys()](https://www.tutorialspoint.com/python3/dictionary_keys.htm) Returns list of dictionary dict's keys |
| 8 | [dict.setdefault(key, default = None)](https://www.tutorialspoint.com/python3/dictionary_setdefault.htm) Similar to get(), but will set dict[key] = default if *key* is not already in dict |
| 9 | [dict.update(dict2)](https://www.tutorialspoint.com/python3/dictionary_update.htm) Adds dictionary *dict2*'s key-values pairs to *dict* |
| 10 | [dict.values()](https://www.tutorialspoint.com/python3/dictionary_values.htm) Returns list of dictionary *dict*'s values |

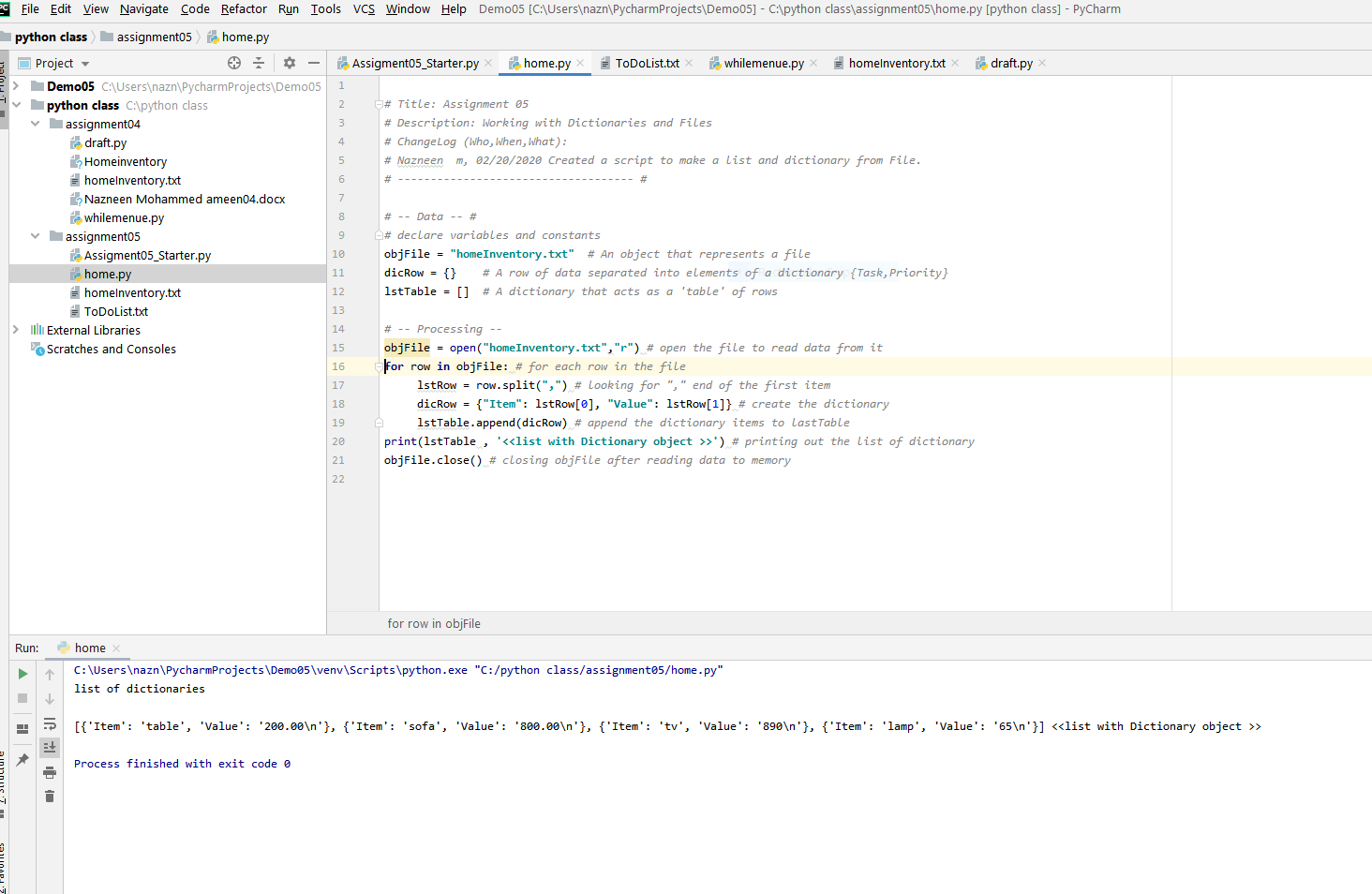
**Table 5-2 shows dictionaries method**

**Working with Table of Dictionaries**

In this example will read data from text file and put it to make a table of dictionary objects.as shown in figure 5-1

**Figure 5-1 shows read from file to build a dictionary**

**figure 5-2 below shows how to create list of dictionaries from a text file .**

**figure 5-2 list of dictionaries**

**Summary**

In this module you learned about dictionaries and its properties, function of dictionaries and how to read from text file to build dictionary.