(https://learner.coursera.help/hc/requests/new)

Mini-project description — “We want... a shrubbery!”

This mini-project is **optional**. Specifically, your grade on this mini-project will not count towards your final grade in this class. While this mini-project may appear silly, its purpose is important: learning how to submit and peer assess your Python mini-projects via *Coursera*. This mini-project is your chance to run through this process and learn how to avoid some of the simple mistakes that have plagued students in previous sessions of this class. We will walk you through each step of the process so that, next week when it is time for Rock-Paper-Scissors-Lizard-Spock, you will understand the process completely.

The logistics of release, completion, submission and assessment of a mini-project

Before proceeding to the details of this mini-project, let's first review the general cycle of release, completion, submission and assessment of a mini-project. Here are the major steps:

1. Each mini-project will be released at 5:01 GMT Saturday morning and the submission phase for the mini-projects begins. For those who have completed the previous mini-project, this will allow you the entire weekend and the following week (plus that Saturday if necessary) to work on the mini-project.
2. The mini-project description and its accompanying video will describe the mini-project's requirements in detail. Each mini-project will include a detailed development process and an accompanying program template (a link to a CodeSkulptor file) that provides a good starting point for the project. Please use it. For help with the mini-project, consult the discussion forums as well as the Code Clinic tips thread which can be easily accessed via the "Coursera" button in the upper right portion of the CodeSkulptor frame.
3. The submission phase for the mini-project ends (and the mini-project is due) at 5:01 GMT on Sunday morning the following weekend. To submit your mini-project, you simply cut and paste the CodeSkulptor URL for your code into the text box below. (If you have trouble generating a CodeSkulptor URL, use the CodeSkulptor save service (http://codeskulptor.appspot.com/save/).) Next, click the Honor Code button and **hit "Submit"**. Since there is no penalty for multiple submissions before the deadline, I strongly suggest that you do not use "Save", but instead always use "Submit". Forgetting to hit "Submit" after pasting the CodeSkulptor URL for your project into the text box is the most common mistake that students make during this entire process. **Don't forget to hit "Submit"!** *Coursera* has a modest grace period for late submissions. Take advantage of it if you are a few minutes late in finishing.
4. After the submission phase ends, the peer assessment phase of the assignment begins. *Coursera* will provide you with the CodeSkulptor URLs for five of your classmate's mini-projects. Each week's mini-project has an associated grading rubric included in the assignment. You will use the pull down tabs at the bottom of the page to assign values for each rubric item to your peer's projects. As a matter of courtesy, please include a written comment if you decide not to give full score on an item. Remember that losing points with no feedback is very frustrating, so spend a little time, and explain what the problem was.
5. *Coursera* will then ask you to self-assess your own mini-project. The idea behind self-assessment is that you will have a better idea of how your own mini-project measures up once you have seen and graded some of your peer's mini-projects. Please remember to hit the "Submit" button after you have

completed your self-assessment. *Coursera* assesses a 20% penalty to the mini-project of a student who fails to submit the required five peer assessments and one self-assessment. (We know this is a harsh penalty, but, as instructors, we don't have the capability to modify this penalty structure.)

6. The submission phase ends and the results phase (where you see your score) begins at 17:01 GMT on the following Wednesday. Your score for the assignment is the sum of the median of your peer's scores on each item of the grading rubric.

The origins of Python

Python (http://en.wikipedia.org/wiki/Python_(programming_language)) is a computer programming language that was conceived in the late 1980s by Guido van Rossum. Perhaps the most common beginner Python question is “Why would anyone name a computer language after a snake?” In fact, Python's name is derived from the television series Monty Python's Flying Circus (http://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). A common theme in the Python community is to make inside references to the television series when working with Python. Our initial mini-project honors this tradition.

Mini-project development process

Your task is simple: modify this program template (http://www.codeskulptor.org/#examples-shrubbery_template.py) to print

```
We want... a shrubbery!
```

in the CodeSkulptor console. Your program should consist of a single line of Python. Before submitting your program, we suggest that you review the grading rubric given below.

Grading rubric — 4 pts total

Your peers will assess your mini-project according to the following rubric. Small deviations from the textual output described above, like an extra period (`.`) or space are fine. Whether moderate deviations, such as just printing `shrubbery`, satisfy an item of the grading rubric is at your peers' discretion during their assessment. You should avoid large deviations such as printing a different Monty Python quote. Here is a breakdown of the scoring:

- 1 pt — A valid CodeSkulptor URL was submitted. No credit if the code is just pasted into the text box.
- 1 pt — Running the program does not throw an error.
- 1 pt — The program prints a message in the console.
- 1 pt — The program prints the desired message:

  ```
  We want... a shrubbery!
  ```