

PY0101EN-2-4-Dictionaries

May 10, 2022

1 Dictionaries in Python

Estimated time needed: **25** minutes

1.1 Objectives

After completing this lab you will be able to:

- Work with and perform operations on dictionaries in Python

Table of Contents

```
<ul>
  <li>
    <a href="https://#dic">Dictionaries</a>
    <ul>
      <li><a href="https://content/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_medium=Exinfluencer&utm_source=Exinfluencer">Content</a></li>
      <li><a href="https://key/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_medium=Exinfluencer&utm_source=Exinfluencer">Key</a></li>
    </ul>
  </li>
  <li>
    <a href="https://#quiz">Quiz on Dictionaries</a>
  </li>
</ul>
```

Dictionaries

What are Dictionaries?

A dictionary consists of keys and values. It is helpful to compare a dictionary to a list. Instead of being indexed numerically like a list, dictionaries have keys. These keys are the keys that are used to access values within a dictionary.

An example of a Dictionary Dict:

```
[7]: # Create the dictionary

Dict = {"key1": 1, "key2": "2", "key3": [3, 3, 3], "key4": (4, 4, 4), ('key5'): 5, (0, 1): 6}
Dict
```

```
[7]: {'key1': 1,
      'key2': '2',
      'key3': [3, 3, 3],
      'key4': (4, 4, 4),
      'key5': 5,
      (0, 1): 6}
```

The keys can be strings:

```
[8]: # Access to the value by the key

Dict["key1"]
```

```
[8]: 1
```

Keys can also be any immutable object such as a tuple:

```
[9]: # Access to the value by the key

Dict[(0, 1)]
```

```
[9]: 6
```

Each key is separated from its value by a colon “:”. Commas separate the items, and the whole dictionary is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this “{}”.

```
[10]: # Create a sample dictionary

release_year_dict = {"Thriller": "1982", "Back in Black": "1980", \
                     "The Dark Side of the Moon": "1973", "The Bodyguard": \
                     ↪ "1992", \
                     "Bat Out of Hell": "1977", "Their Greatest Hits \
                     ↪ (1971-1975)": "1976", \
                     "Saturday Night Fever": "1977", "Rumours": "1977"}
release_year_dict
```

```
[10]: {'Thriller': '1982',
      'Back in Black': '1980',
      'The Dark Side of the Moon': '1973',
      'The Bodyguard': '1992',
      'Bat Out of Hell': '1977',
      'Their Greatest Hits (1971-1975)': '1976',
      'Saturday Night Fever': '1977',
      'Rumours': '1977'}
```

In summary, like a list, a dictionary holds a sequence of elements. Each element is represented by a key and its corresponding value. Dictionaries are created with two curly braces containing

keys and values separated by a colon. For every key, there can only be one single value, however, multiple keys can hold the same value. Keys can only be strings, numbers, or tuples, but values can be any data type.

It is helpful to visualize the dictionary as a table, as in the following image. The first column represents the keys, the second column represents the values.

Keys

You can retrieve the values based on the names:

```
[11]: # Get value by keys
      release_year_dict['Thriller']
```

```
[11]: '1982'
```

This corresponds to:

Similarly for The Bodyguard

```
[12]: # Get value by key
      release_year_dict['The Bodyguard']
```

```
[12]: '1992'
```

Now let us retrieve the keys of the dictionary using the method keys():

```
[13]: # Get all the keys in dictionary
      release_year_dict.keys()
```

```
[13]: dict_keys(['Thriller', 'Back in Black', 'The Dark Side of the Moon', 'The
Bodyguard', 'Bat Out of Hell', 'Their Greatest Hits (1971-1975)', 'Saturday
Night Fever', 'Rumours'])
```

You can retrieve the values using the method values():

```
[14]: # Get all the values in dictionary
      release_year_dict.values()
```

```
[14]: dict_values(['1982', '1980', '1973', '1992', '1977', '1976', '1977', '1977'])
```

