

## Setup Jupyter via Longleaf On Demand

**Super Important:** Make sure you complete the [First-time setup](#) section at the end of this document or you will not be able to start the coursework for CHEM 467.

Once your account is approved you will be able to access Longleaf. There are many ways to access Longleaf but for this course we will be using Longleaf On-Demand.

1. Go to <https://ondemand.rc.unc.edu/pun/sys/dashboard>.
2. Sign in using your onyen.
3. Click on **Jupyter Notebook**.
4. Request resources and click **Launch**.

Home / My Interactive Sessions / Jupyter Notebook

Interactive Apps

Desktops

Longleaf Desktop

Longleaf Desktop (GPU-FULL)

Longleaf Desktop (GPU)

MD Desktop

GUIs

3D Slicer

COMSOL

FSL

Firefox

Freeview

GView

GaussView

ITK-SNAP

MATLAB

Mathematica

Pymol

RStudio Desktop

SAS

Spyder

Stata

Servers

Jupyter Notebook

Jupyter Notebook (GPU)

Jupyter Notebook (GPU-FULL)

RStudio Server

Jupyter Notebook

This app will launch a Jupyter Notebook server on Longleaf.

Number of hours

8

Number of CPUs

1

Unless you are certain you are running parallel code you should specify **one** cpu.

☐ Request GPU

Check if you would like to request a GPU for this session.

Additional Job Submission Arguments

☐ legacy jupyter notebook

Check if you would like to use the legacy jupyter notebook user interface, otherwise you get the jupyterlab interface.

