

# DATA203 Foundational Python (Prof. Maull) / Spring 2026 / HW0

Points Possible	Due Date	Time Commitment (estimated)
10	Friday, February 6 @ midnight	<i>up to 8 hours</i>

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by University or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

## OBJECTIVES

- Familiarize yourself with the JupyterLab environment, Markdown and Python
- Familiarize yourself with Github and basic git
- Explore JupyterHub Linux terminal console integrating what you learned in the prior parts of this homework
- Learn more about and use string manipulation

## WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw0`. Put all of your files in that directory. Then zip or tar that directory, rename it with your name as the first part of the filename (e.g. `maull_hw0_files.zip`, `maull_hw0_files.tar.gz`), then download it to your local machine, then upload the `.zip` to Canvas.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

If you choose not to use the provided notebook, you will still need to turn in a `.ipynb` Jupyter Notebook and corresponding files according to the instructions in this homework.

## ASSIGNMENT TASKS

### (0%) Familiarize yourself with the JupyterLab environment, Markdown and Python

As stated in the course announcement [Jupyter \(https://jupyter.org\)](https://jupyter.org) is the core platform we will be using in this course and is a popular platform for data scientists around the world. We have a JupyterLab setup for this course so that we can operate in a cloud-hosted environment, free from some of the resource constraints of running Jupyter on your local machine (though you are free to set it up on your own and seek my advice if you desire).

You have been given the information about the Jupyter environment we have setup for our course, and the underlying Python environment will be using is the [Anaconda \(https://anaconda.com\)](https://anaconda.com) distribution. It is not necessary for this assignment, but you are free to look at the multitude of packages installed with Anaconda, though we will not use the majority of them explicitly.

As you will soon find out, Notebooks are an incredibly effective way to mix code with narrative and you can create cells that are entirely code or entirely Markdown. Markdown (MD or `md`) is a highly readable text format that allows for easy documentation of text files, while allowing for HTML-based rendering of the text in a way that is style-independent.

We will be using Markdown frequently in this course, and you will learn that there are many different “flavors” of Markdown. We will only be using the basic flavor, but you will benefit from exploring the “Github flavored” Markdown, though you will not be responsible for using it in this course – only the “basic” flavor. Please refer to the original course announcement about Markdown.

### § Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

Play with and become familiar with the basic functions of the Lab environment given to you online in the course.

### \$ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

Please *create a markdown document* and read the documentation for basic Markdown [here](#). Learn to use all of the following:

- headings (one level is fine),
- bullets,
- bold and italics

Again, the content of your document can be whatever you like, just learn some of the basic functionality of Markdown.

### (0%) Familiarize yourself with Github and basic git

Github (<https://github.com>) is the *de facto* platform for open source software in the world based on the very popular [git](https://git-scm.org) (<https://git-scm.org>) version control system. Git has a sophisticated set of tools for version control based on the concept of local repositories for fast commits and remote repositories only when collaboration and remote synchronization is necessary. Github enhances git by providing tools and online hosting of public and private repositories to encourage and promote sharing and collaboration. Github hosts some of the world's most widely used open source software.

**If you are already familiar with git and Github, then this part will be very easy!**

### \$ Task: Create a public Github repo named "hu-f25-data203" and place a README.md file in it.

Create your first file called README.md at the top level of the repository.

Please put your Github username in the file. Aside from that you can put whatever text you like in the file (If you like, use something like [lorem ipsum](#) to generate random sentences to place in the file.). Please include the link to **your** Github repository that now includes the minimal README.md. You don't have to have anything elaborate in that file or the repo.

### \$ Task: Fork the course repository.

Learn to use Github workflows and fork the class repo:

- [https://github.com/kmhuads/s26\\_data203](https://github.com/kmhuads/s26_data203)

### (0%) Explore JupyterHub Linux terminal console integrating what you learned in the prior parts of this homework

The Linux console in JupyterLab is a great way to perform command-line tasks and is an essential tool for basic scripting that is part of a data scientist's toolkit. Open a terminal console in the lab environment and familiarize yourself with your files and basic commands.

### \$ Task: Understand basic Linux file operations

Basic file operations go a long way to understand the way Linux works. In this part, you will understand folders, files and making revisions to a file. These files will be visible within Jupyter, which makes moving from one platform to another seamless. We will create a folder, file, make edits.

- open a Jupyter console
- create a file called README.md
- type `mkdir your_folder_name` to create a folder in filesystem *in the current folder where you are*
- use `cd your_folder_name` to "change directory" and move into the folder you just created
- use `pwd` to "print working directory" to verify you are in the folder you created
- create a file by type `touch README.md` the touch command creates a file if it does not already exist, otherwise it will change the timestamp of that file when it is "touched"
- type `echo "Hello this is test text." > README.md`. This will take the words you typed and "append them" into the file README.md
- to see the contents of your file typing `cat README.md` or `more README.md` or `less README.md`

### \$ Task: Learn to quickly obtain remote files in Linux

The commands `wget` and `curl` are useful for grabbing data and files from remote resources off the web. Using these tools from the command line streamline your workflows and are often faster than writing a program to do the same. These tools will also expand and strengthen your data science skills, adding a few more tools to your toolkit is rarely a bad idea.

