SCALE FOR PROJECT PYTHON - 4 - DOD (/PROJECTS/PYTHON-4-DOD)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere.42yerevan.am:vogsphere/intra-uuid-8c4e1223-2e
```

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.

Copied

- 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/82035/en.subject.pdf)

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.

Copied

☑ Yes ☒ No

---

ex00 Calculate my statistics

The function must take as argument an unknown list of numbers and return the mean, median, Quartile (25% and 75%) Standard Deviation or Variance according to the **kwargs asked.

Your script tester:

```
from statistics import ft_statistics

ft_statistics(42, 115, 30, 151, 6400, toto="mean", tutu="std", tata="quar
tile", titi="djdj")
print("-----")
ft_statistics(51, 735, 455, 8, 7, 74, 75, hello="median", world="var")
print("-----")
ft_statistics(5, 75, 450, 18, 597, 27474, 48575, ejfhhe='fdff', ejdjdejn
="demiane")
print("-----")
ft_statistics(hello="std", world="median")
```

expected output:

```
$> python tester.py
mean : 1347.6
std : 2526.600926145639
quartile : [42.0, 151.0]
-----
median : 74
var : 68437.3469387755
-----
-----
ERROR
ERROR
```

☑ Yes ☒ No

---

ex01 Outer inner

The function must take as argument a number and a function to execute, then execute the function with the number kept in memory
You should only give valid calculations, the goal of the exercise is to understand the closures, not to handle errors.

Copied

Your script tester:

```
from in_out import outer, square, pow

def cube(x: int | float) -> int | float:
    """Returns the cube of x"""
    return (x * x * x)

my_counter = outer(6, square)
print(my_counter())
print(my_counter())
print(my_counter())
print("---")
another_counter = outer(1.6, pow)
print(another_counter())
print(another_counter())
print(another_counter())
print("---")
again_another_counter = outer(4, cube)
print(again_another_counter())
print(again_another_counter())
print(again_another_counter())
```

expected output:

```
$> python tester.py
36.0
1296.0
1679616.0
---
2.1212505710975917
4.929279084950403
2599.697171916239
---
64
262144
18014398509481984
```

Yes No

ex02 My first decorating

Copied

The CallLimit function takes an int as argument and blocks the execution of the function when the number of calls to itself exceeds the limit.

Your script tester.py:

```
from callLimit import callLimit

@callLimit(0)
def h():
    print ("h()")

@callLimit(2)
def j():
    print ("j()")

@callLimit(1)
def k():
    print ("k()")

for i in range(2):
    h()
    j()
    k()
```

expected output:

```
$> python tester.py
Error:  <function h at 0x7fb993152f70> call too many times
j()
k()
Error:  <function h at 0x7fb993152f70> call too many times
j()
Error:  <function k at 0x7fb9930f01f0> call too many times
```

☑ Yes ☒ No

ex03 data class

The data class that takes as Arguments a name and a nickname, create the student's login, and generate a random ID with the generate_id function.
You can change the active state, pass it to False.

```
from new_student import Student

student = Student(name = "Edmund", surname = "agle")
print(student)
```

Copied

expected output: (id is random)

```
$> python tester.py
Student(name='Edmund', surname='agle', active=True, login='Eagle', id='hh
kjdjtxeccjbxh')
$>
```

The login and id should not be initializable and must return an error.
Your script tester.py:

```
from new_student import Student

student = Student(name = "Edward", surname = "agle", login = "pigeon")
print(student)
```

expected output:

```
$> python tester.py
...
TypeError: Student.__init__() got an unexpected keyword argument 'login'
$>
```

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

Copied

Finish evaluation