

(https://profile.intra.42.fr)

SCALE FOR PROJECT PYTHON - 1 - ARRAY (/PROJECTS/PYTHON-1-ARRAY)

You should evaluate 1 student in this team



Git repository

git@vogsphere.42paris.fr:vogsphere/intra-uuid-2c025716-c4c1



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/88179/en.subject.pdf>)

 landscape.jpg (<https://cdn.intra.42.fr/document/document/17375/landscape.jpg>)

 animal.jpeg (<https://cdn.intra.42.fr/document/document/17376/animal.jpeg>)

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 Give my BMI

The program must calculate the BMI of each person and return True if the BMI exceeds the limit given in int.

Your tester.py:

```
from give_bmi import give_bmi, apply_limit

height = [1.71, 1.65, 1.73, 1.95, 1.63]
weight = [65.3, 58.4, 63.4, 94.5, 72.9]

bmi = give_bmi(height, weight)
print(bmi, type(bmi))
print(apply_limit(bmi, 26))
```

Expected output:

```
$> python tester.py
[22.33165760404911, 21.45087235996327, 21.183467539844298, 24.85207
100591716, 27.43799164439761] <class 'list'>
[False, False, False, False, True]
$>
```

You can test the function by sending lists not of the same size, with strings instead of int, empty lists, but do not be too cruel.

 Yes

 No

ex01 2D array

The program must take as parameters a 2D array which prints its shape, and returns a truncated version with the start and end arguments. Check in the code that the student has used the slicing method.

Your tester.py:

```
from array2D import slice_me

family = [[2.10, 78.45],
          [4.15, 6.70],
          [2.10, 98.5],
          [1.88, 75.2]]

print(slice_me(family, 0, 2))
print(slice_me(family, 1, -2))
```

Expected output:

```
$> python test_array2D.py
My shape is : (4, 2)
My new shape is : (2, 2)
[[2.1, 78.45], [4.15, 6.7]]
My shape is : (4, 2)
My new shape is : (1, 2)
[[4.15, 6.7]]
```

☒ Yes

☐ No

ex02 load my image

The program must take in parameters a path of an image and print its shape and return its array, you can test the program with images in jpeg and jpg, test also a path error and corrupted image.

Your tester.py:

```
from load_image import ft_load

print(ft_load("landscape.jpg"))
```

Expected output:

```
\begin{42console}
$> python tester.py
The shape of image is: (257, 450, 3)
[[[19 42 83]
  [23 42 84]
  [28 43 84]
  ...
  [ 0  0  0]
  [ 1  1  1]
  [ 1  1  1]]]
$>\end{42console}
```

☒ Yes☐ No

ex03 Zoom on me

Look in the subject the rendering that the image should have after the "zoom".
If the "zoom" is slightly different from that of the subject, it is not important.

```
$> python zoom.py
The shape of image is: (768, 1024, 3)
[[[120 111 132]
  [139 130 151]
  [155 146 167]
  ...
  [120 156  94]
  [119 154  90]
  [118 153  89]]]
New shape after slicing: (400, 400, 1) or (400, 400)
[[[167]
  [180]
  [194]
  ...
  [102]
  [104]
  [103]]]
$>
```

☒ Yes☐ No

ex04 rotate me

Look in the subject the rendering that the image should have after the "rotation".

If the "rotation" is slightly different from that of the subject, it is not important.

But in any case she had to make a 90 degree rotation counterclockwise and a mirror effect.

The student must do the transposition by himself, the use of `numpy.transpose` is forbidden and it is 0 to the exercise.

✓ Yes

✗ No

ex05 Pimp my image

Look in the subject how the image should look like after the 5 types of image transformation.

Look in the code if the restrictions of the subject have been respected for each function.

You give 1 point per good image transform display.

Your `tester.py`:

```
from load_image import ft_load
from pimp_image import ft_invert, ft_red, ft_green, ft_blue, ft_gre
y

array = ft_load("landscape.jpg")

ft_invert(array)
ft_red(array)
ft_green(array)
ft_blue(array)
ft_grey(array)
```

Rate it from 0 (failed) through 5 (excellent)



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

Règlement Intérieur (<https://profile.intra.42.fr/legal/terms/7>)
Declaration on the use of cookies (<https://profile.intra.42.fr/legal/terms/2>)
Privacy policy (<https://profile.intra.42.fr/legal/terms/5>)
General term of use of the site (<https://profile.intra.42.fr/legal/terms/6>)
Rules of procedure (<https://profile.intra.42.fr/legal/terms/4>)
Terms of use for video surveillance (<https://profile.intra.42.fr/legal/terms/1>)
Legal notices (<https://profile.intra.42.fr/legal/terms/3>)