☐ I would like to receive an email when the session starts

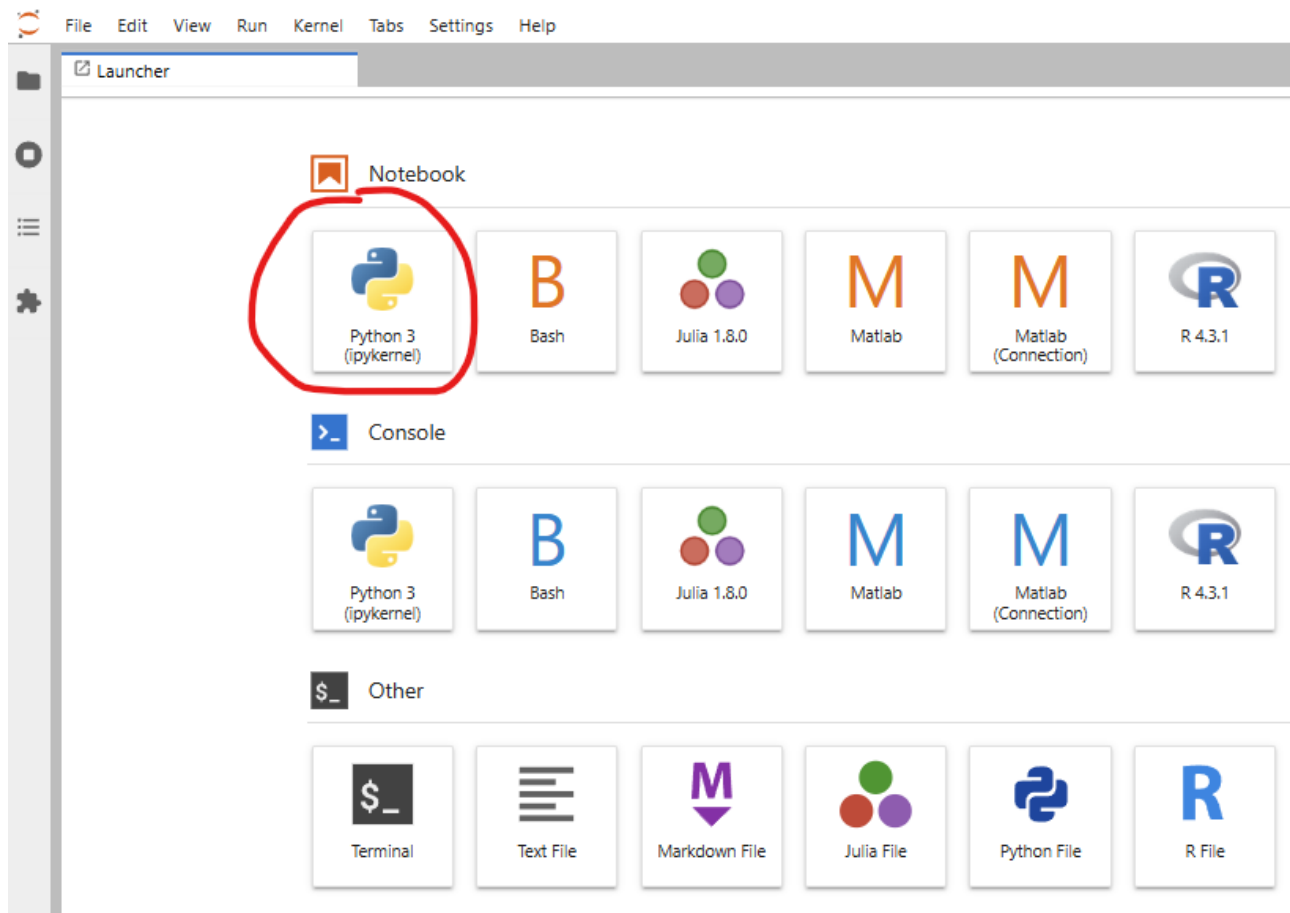
Jupyter startup directory

This field allows you to specify the absolute path to the Longleaf directory where you would like the Jupyter session to start.

Launch

\* The Jupyter Notebook session data for this session can be accessed under the [data root directory](#).

- 5. Wait for computing resources to be allocated (~1 minute).
- 6. Click "Connect to Jupyter".
- 7. In the **Launcher** and under the section labeled **Notebook**, click "Python3 (ipykernel)" to open a notebook running Python.



