PY0101EN-2-3-Dictionaries

August 24, 2020

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.

```
<a href="http://cocl.us/topNotebooksPython101Coursera">
     <img src="https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass
</a>
```

Table of Contents

```
<ul>
    <li>
        <a href="#dic">Dictionaries</a>
        <ul>
            <li><a href="content">What are Dictionaries?</a></li>
            <li><a href="key">Keys</a></li>
        </ul>
    </li>
    <li>
        <a href="#quiz">Quiz on Dictionaries</a>
    </li>
</ul>
<p>
    Estimated time needed: <strong>20 min</strong>
</p>
```

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 "{}".

[3]:
```
# 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={"Thriller":"1082", "Back in black":"1980",\
                   "The Dark Side of the Moon": "1973", "The Body Guard":␣
 ↪"1992",\
                    "Bat Out of Hell": "1877", "Their Greatest Hits (1971-1975)":
 ↪ "1976",\
                    "Saturday Night Fever": "1977", "Rumors": "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:

[ ]:
```
# Get value by keys
```

```
release_year_dict['Thriller']
```

This corresponds to:

Similarly for The Bodyguard

[ ]:
```
# Get value by key

release_year_dict['The Bodyguard']
```

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

[ ]:
```
# 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
```

Quiz on Dictionaries

You will need this dictionary for the next two questions:

[6]:
```
# Question sample dictionary

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

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

a) In the dictionary soundtrack_dict what are the keys ?

[10]:
```
# Write your code below and press Shift+Enter to execute
soundtrack_dic.keys()
```

[10]:
```
dict_keys(['The Bodyguard', 'Satyrday Night Fever'])
```

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

b) In the dictionary soundtrack_dict what are the values ?

[18]:
```
# Write your code below and press Shift+Enter to execute
soundtrack_dic.values()
```

[18]:
```
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.

[15]:
```
# Write your code below and press Shift+Enter to execute
album_sales_dict={"Back in Black": 50, "The Bodyguard": 30, "Thriller":65}
```

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

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

[16]:
```
# Write your code below and press Shift+Enter to execute
album_sales_dict["Thriller"]
```

[16]:
```
65
```

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

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

[21]:
```
# Write your code below and press Shift+Enter to execute
album_sales_dict.keys()
```

[21]:
```
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:

[22]:
```
# Write your code below and press Shift+Enter to execute
album_sales_dict.values()
```

[22]:
```
dict_values([50, 30, 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 to learn how to share your work.

Get IBM Watson Studio free of charge!

```
<p><a href="https://cocl.us/bottemNotebooksPython101Coursera"><img src="https://s3-api.us-geo.
```

About the Authors:

Joseph Santarcangelo 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