We can add an entry:

```
[15]: # Append value with key into dictionary
      release_year_dict['Graduation'] = '2007'
      release_year_dict
```

```
[15]: {'Thriller': '1982',  
      'Back in Black': '1980',  
      'The Dark Side of the Moon': '1973',  
      'The Bodyguard': '1992',  
      'Bat Out of Hell': '1977',  
      'Their Greatest Hits (1971-1975)': '1976',  
      'Saturday Night Fever': '1977',  
      'Rumours': '1977',  
      'Graduation': '2007'}
```

We can delete an entry:

```
[16]: # Delete entries by key  
  
del(release_year_dict['Thriller'])  
del(release_year_dict['Graduation'])  
release_year_dict
```

```
[16]: {'Back in Black': '1980',  
      'The Dark Side of the Moon': '1973',  
      'The Bodyguard': '1992',  
      'Bat Out of Hell': '1977',  
      'Their Greatest Hits (1971-1975)': '1976',  
      'Saturday Night Fever': '1977',  
      'Rumours': '1977'}
```

We can verify if an element is in the dictionary:

```
[17]: # Verify the key is in the dictionary  
  
'The Bodyguard' in release_year_dict
```

```
[17]: True
```

Quiz on Dictionaries

You will need this dictionary for the next two questions:

```
[18]: # Question sample dictionary  
  
soundtrack_dic = {"The Bodyguard": "1992", "Saturday Night Fever": "1977"}  
soundtrack_dic
```

```
[18]: {'The Bodyguard': '1992', 'Saturday Night Fever': '1977'}
```

a) In the dictionary soundtrack_dic what are the keys ?

```
[25]: # Write your code below and press Shift+Enter to execute  
soundtrack_dic.keys()
```

```
[25]: dict_keys(['The Bodyguard', 'Saturday Night Fever'])
```

[Click here for the solution](#)

```
soundtrack_dic.keys() # The Keys "The Bodyguard" and "Saturday Night Fever"
```

b) In the dictionary soundtrack_dic what are the values ?

```
[26]: # Write your code below and press Shift+Enter to execute
soundtrack_dic.values()
```

```
[26]: dict_values(['1992', '1977'])
```

[Click here for the solution](#)

```
soundtrack_dic.values() # The values are "1992" and "1977"
```

You will need this dictionary for the following questions:

The Albums Back in Black, The Bodyguard and Thriller have the following music recording sales in millions 50, 50 and 65 respectively:

a) Create a dictionary album_sales_dict where the keys are the album name and the sales in millions are the values.

```
[30]: # Write your code below and press Shift+Enter to execute
album_sales_dict = {'Back in Black':50, 'The Bodyguard':50, 'Thriller':65}
```

[Click here for the solution](#)

```
album_sales_dict = {"The Bodyguard":50, "Back in Black":50, "Thriller":65}
```

b) Use the dictionary to find the total sales of Thriller:

```
[61]: # Write your code below and press Shift+Enter to execute
x = list(album_sales_dict.values())
s = 0
for i in range(0,3):
    s = x[i] + s

print(s)

# Com list comprehension ele faz pra cada elemento da tupla, não soma num total
y = 0
r = [y + x[i] for i in range(0,3)]
print(r)
```

165

[50, 50, 65]

[Click here for the solution](#)

```
album_sales_dict["Thriller"]
```

c) Find the names of the albums from the dictionary using the method `keys()`:

```
[62]: # Write your code below and press Shift+Enter to execute
album_sales_dict.keys()
```

```
[62]: dict_keys(['Back in Black', 'The Bodyguard', 'Thriller'])
```

[Click here for the solution](#)

```
album_sales_dict.keys()
```

d) Find the values of the recording sales from the dictionary using the method `values()`:

```
[65]: # Write your code below and press Shift+Enter to execute
album_sales_dict.values()
```

```
[65]: dict_values([50, 50, 65])
```

[Click here for the solution](#)

```
album_sales_dict.values()
```

The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python.

1.2 Author

Joseph Santarcangelo

1.3 Other contributors

Mavis Zhou

1.4 Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-01-10	2.2	Malika Singla	Removed the readme for GitShare
2020-09-09	2.1	Malika Singla	Updated the variable <code>soundtrack_dict</code> to <code>soundtrack_dic</code> in Questions
2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab

##

© IBM Corporation 2020. All rights reserved.