1. Read the documentation on each of these commands by typing `man wget` or `man curl` in the terminal.

- `man` stands for *manual* and nearly all versions of Linux have such documentation pages for the majority of commands. If it fails, try the command with the `-h` or `--help` flag, such as `wget --help`
- 2. Make sure your output goes to a file and study the documentation to select the proper flags to do so.
- 3. You can obtain nearly any file anywhere on the Internet with these commands. For example, the Library of Congress interview with jazz great Herbie Hancock from the mid-1980s: <https://www.loc.gov/item/jsmith000096/>
  - click on this interview link
  - choose the dropdown for the **mp3**
  - when the page opens up, there will be a player that starts the interview, copy the URL
  - go to your Jupyter terminal and run the command `wget <the_url_you_just_copied>`, where you will paste the URL you just copied in the `<the_url_you_just_copied>`
- 4. Either on the Library of Congress site or somewhere else, play further with `wget` and `curl` to download some other files you might have of interest.

### (100%) Learn more about and use string manipulation

We learned in lecture that strings are *sequences* in Python.

We also learned that the sequence types have a number of basic operations like concatenation, length, etc.

Strings have a lot of other operations on them that make them exceedingly useful for text processing. In fact, Python is exceptionally good at processing text, as you will see.

First things first, please go to the documentation on Python strings, also known as `str` or “Text Sequence Type”:

- <https://docs.python.org/3/library/stdtypes.html#textseq>

Study it and especially the String methods (see more here: <https://docs.python.org/3/library/stdtypes.html#string-methods>).

You will be using this short **AI-generated writeup (qwen3-instruct/4B; note → there are many inaccuracies in the generated text)** about the impeccable jazz singer (<http://www.ellafitzgerald.com/>):

Ella Fitzgerald is an American jazz singer widely regarded as one of the greatest and most influential vocalists in the history of jazz. Born on April 25, 1916, in Newport News, Virginia, she developed a distinctive voice and style that emphasized precision, phrasing, and emotional expression.

Fitzgerald was known for her impeccable timing, dynamic range, and ability to improvise with grace and subtlety. She was a master of scat singing and had a profound influence on the development of jazz vocal styles. Her work with the legendary Louis Armstrong and her collaborations with musicians like Miles Davis, Duke Ellington, and Ray Charles helped shape the sound of jazz music.

She was also a key figure in the “crossover” movement, bringing jazz to broader audiences through pop and film music. Fitzgerald recorded over 1,000 songs and was inducted into the Grammy Hall of Fame and the National Recording Registry.

Ella Fitzgerald passed away on June 15, 1996, in New York City, leaving behind a legacy that continues to inspire generations of musicians and fans alike.

### \$ Task: Basic String functions.

Use the passage provided and answer the questions. **Provide your answer in a Jupyter Notebook.**

1. Assign a variable called `passage` with the string provided into a multi-line string using `"""` which we talked about in lecture.  
Make sure you preserve the spaces between paragraphs, which are `“\n\n”` (two newlines).
2. Use `str.splitlines()` to determine how many paragraphs are in the string.  
Recall, `splitlines()` returns a *sequence* (i.e. a list sequence), so counting the paragraphs is easier with `len()`.
3. Use `lower()` to produce the lowercase version of the whole string.

### \$ Task: Advanced String functions.

Now that you have `passage` in a variable, there are a few more String operations we want to try to familiarize ourselves with.

1. Use `replace()` to replace all instances of the word " her " with `###`.  
You would be advised to use `str.lower()` first so that all of your words are *normalized*. **Note also the spaces around the word.** So for example, the sentence "The place where her glove was lost." would be "The place where `###` hat was lost."
2. How many " and " words are in the passage?
3. How many times was the word " jazz " used in the passage? Use the lowercase-normalized count.

**\$ Task: Sequence iteration.**

Now we will put looping into our work and ask more complex questions of the text.

To produce a list sequence of all the words in passage then you will learn that `str.split()` will be very valuable.

1. Study what `passage.split()` does and write a sentence fragment explaining what it does.
2. Get the first paragraph (index 0), and return the number of words in the paragraph. **NOTE:** you can include punctuation as part of the word. You will make use of your solution in #2 and `split()`.
3. Use a `for` loop with `split()` to print all words that end in "ing". Provide the count of such words. **Note:** you do not have to remove punctuation, so if you end up with an item like "walking," you can ignore it in your count.
4. How many words end in "er"? Provide the list of those words. You will need to use a `for` loop.
5. How many times does the word "Ella" appear in the passage?

**\$ Task: Find AI errors.**

The text was AI-generated and so has errors in it. For 1 point, list at least one incorrect statement in the passage. You might use: <http://www.ellafitzgerald.com/biography> Allmusic.com's [biography page](#) Do this in a Markdown cell in your notebook.