

Advanced Exercises Set 9

Python

EECS 201 Winter 2020 COVID-19 Edition

Submission Instructions

To receive credit for this assignment you will need to email the staff at eeecs201-staff@eeecs.umich.edu with tar archive(s) of your submissions attached as well as any additional information requested in the body of your email. You may choose the exercises that you wish to do from this set. Each exercise is denoted by its point count. **Extra credit is given for early turn-ins of advanced exercises. These details can be found on the website under the advanced homework grading policy. For these email-based submissions, your latest submission of the tar archive/questions will determine your extra credit.**

1 How are you holding up? (5)

In your submission email, give me a paragraph or two telling me how you're holding up, how your classes have been handled amidst the COVID-19 outbreak (I'm curious as how some EECS classes are handling all this madness, as well as how non-recorded lectures work out for people who are in far off time-zones), and your thoughts on how the school is handling it (e.g. lack of refunds for tuition? Being kicked out of campus housing?).

- Include this report in the body of your email.

2 Interactive debugging with Python (5)

Python actually has an interactive debugger **included in it called `pdb`**! There's multiple ways to access it, such as inserting breakpoints directly into the code via `import pdb; pdb.set_trace()` or `breakpoint()` (in version 3.7 and higher), or running your script with it explicitly from the invocation: `$ python3 -m pdb myscript.py`.

For this exercise, try playing around with the debugger with perhaps the scripts you wrote for the homework or try debugging some other script that you have written. In your email, discuss how you started the debugger and how it felt compared to other debugging methods you have tried (e.g. GDB).

- Include this report in the body of your email.

3 Who needs MATLAB? (5)

I mentioned in lecture that you could probably replace MATLAB with Python + NumPy + SciPy + Matplotlib for general purpose use (unfortunately, MATLAB does have some really nifty tools in more focused areas that can be hard to replace).

For this exercise, we're going to keep it simple since the SciPy stuff tends to be more domain specific: as much as I want to throw FFTs at you, I realize that not everyone has taken EECS 316. The only package dependencies you'll need are NumPy (for data types) and Matplotlib (for plotting). If you don't have those, try installing the `pip` Python package manager and installing those two packages (or do whatever setup you need to install Python packages for your particular system).

Write a script that plots $y = x^3 - 7x^2 + 2x + 2$ for $-4 \leq x \leq 4$ with the title "Who needs MATLAB?" and saves the figure to a PNG file called `output.png`. The number of samples you use is up to you, but the figure should not look jagged from a low sample count.

- Include your script and the produced PNG in the tarball.
- Mention how much previous experience you have had with MATLAB, NumPy, and Matplotlib in your email.

Helpful hints:

- For those unfamiliar with the general MATLAB workflow for generating plots, you'll need to generate samples for your X coordinates and samples for your Y coordinates: these will be matched up by index. For example, your X coordinates could be [-4, -2, 0, 2, 4] and your Y coordinates could be [16, 4, 0, 4, 16] when plotting $y = x^2$, albeit with only 5 samples. These samples would then be passed to a plotting function.
- NumPy comes with its own version of MATLAB's `linspace()`: `numpy.linspace()`. This can come in handy for generating your X coordinate samples.
- If you do this right, given that you have your X coordinate samples, you only need one line to calculate your Y coordinate samples. `numpy.ndarray`s are your friends!
- Matplotlib's MATLAB-like plotting function is `pyplot.plot`. `pyplot` also has a mechanism to save figures.

4 Go wild! (10 with potential for bonus)

The Python standard library is pretty darn big. I have no idea what sorts of problems that you all face in each of your classes or in your life, but chances are there's some problem that you can solve with a Python script(s) and using the Python standard library. Or perhaps the standard library is missing some feature (*cough* YAML parsing *cough*) and you need to install some package.

The point of this particular exercise is to go wild and stretch out your wings with real problems you face: this is probably the best way to learn and get a feel for Python.

Minimum requirements are to use at least one of the more "advanced" modules/packages and have either some sort of file I/O or networking capability. Modules that provide fairly basic/fundamental things like `os`, `math`, `string` don't count: feel free to ask if your choice in "advanced" module is appropriate. Some examples of modules/packages that I find "advanced" enough from the standard library: `csv`, `argparse`, `curses`, `email`, `json`, `html`, `xml` (and friends), `http.client`, `sqlite3`, `tkinter`.

- Include your script(s) and any additional data files if needed into the tarball.
- Mention what packages that your scripts depend on in the body of the email.
- Tell me what your script is about and what it's for in the body of the email.