Python Workshop

Presentation by Mohamed Amine Hammane

Table of content

- ## 1.1: Introduction to python
  1. What is python?
  2. How important is python nowaday!?
  3. Workshop requirements.

- ## 1.2: Let's start with python !
  1. First script (Hello world!)
  2. Variables and operations in python.
     - ❖ Variables.
     - ❖ Arithmetic operations in floats and integers.
     - ❖ String operations.
     - ❖ Boolean expressions.
  3. Input and output.
     - ❖ Input function in python
     - ❖ Print function in python
     - ❖ Open function in python

- 1.3: conditions and loops !
  1. If and else.
  2. For.
  3. While.
- 1.4: Data structure
  1. Lists in python.
  2. Dictionaries in Python.
  3. Tuples in Python.
- 1.5: Functions:
  1. Functions
  2. Function's arguments
  3. Lambda

Chapter 1: Introduction to python !

1. What is python?
2. How important is python nowadays!?
3. Workshop requirements.

What is Python?

Python is a [high-level](#), [object-oriented](#), [interpreted](#) programming language.
It is high-level In built data structure
Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Debugging Python programs is easy: Python has one of the best errors traceback. Returned errors in python are characterized also with descriptions (a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception)

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step.

How important is python nowadays!?

The importance of python nowadays comes on how much it is useful, fast and widely used in most of IT and IoT branches.
Basically, python may be used to program anything you think of.
Python can be used for:

- **AI and machine learning**
- **Data analytics**
- **Data visualization**
- **Programming applications**
- **Web development**
- **Game development**
- **Language development**
- **Cyber Security**

Most of hiring companies demands python as an important skill.

Python workshop requirements

IDE configured to work with python
I recommend:
Pycharm, Sublime text 3
Visual studio code
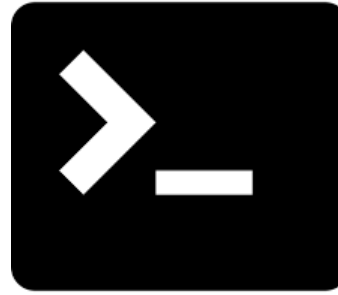
Console to work directly

Python 3.7>=
IPython

Chapter 2: Let's start with python !

```
1. First script (Hello world!)
2. Variables and operations in python.
3. Input and output.
```
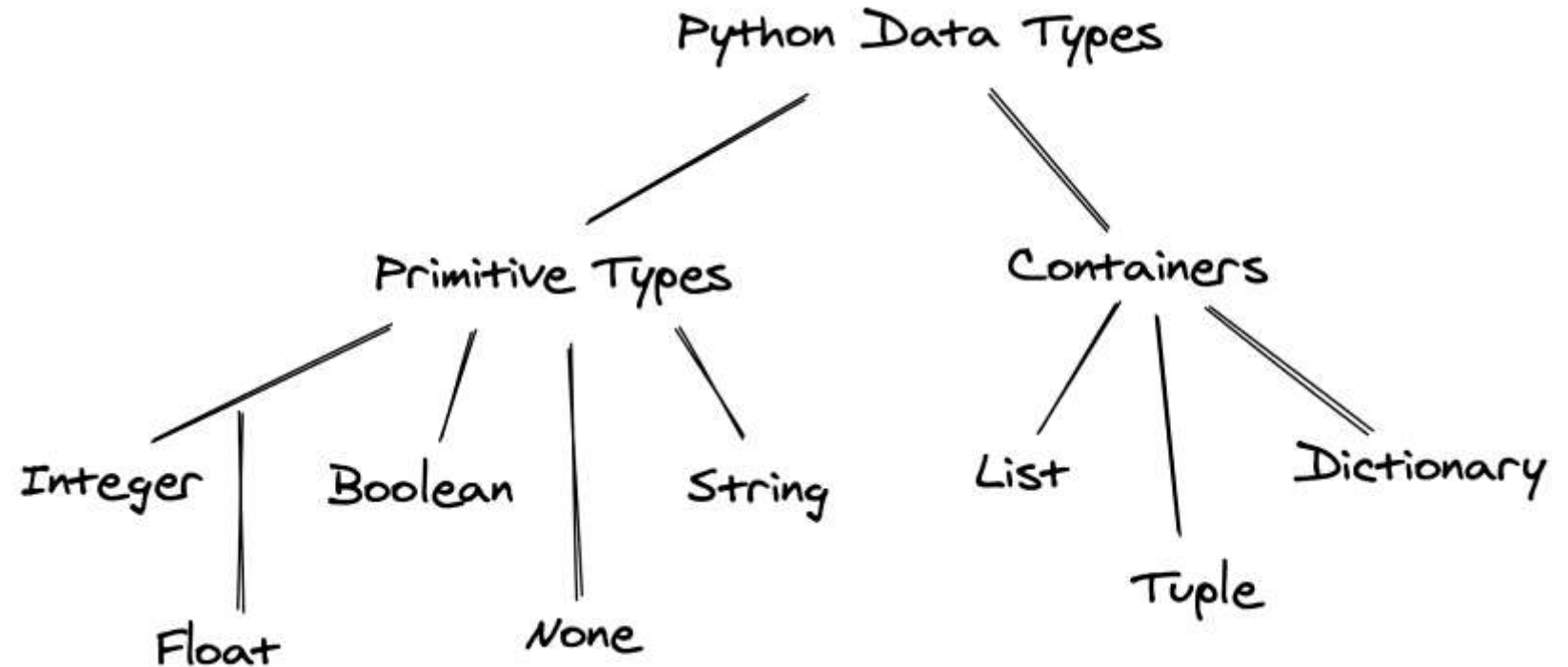
Hello world!

Simple line: print("Hello world!")

Save file with .py extention

Variables:

There is two types
of variable in
python.
- Primitive
- containers

Python Data Types

Primitive Types

Integer

Float

Boolean

None

String

Containers

List

Tuple

Dictionary

Check : https://www.datacamp.com/community/tutorials/data-structures-python

Primitive variables

In python, declaring a `variable` doesn't need any indication of used type.
In fact, you can declare a variable with a `type and change` it by changing the value in the next line.
This is how we declare different types of variable in python

Strings may be declared between two double or single quotes .

Integers are declared by numbers with no decimal points.

Floats presented with giving the floating side of a number.

Boolean expressions take two values (True or False).


```
IPython: C:Windows/system32

In [13]: name = "mohamed" #string

In [14]: age = 19 #integer

In [15]: iq = 170.0 #flaot

In [16]: lying = True #boolean

In [17]:
```

Arithmetic operations in floats and integers

Operation symbol	Operation name	example
+	Addition	x+y
-	Subtraction	x-y
*	Multiplication	x*y
**	Exponentiation	x**y
/	Division	x/y
//	Integer division	x//y
%	Rest of division (modulo)	x%y

```
In [8]: x + y
Out[8]: 8

In [9]: x - y
Out[9]: 2

In [10]: x * y
Out[10]: 15

In [11]: x ** y
Out[11]: 125

In [12]: x / y
Out[12]: 1.6666666666666667

In [13]: x // y
Out[13]: 1

In [14]: x % y
Out[14]: 2

In [15]:
```

With x = 5 and y = 3

String operations

❖ methods

When you declare a variable in C, the variable is assigned to a specific memory location with enough space. But python doesn't work this way. Variables in Python are objects that don't have specific space. They also have functions that change the value of the variable.

Example:
str().replace()


```
IPython 7.31.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: pro = "trying to delete all 't's in this sentence"

In [2]: deleted = pro.replace("t","")

In [3]: deleted
Out[3]: "rying o delee all ''s in his senence"

In [4]: pro
Out[4]: "trying to delete all 't's in this sentence"

In [5]: _
```

String operations

❖ Addition and multiplication in python

Python had made it easy to programmers to add characters into strings, manipulate it and use multiple functions. So how we can do that?

You can simply do that with arthimetic operations symbols "+" and "*".

Example:
"I shall ask for payment,"+" not kidding"
This will return the following:
"I shall ask for payment, not kidding"


```
IPython: C:Users/ZeD_OnE

In [29]: word = "HCn1 "

In [30]: word*5
Out[30]: 'HCn1 HCn1 HCn1 HCn1 HCn1 '

In [31]: word + "Hamouda"
Out[31]: 'HCn1 Hamouda'

In [32]: word + "never ask me about my weird examples"
Out[32]: 'HCn1 never ask me about my weird examples'

In [33]:
```

Input and output

input

This function is used to receive values and initialize them into a variable.

```
In [39]: p = input()
15

In [40]: p
Out[40]: '15'

In [41]: p = input("here a text printed before taking the value : ")
here a text printed before taking the value : 15

In [42]: p
Out[42]: '15'

In [43]: type(p)
Out[43]: str

In [44]: p = int(input("here a text printed before taking the value : "))
here a text printed before taking the value : 15

In [45]: p
Out[45]: 15

In [46]: type(p)
Out[46]: int

In [47]:
```

print

This function is used to print variables value and textes to the console.

```
In [34]: variables = 15

In [35]: print(15)
15

In [36]: print(variables)
15

In [37]: print("variables value is ",variables)
variables value is   15

In [38]:
```

open

We may use open() to open files to read or write data to them

```
]: open("dude.txt","a+").write("hahaha")
]: 6

]:
```

Bonus!

docstrings

A Python docstring is a string used to document a Python module, class, function or method, so programmers can understand what it does without having to read the details of the implementation.

comments

A comment in Python starts with the hash character, # , and extends to the end of the physical line

Chapter 3: conditions and loops !

