

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. This mini-project description and its accompanying video describes the mini-project's requirements in detail. Each mini-project includes 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.
2. For subsequent mini-projects, the material for each mini-project also includes a page titled "Code Clinic Tips" page that includes common errors and hints. These items are based on observing 10+ sessions of students work through the associated mini-project.
3. Once you have completed your mini-project, you can submit your mini-project by copying and pasting the CodeSkulptor URL for your code into the text box provided on the peer assessment page. (If you have trouble generating a CodeSkulptor URL, use the [CodeSkulptor save service](#).)
4. Once you have submitted the URL for your solution, Coursera will prompt you to assess the submissions of five of your peers. In particular, *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. Once you have assessed the work for five of your peers, your score for the assignment will be available. The score is the sum of the median of your peer's scores on each item of the grading rubric. Note that by taking the median of your peers' scores, any errors in grading by one of your peers will not affect your final grade.

The origins of Python

[Python](#) 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](#). 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](#) 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!"**.