Tutorial 1 : Python basics and more

In this first tutorial, we will learn how to write simple programs in python. The objective of this first tutorial is to familiarize yourself with the development environment and the syntax of this language. While resolving some simple algorithmic problems.

Environment setting :

- Download and install python 3.6 or a higher version : https://www.python.org/downloads/
- Install an IDE of your choice : Pycharm, Xcode, Atom, Visual Studio Code, …
- All the python programs should be structured in a project and stored in python files (.py) (not only executed in the python shell
- Remember that python.org contains the documentation and the specification of python
- Stack overflow has probably the answer to everything you need

Exercise 1 :

1. Variables

Variables are containers for storing data values. Unlike other programming languages, Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

a. Declare 2 variables x and y having values 2 and “Pierre”
b. Print the values of these variables
c. Print the type of the x and y
d. Change the value of x to “Marie” and print again its type

2. Lists, Sets, Dictionaries

There are four collection data types in the Python programming language:

- List is a collection which is ordered and changeable. Allows duplicate members.
- Tuple is a collection which is ordered and unchangeable. Allows duplicate members.
- Set is a collection which is unordered and unindexed. No duplicate members.
- Dictionary is a collection which is unordered, changeable and indexed. No duplicate members.

More information : https://docs.python.org/3.6/library/stdtypes.html#sequence-types-list-tuple-range

3. Conditions and loops

Python supports the usual logical conditions from mathematics, these conditions can be used in several ways, most commonly in if statements and loops.

An if statement is written by using the if keyword.

a. Declare two variables a and b with different values

b. Write an if statement to test whether b is greater than a
c. Print a message according to the results of the test

Loops : Given an integer A. For all non-negative integers i < N, print i^2. Do it using different iterators

4. Functions

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

5. Problem 1 :

Given a time in 12 hour AM/PM format, convert it to military 24 hour time.
Note: Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock. Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Function Description :
Complete the timeConversion function who should return a new string representing the input time in 24 hour format.

The timeConversion function has the following inputs and outputs:

- s: a string representing time in 12 hour format
- Input format : a single string containing a time in 12 hour clock format (i.e. hh:mm:ssAM or hh:mm:ssPM), where 01=<hh=<12 and 00=<mm,ss=<59.
- Output Format : Convert and print the given time in 24 hour format, where 00=<hh=<24
- Example : Input : 07:05:45PM => Output : 19:05:45

6. Problem 2 :

Given two numbers A and B. Iterate from the smallest to the biggest number and print the sum of the current number and previous number at each iteration. At the end, print the sum of all the intermediate calculated numbers.

Example :
Input : A=2, B=4
⇨ Print 3 ; 2+1
⇨ Print 5; 3+2
⇨ Print 7
⇨ End : Print 15
Tips :

- range and sum functions

7. Problem 3 :

Write a function that take a string as input and display only the characters which are present at an even index

- Example : The input "python" should print ‘p’, ‘t’, ‘o’.

8. Problem 4

Write a function that takes as input a long string containing multiple words. Print back the same string, except with the words in backwards order.

- Example: Input string “My name is Michele” => Output : The string “Michele is name My”

9. Problem 5

Take a list, say for example this one: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a function that prints out all the elements of the list that are less than 5.

Extras:

Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.

Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.