## Setting up your Jupyter notebook

Jupyter notebooks are made up of two main types of cells: **Markdown cells** and **Code cells**. You run a Jupyter notebook one cell at a time by click inside the cell and press **shift+enter**.

### Markdown cells

Jupyter notebooks allow us to use a language called Markdown to type notes and descriptions in markdown cells. It's a good practice to have your first cell be markdown to explain the purpose of the notebook.

### Basic markdown syntax

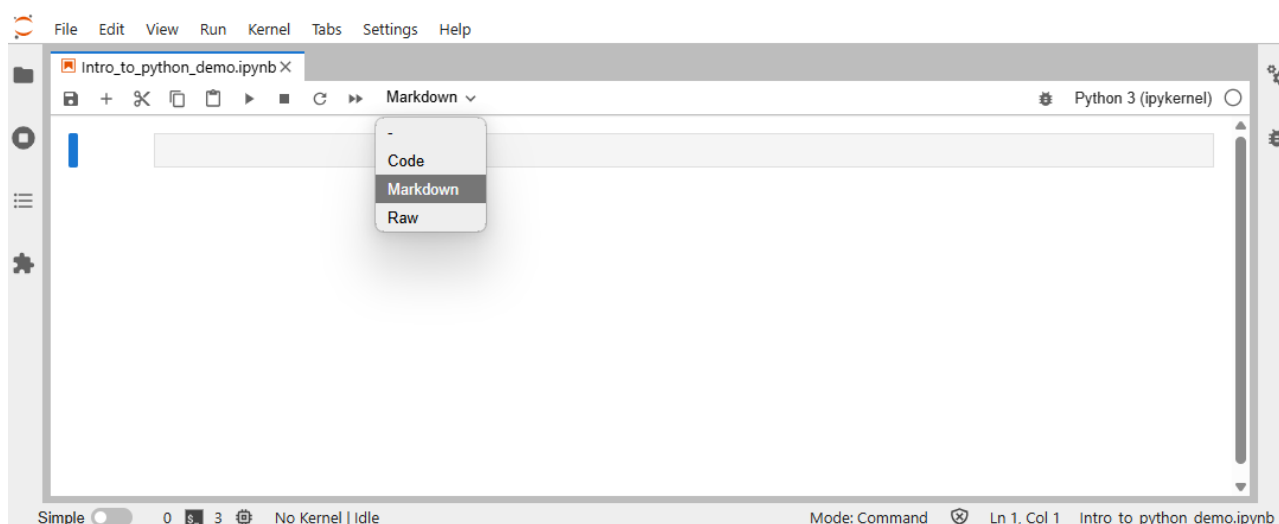
In the Markdown language, we create headers and lists using the following syntax.

