(https://projects.intra.42.fr/scale_teams/3301137/edit#)

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

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

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT PISCINE PYTHON DJANGO (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/42CURSUS-PISCINE-PYTHON-DJANGO) / DAY 03 (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/42CURSUS-PISCINE-PYTHON-DJANGO-DAY-03)

You should evaluate 1 student in this team

Git repository

```
git@vogsphere-v2.s19.be:vogsphere/intra-uuid-c748724c-9af8-4fd4-b4c2-
```

Introduction

For the smooth running of this evaluation, please respect the following rules:

- Remain polite, kind, respectful and constructive whatever happens during this conversation. It's a matter of confidence between you and the 42 community.

- Highlight the potential problems you ‘ve had with the work you're presented to the person or the group you're grading, and take the time to talk about and discuss those issues.

- Accept the fact that the exam subject or required functions might lead to different interpretations. Listen to your discussion partner's perspective with an open mind (are they right or wrong ?) and grade them as fairly as possible.
42's teaching methods can make sense only if peer-evaluation is taken seriously.

Guidelines

- You must only evaluate what you will find in the student's or group's GiT repository.

- Take the time to check that the GiT repository matches the student or group and the project.

- Double check that no malicious alias was used to mislead you and make you

grade something different from the official repository content.

- If a script supposed to help evaluate the exam is supplied by either side, the other side will have to strictly check it to avoid nasty surprises.

- If the evaluating student has not yet taken this project, they will have to read the exam subject in its entirety before starting the evaluation.

- Use the flags available on this grading system to signal an empty or non-funcional project, a norm flaw, cheating, etc. In that case, evaluation stops and final grade is 0 (or -42 if it's a cheating problem). However, if it's not a cheating problem, you are invited to keep talking about the work that has been done (or not done, as a matter of fact) in order to identify the issues that lead to this stalemate and avoid it next time.

Attachments

- subject.pdf (https://cdn.intra.42.fr/pdf/pdf/24835/en.subject.pdf)

Foreword

This section is dedicated to the evaluation start and the checking of prerequisites. It's not graded, but if something's wrong or a condition is not met, here or anytime during the evaluation, the grade is 0 and a flag can be ticked if necessary.

Observing the instructions

Observing the instructions

- The repo contains the evaluated student's or group's work.
- The evaluated student or group can explain their work anytime during the evaluation.
- General and specific instructions of the day will be observed during the whole evaluation.

- Yes
- No

Python-Django Training D03

- For each exercise, make sure that the file's global scope doesn't include any line of code, except for the module imports. - For each exercise, make sure that no import has been made except the ones excplicitly authorized in the subject. - If one of those rules is not observed, the whole exercise is invalid. - Hard typing the result of an exercise in the code invalidates this exercise. - No code whatsoever should be included in the global scope of the file. This aims to force the use of functions. - You must test the exercises with Python3. If one exercise doesn't work with Python3, it's considered invalid, even if it works with Python2.

Exercice 00 - Antigravity

The exercise was just about using the 'geohash' function found in the 'Antigravity' module. If the student has recoded this function, well... congratulations... I think. But you still have to make sure everything works as it should.

Check that:

- The program takes three arguments: longitude, latitude and datedow (in any order).

- If one argument is missing or if it's not correct, the program must quit properly after displaying a relevent error message.

- The program returns a correct result: open the console under Python3, run 'import Antigravity' and 'Antigravity.geohash(latitude, longitude, datedow)' and check that for identical arguments, you get indentical results.

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

□ Yes □ No

Exercise 01 - Pip

- There is a script which name ends with '.sh' and a Python program that's logically a '.py'.

Run the script:

- The script must display the pip version.

- The script must install the 'path.py' development version from a github address in the folder named 'local_lib' in the repo folder.

- The script must list the install logs in a file with a ".log" extension.

- The script must execute the "my_program.py" program.

- Check that the Python program imports the path.py module from the 'local_lib' folder created by the script.

- The program must create a folder as well as a file inside it, then write something in this filer and display its content.

Relaunch the script:

- The script must redownload the library in development version from its Github repo, crushing the previous version.

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

□ Yes □ No

Exercise 02 - request an API

- There is a request_wikipedia.py file.

- There is a requirement.txt file that allows your to install the

necessary libraries in a VirtualEnv or on the system.

- Run the program without any parameter, then with an invalid one. The program must quit with a relevant error message. No file must be created.

- Run the program with a functional request. A file named after the request must be created, ending with '.wiki' and including the matching article. The latter must not include JSON formatting or Wiki Markup. There might be some markups left such as "Infobox" and the "" tags. Check that the dewiki library has been used requesting the address in the browser. You must clearly see a difference.

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

□ Yes □ No

Exercise 03 - HTML parsing

- There is a roads_to_philosophy.py file.

- There is a requirement.txt file that allows your to install the necessary libraries in a VirtualEnv or on the system.

- Run the program without any parameter, then with an invalid one. The program must quit with a relevant error message. No file must be created.

- Test the program with the example given in the subject. Is the output right?

- Test the program with a request leading to a stalemate. The program must stop with an explicit message.

- Test the program with another functional request and test the same request on Wikipedia in your browser. Click the first link in the introduction paragraph of each article that does not include 'Help:IPA'. You must get the same result as the program.

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

□ Yes □ No

Exercise 04 - Virtualenv

- Ask the evaluated student to launch the script. '(django_venv)' must appear in your prompt. This means a virtualenv was created and activated. Make sure no virtualenv existed beforehands and it's really been created.

Then:

- Enter the 'python --version' command to make sure you're running the Version 3 of Python. (If Python has been aliased, run 'unalias python'

before)

- Enter the 'pip freeze' command. The terminal must only send you the versions of Django and psycopg2.

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

☐ Yes ☐ No

Exercise 05 - Hello World

If Django is not installed on your system, you can use the previous exercise's virtualenv to run this evaluation.

- You must find a folder with a Django project. Once inside, run the server with the 'python manage.py runserver'command. You should not see any Traceback appear. Don't mind the eventual red messages about migrations.

- Go to the following address in your browser: 'localhost:8000/helloworld'. You should see a simple but stylish 'Hello World !'

If one of the elements is missing or the Python script and/or program crashes, this exercise is invalid.

☐ Yes ☐ No

Ratings

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

☐ Ok

☐ Empty work ☐ No author file W Invalid compilation ☐ Norme ☐ Cheat d Crash

l Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation