SCALE FOR PROJECT

PISCINE PYTHON FOR DATA SCIENCE / DAY 04

Introduction

The methodology of School 21 makes sense only if peer-to-peer assessments are done seriously. This document will help you to do it properly.

- Please, stay courteous, polite, respectful and constructive in all communications during this assessment. The bond of trust between community 21 and you depends on it.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.

Guidelines

- Evaluate only the files that are on the GIT repository of the student or group.
- Doublecheck that the GIT repository is the one corresponding to the student or the group as long as to the project.
- Meticulously check that nothing malicious has been used to mislead you and have you assess something except the content of the official repository.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.
- Use the special flags in the scale to report an empty or non-functional solution as long as a case of cheating. In these cases, the assessment is completed and the final grade is 0 (or in a case of cheating is -42). However, except for a case of cheating, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them for the next assessment.

- You must stop giving points from the first wrong exercise even if the following exercises are correct.

Attachments

- The exercises

Preliminaries

Respect the rules:

- The repository contains the work of the student (or group).
- The student is able to explain their work at any time during the assessment.
- The general rules and any rules specific to the day are respected throughout the assessment.

Yes | No

Piscine Python/Data Science D04

- Any hardcoded result is worth zero for the exercise.

Exercise 00 – List comprehensions

- The script has two functions: one with a loop and append and the other with a list comprehension?
- timeit calls the functions 90 000 000 times?
- The time spent by both functions is displayed ascendingly in the string?
- Ask to modify the script: it should give the results of both functions. Are they correct and identical?

In all other cases, the test is failed.

Yes | No

Exercise 01 – Map

- The script has three functions: one with a loop and append, another with a list comprehension, and the other with a map?
- timeit calls the functions 90 000 000 times?
- The time spent by all the functions is displayed ascendingly in the string?
- Ask to modify the script: it should give the results of all the functions. Are they correct? `map()` will have None inside the list.

In all other cases, the test is failed.

Yes | No

Exercise 02 – Filter

- The script has four functions: one with a loop and append, another with a list comprehension, another with a map, and the other with a filter?
- Ask to modify the script: it should give the results of any function. Try all the functions. Are their results correct and identical?
- If you give a wrong function name, does the script make an exception?

In all other cases, the test is failed.

Yes | No

Exercise 03 – Reduce

- The script has two functions: one with a loop and sum, another with a reduce?
- Ask to modify the script: it should give the results of any function. Try all the functions. Are their results correct and identical?
- If you give a wrong function name, does the script make an exception?

In all other cases, the test is failed.

Yes | No

Exercise 04 – Counter

- The script has two functions: one with a dict of unique values and their counts, another with the top 10 numbers?
- Ask to modify the script: it should give the results of all 2 functions. Are the results of corresponding functions correct and identical?

In all other cases, the test is failed.

Yes | No

Exercise 05 – Generator

- the first script reads all the file lines into a list and then returns it?
- the second uses a generator and yields?
- both scripts show Peak Memory Usage in GB?
- both scripts show User Mode Time + System Mode Time in seconds?

In all other cases, the test is failed.

Yes | No

Ratings

Don’t forget to check the flags corresponding to the review.

Ok | Outstanding project
Empty work | Incomplete work | Cheat | Crash | Forbidden function

Conclusion

Leave a comment on this evaluation.