Element	Basic Markdown Syntax
Heading	# H1
	## H2
	### H3

Element	Basic Markdown Syntax
Ordered List	1. First item
	2. Second item
	3. Third item
Unordered List	- First item
	- Second item
	- Third item

Let's practice!

1. Open a new Jupyter notebook using the [instructions above](#).
2. Click the first cell.
3. Select **Markdown** from the cell type dropdown menu in the notebook ribbon (shown below).



4. Paste the following into the markdown cell.

```
# Python for chemists
```

```
UNC CHEM 467
```

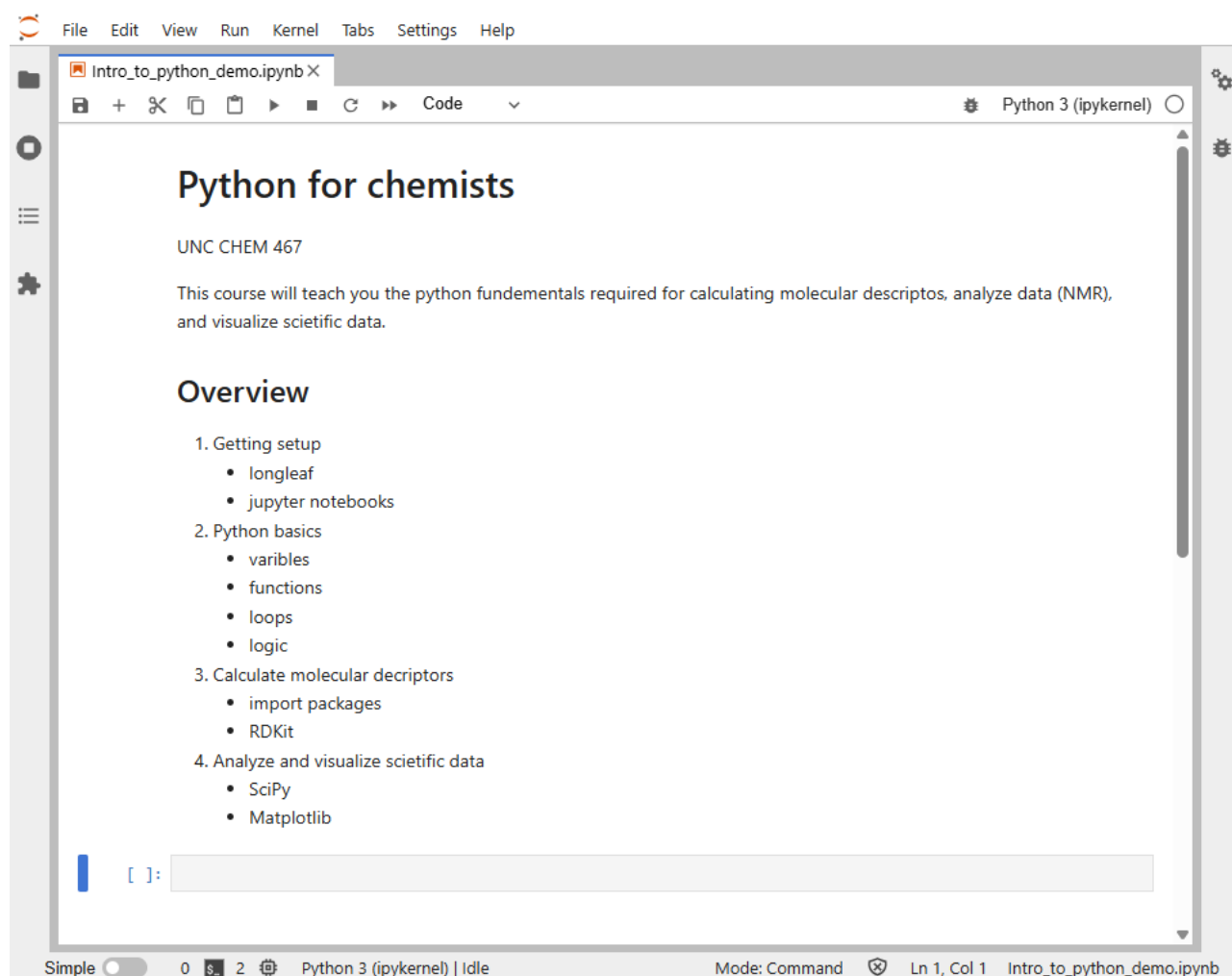
```
This course will teach you the python fundamentals required for  
calculating molecular descriptos, analyze data (NMR), and visualize  
scietific data.
```

```
## Overview
```

