(https://cognitiveclass.ai/)

# Dictionaries in Python

**Welcome!** This notebook will teach you about the dictionaries in the Python Programming Language. By the end of this lab, you'll know the basics dictionary operations in Python, including what it is, and the operations on it.



(http://cocl.us/topNotebooksPython101Coursera)

## Table of Contents

Estimated time needed: **20 min**

# 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` :

In [2]:

```
# Create the dictionary

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

Out[2]:

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

The keys can be strings:

In [3]:

```
# Access to the value by the key

Dict["key1"]
```

Out[3]:

1

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

In [4]:

```
# Access to the value by the key

Dict[(0, 1)]
```

Out[4]:

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 " **{}** ".

In [5]:

```
# 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
```
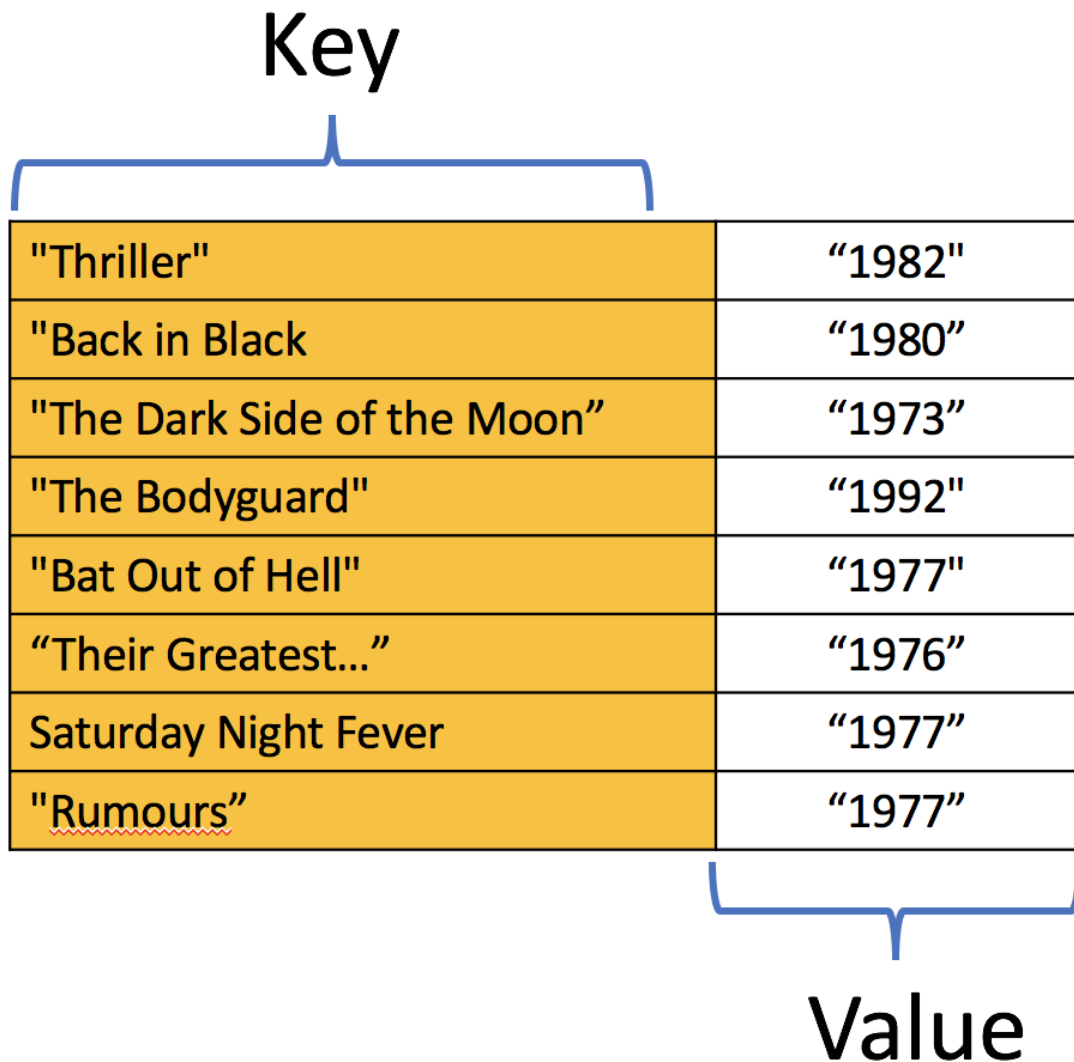
Out[5]:

```
{'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.

## Key

| | |
|---|---|
| "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" |

## Value

## Keys

You can retrieve the values based on the names:

In [6]:

```
# Get value by keys

release_year_dict['Thriller']
```

Out[6]:

```
'1982'
```

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**

In [7]:

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

Out[7]:

```
'1992'
```

| | |
|---|---|
| "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 you retrieve the keys of the dictionary using the method `release_year_dict()`:

In [8]:

```
# Get all the keys in dictionary

release_year_dict.keys()
```

Out[8]:

```
dict_keys(['Thriller', 'Back in Black', 'The Dark Side of the Moon',
'The Bodyguard', 'Bat Out of Hell', 'Their Greatest Hits (1971-197
5)', 'Saturday Night Fever', 'Rumours'])
```

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

In [9]:

```
# Get all the values in dictionary

release_year_dict.values()
```

Out[9]:

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

We can add an entry:

In [10]:

```
# Append value with key into dictionary

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

Out[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',
 'Graduation': '2007'}
```

We can delete an entry:

In [11]:

```python
# Delete entries by key

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

Out[11]:

```
{'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:

In [12]:

```python
# Verify the key is in the dictionary

'The Bodyguard' in release_year_dict
```

Out[12]:

```
True
```

# Quiz on Dictionaries

**You will need this dictionary for the next two questions:**

In [13]:

```python
# Question sample dictionary

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

Out[13]:

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

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

In [14]:

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

Out[14]:

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

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

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

In [16]:

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

Out[16]:

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

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

---

**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.

In [23]:

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

Out[23]:

```
{'Back in Black': '50', 'The Bodyguard': '50', 'Thriller': '65'}
```

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

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

`In [25]:`

```
# Write your code below and press Shift+Enter to execute
album_sales_dict['Thriller']
```

`Out[25]:`

```
'65'
```

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

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

`In [26]:`

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

`Out[26]:`

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

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

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

`In [27]:`

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

`Out[27]:`

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

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

# 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 (https://cognitiveclass.ai/blog/data-scientists-stand-out-by-sharing-your-notebooks/) to learn how to share your work.

# Get IBM Watson Studio free of charge!

## Get IBM Watson Studio free of charge!

Build and train AI & machine learning models, prepare and analyze data – all in a flexible, hybrid cloud environment. Get IBM Watson Studio Lite Plan free of charge.

| Learn | Create | Collaborate |
|---|---|---|
| Get started or get better with built-in learning. | Use the best of open source tooling with IBM innovation. | Work smarter using community, work faster with your team. |

**Sign Up For a Free Trial**

(https://cocl.us/bottemNotebooksPython101Coursera)

## About the Authors:

Joseph Santarcangelo (https://www.linkedin.com/in/joseph-s-50398b136/) is a Data Scientist at IBM, and holds a PhD in Electrical Engineering. His research focused on using Machine Learning, Signal Processing, and Computer Vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: Mavis Zhou (www.linkedin.com/in/jiahui-mavis-zhou-a4537814a)