Python exercises

Academic course 2022/2023

Prepared by: Guillermo Bejarano Pellicer

Date: January 9th, 2023

1. Indications

Every exercise will be solved in an **independent script**, called ***ExerciseX.py***, being *X* the number of the corresponding exercise in the following list.

All exercises will be included in a compressed folder called ***PythonFirstnameLastname.zip***, being *Firstname* the student's first name, and *Lastname* the student's family name. This compressed folder will be submitted using the corresponding task in Moodle.

2. Assessment

- If the student does not correctly interpret a given exercise, the corresponding mark will be 0.
- Every exercise may be correct (10), mediocre (5), or incorrect (0).
- The final mark will be proportional to the number of exercises correctly solved. The first 17 exercises weigh 90% of the global mark, while the last one weighs 10%.
- If cheating is detected, all involved students' marks will be 0.

3. List of exercises

1) Create a list of 10 disorganised elements and sort it.

2) Create the following list, L1 = ['a', 'c', 'd', 'a', 'l', 'm']:
   a. Use a method to find out how many times the list contains the 'a' character.
   b. Use a method to delete the 'd' character from the list.
   c. Use a method to insert the 'p' character at the second position of the list.
   d. Use a method to sort the list.

3) Use the ***split*** method to split a given paragraph into a list of words. Print the list of words on the screen.

4) Use the ***strip*** method to convert a string expressed as an URL (for instance, "www.marca.com") into the equivalent string without "www." and ".com". Note: use more than one line of code if necessary.

5) Create a dictionary called *pizza*, which contains the following *keys*:
   Name: a certain string.
   Ingredients: a list with all ingredients.
   Price: a floating numeric value.

   Use a method to create a list including only the values of the *keys*.

6) Using the ***input***() command and the ***if-else*** sentences, determine if the first number of your DNI is greater or smaller than the last one.

7) Create a list with the numbers of your DNI. Using a ***for*** loop and the ***if*** sentence, determine the highest number and its position within the DNI. If there are two or more positions featuring the highest number, indicate all of them.

8) A palindrome is a word which reads the same backward as forward, for example:

1. Aba
2. Aérea
3. Allá
4. Ama
5. Ana
6. Anana
7. Arañara
8. Arenera
9. Arepera
10. Ata

Create a script that checks if a given word is a palindrome (***True***) or not (***False***), without considering the accents. Note: use ***break*** to exit the loop.

9) Create the ***subtract*** function and make the data to be received from the terminal.

10) Create a function that determines if two lists are identical. Two lists are identical only if their lengths are the same and their elements at every position are the same.

11) Create a function that receives an integer number as its input and returns ***True*** if the number is prime and ***False*** if it is not.

12) Create a function that receives a list of floating numbers as its input and returns the sum of all its elements.

13) Based on the function created in Exercise (12), create a function that multiplies all the elements located at odd positions of the list. The function should work for any list length. The first position of the list (0 position) will be considered as odd.

14) Create a function that receives a list of floating numbers as its input and returns the product of all its elements.

15) Given the following list: ["a", 1, [1,2], 2, (1,2), 3], create a function that returns a list including only the integer elements from the original list.

16) There is no method implementing the comparison between a list and a tuple in Python. Create a function that receives a list and a tuple as its inputs and returns ***True*** if the tuple is greater than the list and ***False*** otherwise. The list and

the tuple must only contain either numbers or characters. The comparison should work in the same way as the comparison between only lists and only tuples.

17) Create a *lambda* function that computes the square of a number.

18) Create a class that represents a student. The class variables will be:
    a. Age (integer)
    b. Repeating (boolean)
    c. E-mail (string)
    d. Marks (list of three elements, every one representing the mark of one of the partial exams, assuming that there are three)
    e. Absences (integer, indicating the times that the student missed classes)

    Create the following methods:
    a. Method that returns the student's average mark.
    b. Method that indicates whether the student has missed classes more than 5 times.

    Once that the class is created, define a list of three students, including at least an average failing grade. Using a loop, go through the list of students and print the failing student's data separated by commas to a text file.