1. Getting setup
  - longleaf
  - jupyter notebooks
2. Python basics
  - variables
  - functions
  - loops

- logic
- 4. Calculate molecular descriptors
  - import packages
  - RDKit
- 5. Analyze and visualize scientific data
  - SciPy
  - Matplotlib

5. Press **shift+enter** to evaluate the markdown cell. Now your notebook should look like the following:



## Code cells

Jupyter notebooks allow us to run python code interactively and display the outputs all in one file. These features make Jupyter notebooks good for prototyping new code and for keeping a record of how you analyzed your data.

Let's practice!

1. Paste the following interactive python code into the empty code cell below our Markdown cell.

```
# Example of interactive python code in Jupyter

# Import packages
import numpy as np
import matplotlib.pyplot as plt
from ipywidgets import interact

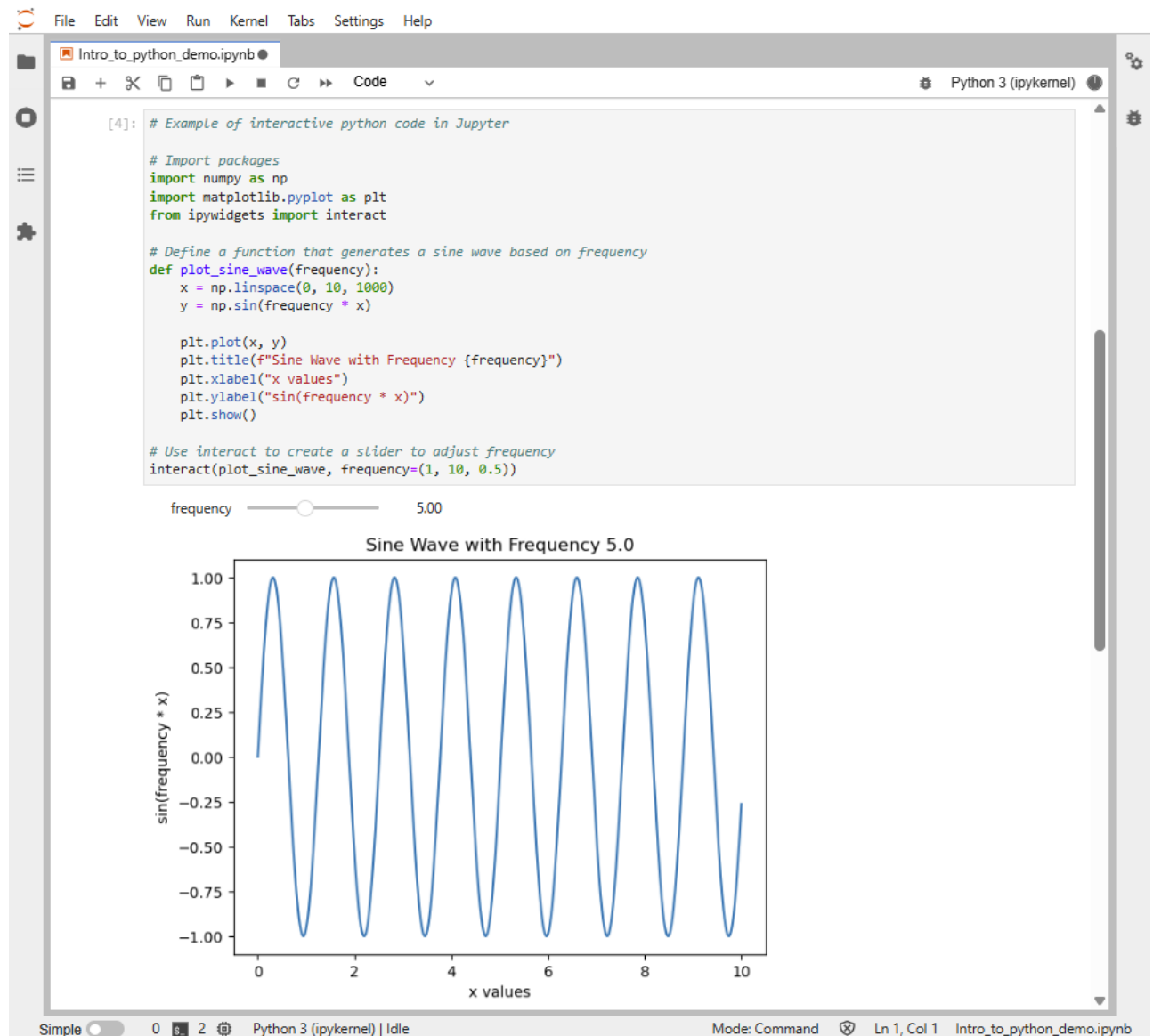
# Define a function that generates a sine wave based on frequency
def plot_sine_wave(frequency):
    x = np.linspace(0, 10, 1000)
    y = np.sin(frequency * x)

    plt.plot(x, y)
    plt.title(f"Sine Wave with Frequency {frequency}")
    plt.xlabel("x values")
    plt.ylabel("sin(frequency * x)")
    plt.show()

# Use interact to create a slider to adjust frequency
interact(plot_sine_wave, frequency=(1, 10, 0.5))
```

**Note:** in Python (and therefore in Jupyter code cells) the # symbol is used to denote comments that will not be interpreted as code.