1. If and else.
2. For loop.
3. While loop.

Conditions

Python had replaced backets that identify the beginning and the end of if statements and loops with spaces. so, it basically checks the beginning and the and of spaces to determine the end of statements.
This is how we write and if statement in Python

❖ If

```
if condition:
        print("it is True")
(edited)
```

❖ else

```
else:
        print("all conditions are false")
```

❖ elif

```
elif conditionNbr2:
        print("conditionNbr2 True")
```

Loops

❖ For loop:

For loop syntax in Python doesn't look like any in other programming language but it does mostly the same thing. For loop uses two key words `in` and `range`.

range is a function that returns a list of integers from 0 to n-1 for range(n) in is such a complicated command used by Python. What it really do is scraping the list of integers or any kind of lists
We can present for loop in two ways:

```
HCn1 ZeD_OnE  Today at 12:42 PM
for i in range(50):
        print("all conditions are false")
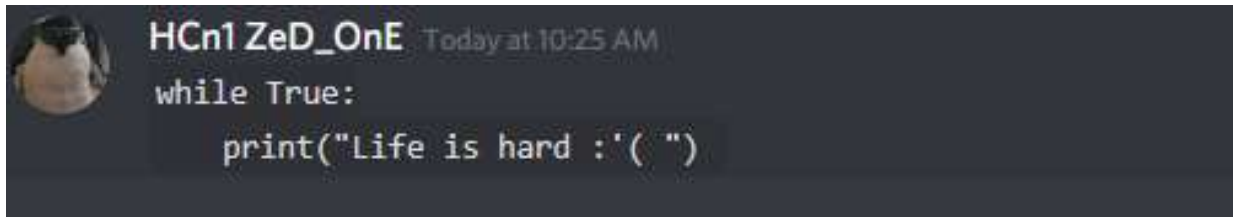```

Scarping a list were made
By range function.

```
for i in ["1",12,11,"31"]:
        print("all conditions are false")
```

Scarping manually made
List.

Loops

❖ while loop:

While instruction is so simple. It doesn't look so different then other programming languages. Here is the structure

BONUS: use break statement to break out of loop.

Example:
```
s=0
while True:
    s=s+1
    if s == 5:
        break
```

Welcome among us junior Python developer

Let's Have fun !

Chapter 4: Data structure

1. Lists in Python
2. Dictionaries in Python
3. Arrays in Python

Lists in Python

Lists are variables that used to store data of multiple types. Every data in those lists have a specific index to it. Python lists don't require to specify a length to it, you may append value to it and the length will change

Example:
data = [1,2,3,"Mohamed",[1,2,3]] #this list includes another list, integer and a string.
data.append(2.5) #this list includes another list, integer and a string and a float.

Lists in python are objects as well. They have multiple methods to use with. Most important of them:

sort() #sort a list of int from the smallest to the biggest.
remove() #remove an element in list based in its index.
index() #return an index of a list element's value.

Let's work with lists

Dictionaries in Python

Dictionaries are container type variables that stores value by a defined key in a JSON form. Example:
```
Example = {
  'name': {'data': 'value'},
  'name2': {'data':'value2'},
  'ID': 2
}
```

Let's try to access 'value'
We write:
```
value = Example['name']['data']
```

Basically we take the value of defined key 'name' which is {'data' : 'value'} in a dictionary type.
Then we took the value of 'data' from the first grabbed value.

Tuples in Python

Tuples are also variables that stores data that can't be change the moment you set it up.
Example = ('Mohamed',2)

We can't append nor overwrite an elements in a tuple. But the on variable is possible
For ex:
Example = (21,12)
Example = (12,21)

Task: make an encryption with Python !

Spy encryption test

Task: make a stars table with Python !

Stars table game

Chapter 4: Functions !

Functions

Functions are some instructions and code lines program use to avoid repeating it again and again.
For example. We calculate product of numbers in line 10, 112 and 510, we make a function only one and call it every time we need it.
For example:

```
def Message():
    print('I know I am not funny')
Message()
```

We may call a function as much as we want and we can ask it for return or multiple returns for example:

```
def Message():
    print('I know I am not funny')
    return 0
Returning = Message()
```

Function's arguments

Sometimes functions need some users inputs to do certain actions. This inputs are called arguments. The programmer may specify a default argument value and argument type
Look to the next example:

```
def functionOne(a,b,c):
    return a+b+c;
def functionTwo(a=0,b=0,c=0):
    return a+b+c;
def functionThree(a:int,b:int,c:int):
    return a+b+c;
```

In all this functions the return is one whatever is a and b or c. only few changes made we are going to talk about.

Task: make quadratic equation solver !

Let's develop functions !

Lambda

Python had made developing mathematic functions so easy and fast with the known method 'lambda'.
Lambda is a method used to make small anonymous functions (mostly mathematic functions).
Syntax: lambda *arguments : expression*
Some examples:

```python
y = lambda a : a + 10
x = lambda a, b : a * b
print(x(5, 6),y)
```

We may inherit a lambda inside a function as well, for example:

```python
def exp(y):
    x = lambda a:a**2
    return x(y);
exp(10)
```

Payment time !