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

SCALE FOR PROJECT PYTHON MODULE 03 (/PROJECTS/PYTHON-MODULE-03)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere.42paris.fr:vogsphere/intra-uuid-7738bcdb-1c07    📋
```

Comments

This fourth module is the evolution of two original projects created by the
Paris-based student organization 42 AI.
They are named Bootcamp Python and Bootcamp Machine Learning.
active members of 42 AI re-designed both of them for the school curriculum.

Bootcamps has been developped between August 2019 and March/April 2020.
Active 42AI members organized severals sessions of 2 weeks to 42Paris students
to offer them the possibility to get famliar with Python and basics concepts
of machine learning.

The success of those sections brings the pedagogy to accept the idea to integrate
the 2 bootcamps to the curriculum (initial discussion (01-05/2019) with 42 Paris
pedago team highlighted a categorical opposition/refusal to this idea)

The transcription had been realized over the direction of Matthieu David.
Several 42AI members contributed to the redaction on the correction scales.
For futur corrections on the scale, please contact the 42AI association via
contact@42ai.fr or the current 42AI pedagogical supervisor.

Introduction

The Bootcamp Python and Bootcamp Machine Learning were originally created by
[42AI](https://github.com/42-AI) active members and were adapted to 'piscine'
format for the school 42 curriculum.
For any issue or suggestion: [42paris_staff_pedagogy](https://42born2code.slack.com/archives/C7NF60E0Z)
and
[42AI](https://github.com/42-AI/bootcamp_python/issues).

As usual, you have to observe the following courtesy rules:

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

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

- You must consider that there might be some differences in the
understanding of and approach to project instructions, and the scope of
its functionalities, between you and your peers. Always remain open-minded
and grade them as fairly as possible. The pedagogy is valid only and only
if peer-evaluation is conducted seriously.

The goal of this module is to get started with the library Numpy.

Disclaimer

The serie of modules started to be produce at the time of the release of
Python 3.7. Students are free to use later version of Python as long as they
verified the producted code complies with all the aspects precised in the
subjects.
As a consequence we recommend to students to perform the modules with the
the Python version 3.7 (but this is just an advice).
Version can be checked with the command ```python -V```.

Guidelines

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

- Double-check that the GiT repository does belong to the student.
Ensure that the work is the one expected for the corrected exercise
and don't forget to verify that the command "git clone" is run in an empty
folder.

- Check carefully that no malicious aliases were used to make
you evaluate files that are not from 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 them to read the
entire subject prior to starting the defense.

- Use the flags available on this scale to signal an empty repository,

non-functioning program, a Norm error (specified next in general rules),
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.

- Use the appropriate flag.

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

- 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.

- The Norm: The PEP 8 standards is not mandatory, but recommended.

- The function eval is never allowed.

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

- Copiable code is included in the code_blocks.md in attachments.
(block xx.xx.xx) is an indication that the block of code with the
same id on top should be used

Attachments

☐ code_blocks.md (https://cdn.intra.42.fr/document/document/9754/code_blocks.md)

☐ subject.pdf (https://cdn.intra.42.fr/pdf/pdf/54996/en.subject.pdf)

☐ solar_system_census.csv (https://cdn.intra.42.fr/document/document/9755/solar_system_census.csv)

☐ 42AI.png (https://cdn.intra.42.fr/document/document/9756/42AI.png)

☐ elon_canaGAN.png (https://cdn.intra.42.fr/document/document/9757/elon_canaGAN.png)

Exercise 00 - NumPyCreator

Basic tests

To test this exercise, you can launch the follwing command on your terminal:
python3

then : (block 03.00.00)

In this python interpretor, check if all the following cases have a correct behaviour:
- (block 03.00.01)

- (block 03.00.02)

- (block 03.00.03)

- (block 03.00.04)

- (block 03.00.05)

- (block 03.00.06)

- (block 03.00.07)

- (block 03.00.08)

For values contained in identity and from_shape function, both integer and float values are acceptable.

✓ Yes   × No

Exercise 01 - ImageProcessor

Implementation

To test this exercise, you can launch the follwing command on your terminal:
python3

then : (block 03.01.00)

Verify the class ImageProcessor and that the methods 'load' and 'display'
are implemented in \"ImageProcessor.py\".

- The method 'load :
  (block 03.01.01)
  The print should display a nested list of RGB pixel numbers contained by the image (cf subject)

- The method 'display' :
  (block 03.01.02)
  The image should be displayed in a separate window when running in the console.

✓ Yes   × No

Exercise 02: ScrapBooker

Basic tests

Perform basic tests to verify the proper behaviour of the different methods.
First, launch the python interpreter by taping the following command on your terminal:
python3

then copy paste the commands and check if they have the same behaviour as here:

- (block 03.02.00)

- (block 03.02.01)

- (block 03.02.02)

- (block 03.02.03)

        ⊘ Yes   ✕ No

Exercise 03: ColorFilter

Basic tests

Test the different methods, by taking a picture of your choice and applying the filters.
You should observed the 'to_green' method generate a new array corresponding to the original
image with a green filter, and so on.

        ⊘ Yes   ✕ No

Exercise 04 - K-means Clustering

Implementaion & Basic tests

- First verify the correct implementation of the class (constructor and methods).
  Try launching the following command with the corresponding path on your terminal:
  (block 03.04.00)

- The centroids are randomly initialized.

- Program can handles different set of values (reasonnable ones) for 'ncentroid' and 'max_iter' arguments.

- For a fix value of 'ncentroid', centroids coordinates slightly differ when running the algorithm multiple times.

- Results are coherent and consistent trough multiple program execution.
  Most of the time, centroids converged to similar coordinates and number people associated to each centroids varies little.

⊘ Yes   ✕ No

Ratings

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

✔ Ok

▌ Empty work   ▨ Cheat   ♟ Crash   ♟ Incomplete group   ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Tres bon travail bien explique

Finish evaluation