2. Press **shift+enter** to evaluate the code cell. Now your notebook should look like the following:



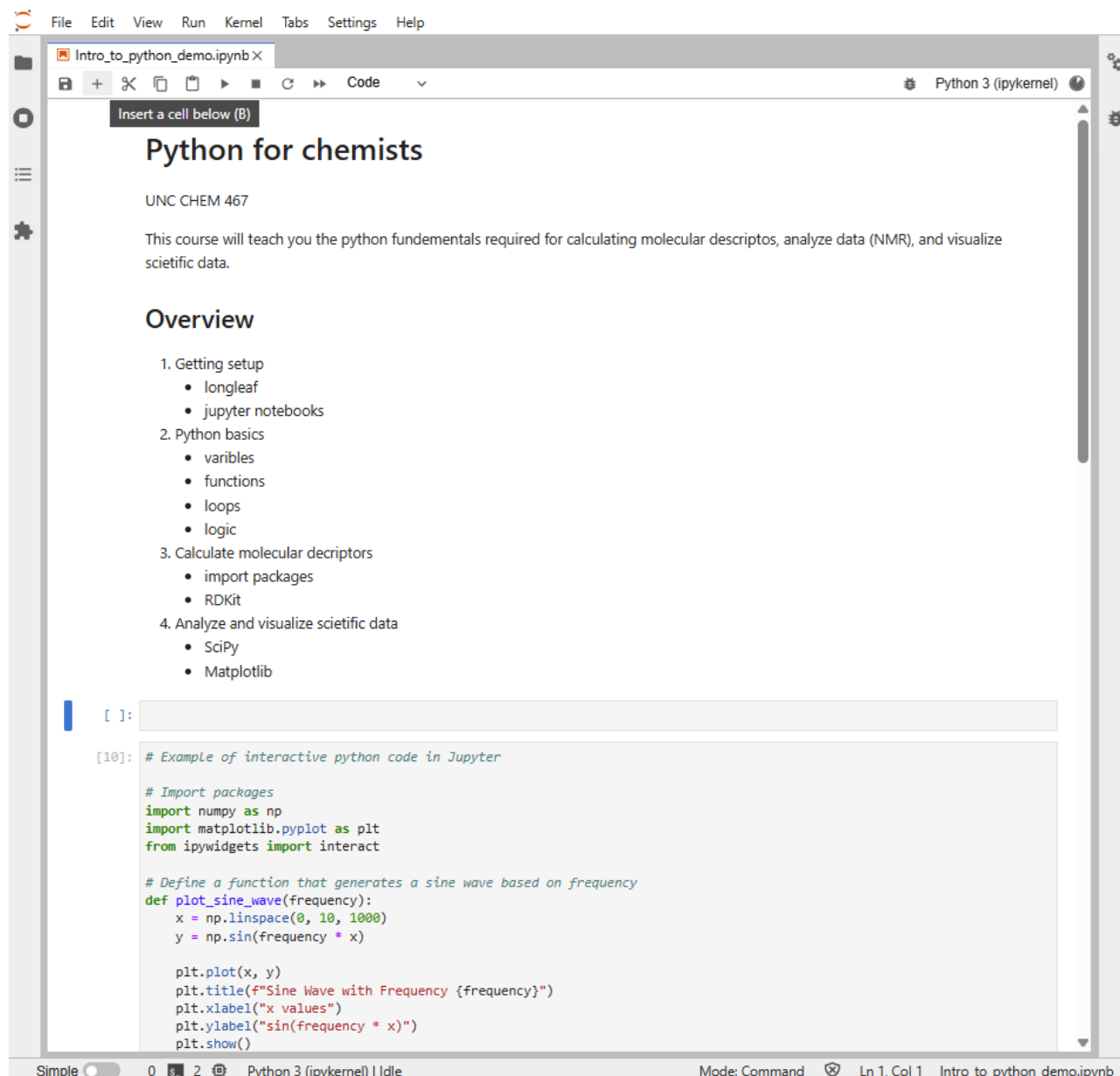
**Note:** move to the frequency slider to interact with the output.

## How to insert and delete cells

1. Insert a new cell below our markdown cell.

- click the markdown cell
- click the **+** button in the ribbon to insert a cell below

Now your notebook should look like the following:



2. Delete the empty cell.

- right-click the cell
- click **Delete Cells**

## First-Time Setup (Super important)

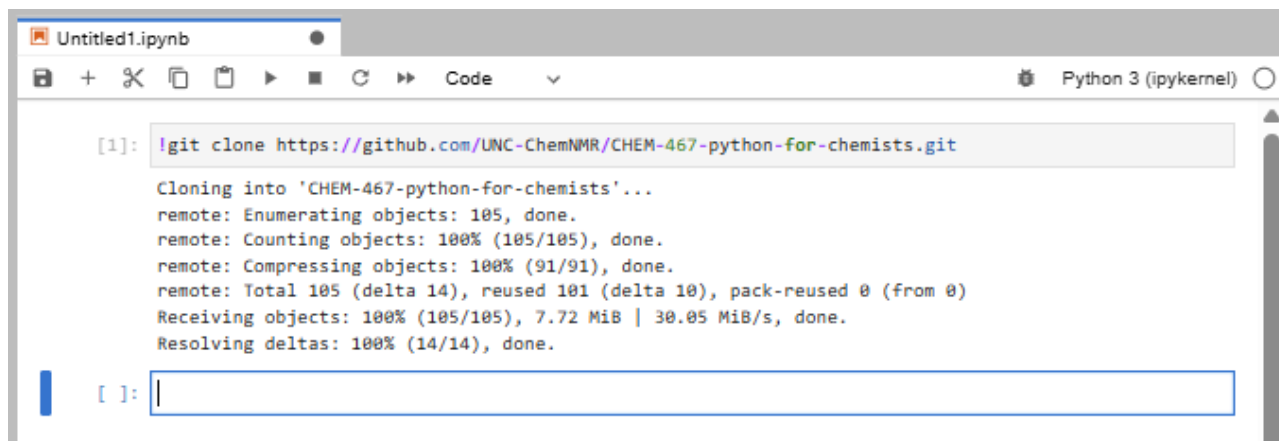
Clone the class repository

This will copy all the course material into your Longleaf storage.

1. Create a new code cell.
2. Paste this code into the code cell:

```
!git clone https://github.com/UNC-ChemNMR/CHEM-467-python-for-chemists.git
```

3. Run the code cell by pressing **shift+enter**
4. You should see an output that looks similar to this:

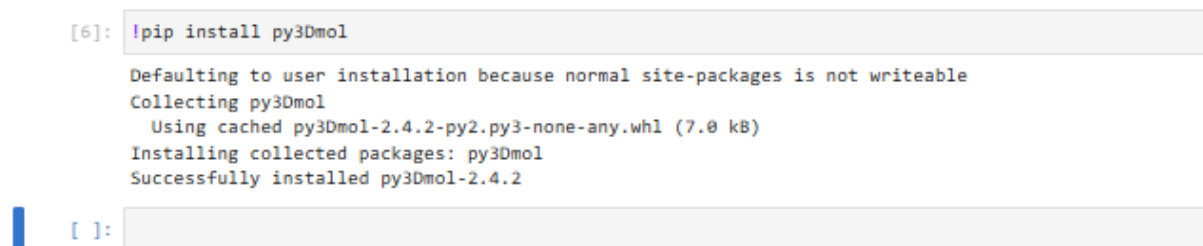


The screenshot shows a Jupyter Notebook window titled 'Untitled1.ipynb'. The first code cell contains the command `!git clone https://github.com/UNC-ChemNMR/CHEM-467-python-for-chemists.git`. The output shows the cloning process: 'Cloning into 'CHEM-467-python-for-chemists'...', 'remote: Enumerating objects: 105, done.', 'remote: Counting objects: 100% (105/105), done.', 'remote: Compressing objects: 100% (91/91), done.', 'remote: Total 105 (delta 14), reused 101 (delta 10), pack-reused 0 (from 0)', 'Receiving objects: 100% (105/105), 7.72 MiB | 30.05 MiB/s, done.', and 'Resolving deltas: 100% (14/14), done.'. A second, empty code cell is visible below.

## Install py3Dmol

1. Create a new code cell.
2. Paste this code into the code cell:  

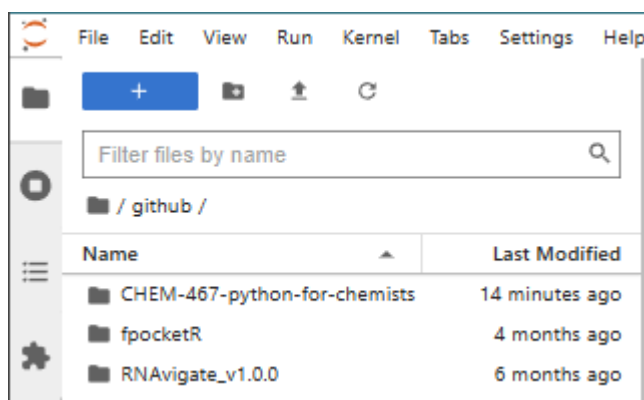
```
!pip install py3Dmol
```
3. Run the code cell by pressing **shift+enter**
4. You should see an output that looks similar to this:



