IBM Developer SKILLS NETWORK

# Dictionaries in Python

Estimated time needed: **20** minutes

## Objectives

After completing this lab you will be able to:

- Work with libraries in Python, including operations

## Table of Contents

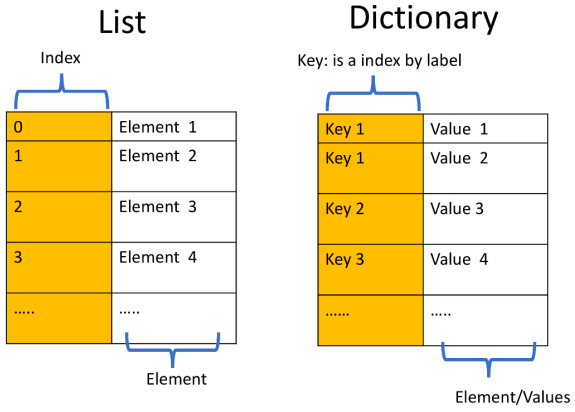


---

## Dictionaries

### What are Dictionaries?

A dictionary consists of keys and values. It is helpful to compare a dictionary to a list. Instead of the numerical indexes such as 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`:

```
[ ]:
# Create the dictionary

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

The keys can be strings:

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

Dict["key1"]
```

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

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

Dict[(0, 1)]
```

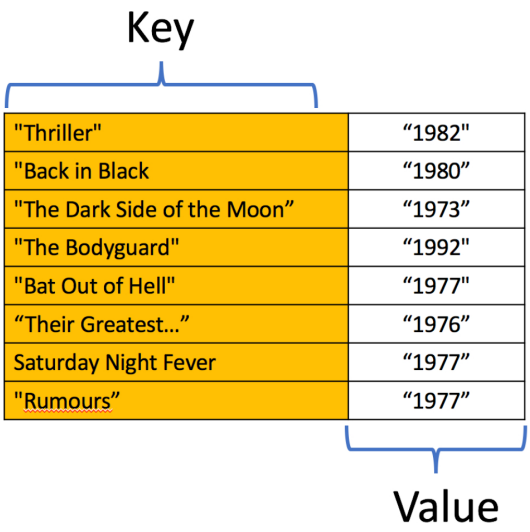
Each key is separated from its value by a colon "`:`". Commas separate the items, and the whole dictionary is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this "`{}`".

```
[ ]:
# 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
```

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:

[ ]:
```
# Get value by keys

release_year_dict['Thriller']
```

This corresponds to:

| | |
|---|---|
| "Thriller" | “1982" |
| "Back in Black | “1980” |
| "The Dark Side of the Moon” | “1973” |
| "The Bodyguard" | “1992" |
| "Bat Out of Hell" | “1977" |
| “Their Greatest...” | “1976” |
| “Saturday Night Fever” | “1977” |
| "Rumours” | “1977” |

Similarly for The Bodyguard

[ ]:
```
# Get value by key

release_year_dict['The Bodyguard']
```

| | |
|---|---|
| "Thriller" | “1982" |
| "Back in Black | “1980” |
| "The Dark Side of the Moon” | “1973” |
| "The Bodyguard" | “1992" |
| "Bat Out of Hell" | “1977" |
| “Their Greatest...” | “1976” |
| “Saturday Night Fever” | “1977” |
| "Rumours” | “1977” |

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

[ ]:
```
# Get all the keys in dictionary

release_year_dict.keys()
```

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

[ ]:
```
# Get all the values in dictionary

release_year_dict.values()
```

We can add an entry:

[ ]:
```
# Append value with key into dictionary

release_year_dict['Graduation'] = '2007'
release_year_dict
```

We can delete an entry:

[ ]:
```
# Delete entries by key

del(release_year_dict['Thriller'])
del(release_year_dict['Graduation'])
release_year_dict
```

We can verify if an element is in the dictionary:

```
[ ]: # Verify the key is in the dictionary

'The Bodyguard' in release_year_dict
```

---

**Did you know?** IBM Watson Studio lets you build and deploy an AI solution, using the best of open source and IBM software and giving your team a single environment to work in. Learn more here.

## Quiz on Dictionaries

You will need this dictionary for the next two questions:

```
[ ]: # Question sample dictionary

soundtrack_dic = {"The Bodyguard":"1992", "Saturday Night Fever":"1977"}
soundtrack_dic
```

a) In the dictionary `soundtrack_dic` what are the keys ?

```
[6]: def soundtrack_dic:
         soundtrack_dic = {"The Bodyguard":"1992", "Saturday Night Fever":"1977"}
```

```
  File "<ipython-input-6-68afdcea4b9a>", line 1
    def soundtrack_dic:
                      ^
SyntaxError: invalid syntax
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
soundtrack_dic.keys() # The Keys "The Bodyguard" and "Saturday Night Fever"
-->
```

b) In the dictionary `soundtrack_dic` what are the values ?

```
[ ]:
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
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.

```
[ ]:
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_sales_dict = {"The Bodyguard":50, "Back in Black":50, "Thriller":65}
-->
```

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

```
[ ]:
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_sales_dict["Thriller"]
-->
```

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

```
[ ]:
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_sales_dict.keys()
-->
```

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

```
[ ]:
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_sales_dict.values()
-->
```

---

## The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python. However, there is one more thing you need to do. The Data Science community encourages sharing work. The best way to share and showcase your work is to share it on GitHub. By sharing your notebook on GitHub you are not only building your reputation with fellow data scientists, but you can also show it off when applying for a job. Even though this was your first piece of work, it is never too early to start building good habits. So, please read and follow this article to learn how to share your work.

---

## Get IBM Watson Studio free of charge!


## Author

Joseph Santarcangelo

## Other contributors

Mavis Zhou

## Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|---|---|---|---|
| 2020-09-09 | 2.1 | Malika Singla | Updated the variable soundtrack_dict to soundtrack_dic in Questions |
| 2020-08-26 | 2.0 | Lavanya | Moved lab to course repo in GitLab |