TechOrigin

# Intro to Python

## Access this notebook on GITHUB or COLAB

TECHNOLOGY SCHOOL 1.0 (2022)
DATA ANALYSIS AND DATA VISUALIZATION

# Table of Contents

## Click on the links to go directly to specific sections on the notebook.

Estimated time needed: **50 min**

---

## Python - Let's get to work now!</h1>

**Welcome!** This series of notebook will teach you to deepen your knowledge of Python programming language. You can go a step further by doing more than you have learnt here ?.

### Tuples

Consider the following tuple:

In [ ]:

```
# Exercise

genres_tuple = ("pop", "rock", "soul", "hard rock", "soft rock", \
                "R&B", "progressive rock", "disco")
```

# Exercise

Loading [MathJax]/extensions/Safe.js

1. Find the length of the tuple, `genres_tuple`:
2. Access the element, with respect to index 3:
3. Use slicing to obtain indexes 3, 4 and 5:
4. Find the first two elements of the tuple `genres_tuple`:
5. Find the first index of `"disco"`:
6. Generate a sorted List from the Tuple `C_tuple=(-5, 1, -3)`:

In [ ]:

```
# Write your code below and press Shift+Enter to execute
len(genres_tuple)
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
genres_tuple[3]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
genres_tuple[3:6]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
genres_tuple[0:2]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
genres_tuple.index("disco")
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
C_tuple = (-5,1,-3)
NewC_tuple = sorted(C_tuple)
NewC_tuple
```

### Lists

# Exercise

1. Create a list `a_list`, with the following elements `1`, `hello`, `[1,2,3]` and `True`.
2. Find the value stored at index 1 of `a_list`.
3. Retrieve the elements stored at index 1, 2 and 3 of `a_list`.
4. Concatenate the following lists `A = [1, 'a']` and `B = [2, 1, 'd']`:

In [ ]:

```
# Write your code below and press Shift+Enter to execute
a_list = [1, 'hello', [1,2,3], True]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
a_list[1]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
a_list [1:4]
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
A = [1, 'a']
B = [2, 1, 'd' ]

A + B
```

In [ ]:

```

# Exercise

B=["a","b","c"]
```

## Exercises

1. Make a list of 10 elements and select only the last 2 elements
2. Take that same list of 10 elements and select every other element starting with the very first element.
3. Select every other element starting with the second element.

In [ ]:

```
ordinary_list = ['chiamaka', 'chisom', 'chineye', 'chioma', 'chika', 'chimezie',
                 'chidigo', 'chikadibia', 'chinonso', 'chidinmma']
```

In [ ]:

```
print(ordinary_list[-2:])
```

In [ ]:

```
print(ordinary_list[::2])
```

In [ ]:

```
print(ordinary_list[2::])
```

### Sets

# Exercise

1. Convert the list `['rap','house','electronic music', 'rap']` to a set:
2. Consider the list `A = [1, 2, 2, 1]` and set `B = set([1, 2, 2, 1])`, does `sum(A) = sum(B)`
3. Create a new set `album_set3` that is the union of `album_set1` and `album_set2`:
4. Find out if `album_set1` is a subset of `album_set3`:

In [ ]:

```

```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
set(['rap','house','electronic music','rap'])
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
A = [1, 2, 2, 1]
B = set([1, 2, 2, 1])
print("the sum of A is:", sum(A))
f B is:", sum(B))
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute

album_set1 = set(["Thriller", 'AC/DC', 'Back in Black'])
album_set2 = set([ "AC/DC", "Back in Black", "The Dark Side of the Moon"])

album_set3 = album_set1.union(album_set2)
album_set3
```

In [ ]:

```
# Write your code below and press Shift+Enter to execute
album_set1.issubset(album_set3)
```

# Exercise

1. Cast the following list ['A','B','C','A','B','C'] to a set:
2. Add the string `'D'` to the set S={'A','B','C'}
3. Find the intersection of set `A={1,2,3,4,5}` and `B={1,3,9, 12}`
4. When should I use a `set` instead of a `list`?
5. What is an example of a problem where a `set` might be part of the solution?

In [ ]:

```
a = ['A','B','C','A','B','C']
b = set(a)
b
```

In [ ]:

```
S={'A','B','C'}
S.add('D')
S
```

In [ ]:

```
A={1,2,3,4,5}
B={1,3,9,12}
A & B
```

- A set enable you assign permanent memory space to a variabele while a list cannot.
- A set allows you compare directly between two sets while a set cannot

Assignment of random slots to a from a wide range of options.

### Dictionaries

In [ ]:

```
# Question sample dictionary

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

# Exercise

1. In the dictionary `soundtrack_dict` what are the keys ?

Loading [MathJax]/extensions/Safe.js

2. In the dictionary `soundtrack_dict` what are the values ?

In [ ]:

```
soundtrack_dic.keys()
```

# Exercise

1. Create a dictionary `album_sales_dict` where the keys are the album name and the sales in millions are the values from your class notebook.
2. album_sales_dict["Thriller"]
3. Find the names of the albums from the dictionary using the method `keys` :
4. Find the names of the recording sales from the dictionary using the method `values` :

In [ ]:

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

In [ ]:

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

In [ ]:

```
album_sales_dict.keys()
```

In [ ]:

```
album_sales_dict.values()
```

## About this Instructor:

</div>

ChisomLoius is very passionate about Data Analysis and Machine Learning and does lot of free lance teaching and learning. Holding a B.Eng. in Petroleum Engineering, my focused is leveraging the knowledge of Data Science and Machine Learning to help build solutions in Education and High Tech Security. I currently work as a Petrochemist.

#
M
Ir

Visit our website, or further enquire more informatio via our email.

---

Loading [MathJax]/extensions/Safe.js

Loading [MathJax]/extensions/Safe.js