The screenshot shows the output of the `!pip install py3Dmol` command. The output text is: 'Defaulting to user installation because normal site-packages is not writeable', 'Collecting py3Dmol', 'Using cached py3Dmol-2.4.2-py2.py3-none-any.whl (7.0 kB)', 'Installing collected packages: py3Dmol', and 'Successfully installed py3Dmol-2.4.2'. Below the output is an empty code cell.

## Navigating the course material in Jupyter

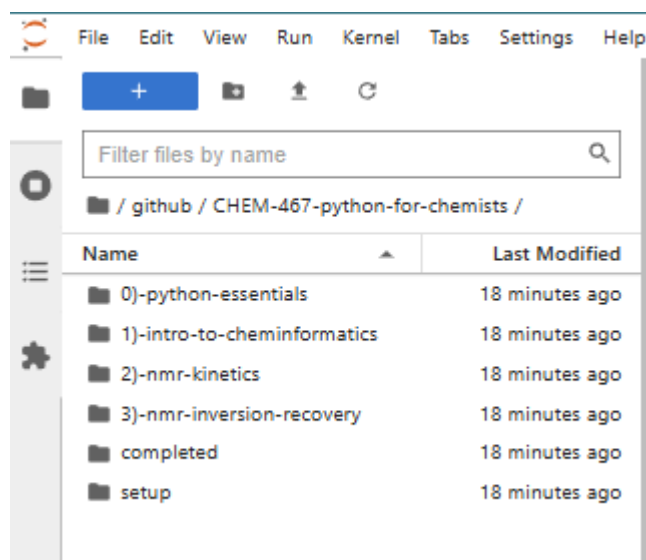
1. You should now see a folder (or directory) containing all the course material:



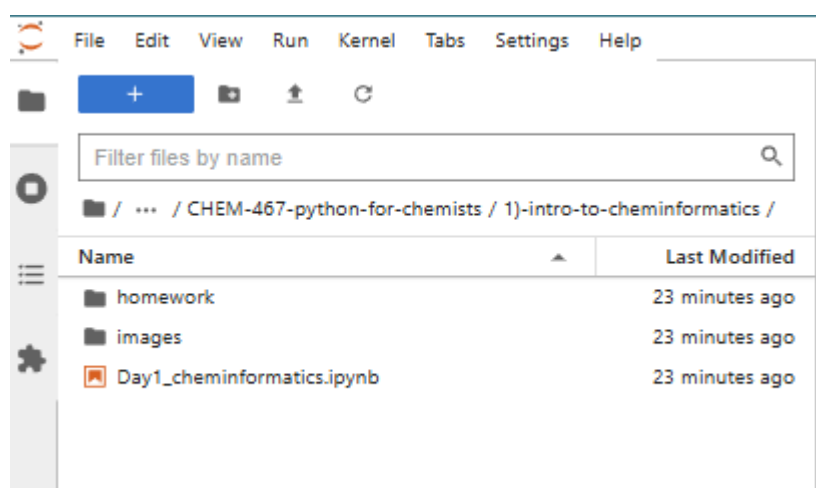
You're are done setting up your jupyter environment for CHEM 467.

2. You can double-click on the course folder to open it.





3. In-class notebooks and homework templates are located inside the numerated sub-directories.



You are not done setting up for CHEM 467!

## Additional resources

Hopefully you have a good idea of how to run a Jupyter notebook in Longleaf now! If you have more questions take a look at the following resources.

- [Longleaf OnDemand](#)
- [Jupyter notebook user interface](#)