SCALE FOR PROJECT PYTHON - 2 - DATATABLE (/PROJECTS/PYTHON-2-DATATABLE)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere.42yerevan.am:vogsphere/intra-uuid-16795abc-f62b-4
```

Introduction

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the person (or the group) evaluated the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.

- You must consider that there might be some difference in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used

to facilitate the grading.

- If the evaluating student has not completed that particular
project yet, it is mandatory for this student to read the
entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository,
non-functioning program, cheating, and so forth.
In these cases, the grading is over and the final grade is 0,
or -42 in case of cheating. However, except the exception of cheating, you
are encouraged to continue to discuss your work even if the later is in
progress in order to identify any issues that may have caused the project
failure and avoid repeating the same mistake in the future.

- Remember that for the duration of the defense, no other unexpected,
premature, or uncontrolled termination of the program, else the final
grade is 0 for the exercise, and continue the evaluation.

- You should never have to edit any file except the configuration file if
the latter exists. If you want to edit a file, take the time to explain why
with the evaluated student and make sure both of you agree on this.

- Lib imports must be explicit, for example importing "from pandas import *"
is not allowed, you must put 0 to the exercise and continue the evaluation.

- Your exercises are going to be evaluated by other students,
make sure that your variable names and function names are appropriate and civil.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/82033/en.subject.pdf)

population_total.csv (https://cdn.intra.42.fr/document/document/15948/population_total.csv)

life_expectancy_years.csv (https://cdn.intra.42.fr/document/document/15949/life_expectancy_years.csv)

income_per_person_gdppercapita_ppp_inflation_adjusted.csv
(https://cdn.intra.42.fr/document/document/15950/income_per_person_gdppercapita_ppp_inflation_adjusted.csv)

Mandatory Part

Error Management

Carry out AT LEAST the following tests to try to stress the error
management

- The repository isn't empty.
- No cheating.

- No forbidden function/library.
- There is no global variable.
- The executable is named as expected.
- Norminette shows no errors. (pip install flake8, alias norminette=flake8, use flag Norme)
- Your lib imports must be explicit, for example you must "import numpy as np". (Importing "from pandas import *" is not allowed, and you will get 0 on the exercise.)
- If an exercise is wrong, go to the next one.

☑ Yes ☒ No

---

ex00 Load csv

Test to load the files 'life_expectancy_years.csv' or
'population_total.csv', it must work.
Then you can test with bad path or bad file format, the function must handle
the errors in a clean way.

Your script tester:

```
from load_csv import load

print(load("life_expectancy_years.csv"))
```

Expected output:

```
$> python tester.py
Loading dataset of dimensions  (195, 302)
country  1800  1801  1802  1803  ...  2096  2097  2098  2099  2100
Afghanistan  28.2  28.2  28.2  28.2  ...  76.2  76.4  76.5  76.6  76.8
  ...
```

☑ Yes ☒ No

---

ex01 draw my country

Look in the subject the expected result to this exercise,
the graph must display the Life expectancy projection of France or the country of your campus.
The graph must have a title and a legend for each axis.

☑ Yes ☒ No

---

ex02 compare my country

Look in the subject the expected result to this exercise,
the graph must display the population projection of France or the country
of your campus versus other country of your choice.
The graph must have a title and a legend for each axis.

☑ Yes ☒ No

ex03 draw my year

Look in the subject the expected result to this exercise,
The graph should show the projection of life expectancy to the gross domectic product of the year 1900 for each country.
The graph must have a title and a legend for each axis.

☑ Yes ☒ No

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok ★ Outstanding project

Empty work | Incomplete work | Norme | Cheat | Crash | Concerning situation | Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation