CS-101 CS for All | Project 3: Visualizing Natural Language

Due date: Monday, October 10 at 11:00pm

You can submit your code to Gradescope more than once, and we will only see the latest version. However, if you submit after the due date, then your submission will appear to us to be late.

Please don't use the internet for anything more than looking up syntax and simple commands from reference websites. You should definitely not search online for anything resembling strategy. Keep such questions in-house by asking peers, teaching assistants, and professors.

---

1 Instructions

Use the `nltk` and `matplotlib` libraries to accomplish something that makes use of both libraries.

The tools that you should use are:

1. **The nltk library**: Use at least two of the built-in methods that this library provides.
2. **The matplotlib library**: Use at least one of the built-in methods that this library provides.
3. **Loops**: The natural language problem that you come up with should require at least one loop that helps you avoid a lengthy copy-and-paste chunk of code.
4. **Conditionals**: Your problem should also require at least one conditional statement that helps you make one or more important choices.
5. **Functions**: Write at least one helpful function that does a job or returns one or more values that simplifies your overall task in a substantial way.

You are of course welcome to ask any of us, if need be, what we mean by words like "substantial" and "important". Feel free to use other tools, such as randomness, that you learned in class or learn about through some other source.

2 Collaboration and Citations

Please see the syllabus for details. You are welcome to work with another person on this project. Each of you should submit your code (it should be identical) individually, and make your own video.

3 Assessment of Code

We will grade each of the following categories on a scale from 0.0 to 4.0 (by 0.5s) that imitates how grade point averages are determined. We will think of a 3.0 as being very good with some flaws, while a 3.5 or 4.0 each denotes excellence. Grades lower than 3.0 simply indicate greater flaws.

- **(4.0 points) Focus**: *Your program is designed to do the overall job that we asked you to complete.* Some projects will have more complex instructions than others.
- **(4.0 points) Accuracy**: *Your code non-trivially satisfies our instructions.* Part of learning the content of this course is figuring out why we teach you the tools we do. You should use the tools appropriately for their intended purpose, which means that you should carefully consider what tasks you set for yourself.
- **(4.0 points) Ambition**: *You stretched yourself in what you chose to create, in its complexity and its details.* It is next to impossible to associate a time limit with ambition, as you have varying degrees of experience, so just challenge yourself and do your best. You should aim to accomplish at least one thing that you didn't directly learn in class or at lab.

4 Assessment of Video

Once your code is finished, you should use Zoom to create a video in which you share your screen and demo your program for us. If you worked with a partner, then each of you should make your own video. This is the rubric that we will use to assess your video:

- **(0.5 points)** *You explained (if need be) how to interact with your program.* You may not need to explain anything, of course, if all we would have to do is hit the Run button.
- **(0.5 points)** *Your video was an appropriate length.* Aim for at most five minutes. If you go past five minutes, your content had better be worth it!

- **(0.5 points)** *Your tour of your output was clear and helpful.* In some cases, the output will basically speak for itself, and if so, you do not need to elaborate much. But in other cases, your program will do something complex, so you should make sure that your narration tells us clearly what it can do.
- **(0.5 points)** *You demonstrated how your program met the demands of our instructions.* You can do this by pointing out certain features of your output and/or by showing us your code. If you do show us your code, don't scroll around like a squirrel. Stay calm and give us a chance to read it.
- **(2.0 points)** *You sufficiently explained how the most complex part of your code worked.* Pretend that you are explaining your code to a friend who knows Python but has never seen your code, and tell them the overall purpose of the code as well as how each line of the code itself contributes toward that purpose.

Put your video on Google Drive and, when you submit your code to Gradescope, you will have a chance to share that link with us: click "Share", click "Get link", click the drop-down that says "Restricted", and from this list choose "Hamilton College". Then copy the link and paste it into your Gradescope submission.

Anything you'd like to add to the above assessment is most welcome, including explanations of how you went above and beyond, or narrated tours of parts of the code that you're proud of.