SCALE FOR PROJECT

PISCINE PYTHON FOR DATA SCIENCE / DAY 00

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 depend 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

Pool Python/Data Science D00

- Any hardcoded result is worth zero for the exercise.

Exercise 00 – First shell script

- files `hh.sh` and `hh.json` exist in the directory of the repository?
- `hh.sh` is executable?

- `hh.sh` stores the file hh.json as a result of execution?
- `hh.json` contains information about exactly 20 vacancies?
- `hh.sh` works with different keywords for the search as the argument?
- `hh.json` formatted in the way that each field of JSON is placed on a different line?

In all other cases, the exercise is wrong.

Yes | No

Exercise 01 – Transforming JSON to CSV

- files `filter.jq` and `hh.csv` exist in the directory of the repository?
- `filter.jq` produces file hh.csv?
- `hh.csv` contains only specified 5 columns?
- `hh.csv` contains all the values in any row delimited by a comma?
- `hh.csv` contains the headers?
- nothing except authorized function is used to produce the result?

In all other cases, the exercise is wrong.

Yes | No

Exercise 02 – Sorting a file

- files `sorter.sh` and `hh_sorted.csv` exist in the directory of the repository?
- `sorter.sh` is executable?
- `sorter.sh` stores the file `hh_sorted.csv` as a result of execution?
- `hh_sorted.csv` is sorted correctly?
- `hh_sorted.csv` is a valid CSV file?

- `hh_sorted.csv` contains the headers?
- nothing except authorized function is used to produce the result?

In all other cases, the exercise is wrong.

Yes | No

Exercise 03 – Replacing string in a file

- files `cleaner.sh` and `hh_positions.csv` exist in the directory of the repository?
- `cleaner.sh` is executable?
- `cleaner.sh` stores the file `hh_positions.csv` as a result of execution?
- `hh_positions.csv` is sorted correctly?
- `hh_positions.csv` is a valid CSV file?
- `hh_positions.csv` contains the headers?
- string “Data Scientist” is removed accordingly to the exercise?
- string “Data Scientist” is kept accordingly to the exercise?
- nothing except authorized function is used to produce the result?

In all other cases, the exercise is wrong.

Yes | No

Exercise 04 – Descriptive statistics

- files `counter.sh` and `hh_uniq_positions.csv` exist in the directory of the repository?
- `counter.sh` is executable?
- `counter.sh` stores the file `hh_uniq_positions.csv` as a result of execution?
- `hh_uniq_positions.csv` is sorted correctly?

- `hh_uniq_positions.csv` is a valid CSV file?
- `hh_uniq_positions.csv` contains the headers?
- nothing except authorized function is used to produce the result?

In all other cases, the exercise is wrong.

Yes | No

Exercise 05 – Partitioning and concatenation

- files `partitioner.sh` and `concatenator.sh` exist in the directory of the repository?
- `partitioner.sh` and `concatenator.sh` is executable?
- `partitioner.sh` stores the files with correct names corresponding to the dates?
- each file produced by `partitioner.sh` contains only the dates that correspond to the names of the files?
- each file produced by `partitioner.sh` is a valid CSV file?
- each file produced by `partitioner.sh` contains the headers?
- `concatenator.sh` takes as input all the files of `partitioner.sh`?
- the result of `concatenator.sh` is identical to the result of Exercise 03?
- nothing except authorized function is used to produce the result?

In all other cases, the exercise is wrong.

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.