



Chem 361: Machine Learning in Chemistry

Xuhui Huang
Spring 2025

Tutorial:

set up your Python environment for data analysis and ML

Introduction to Python

Python is one of the major programming languages with wide applications today



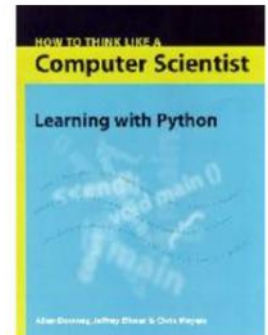
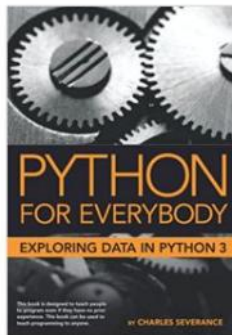
- Simplicity and versatility
- Readability
- Extensive ecosystem



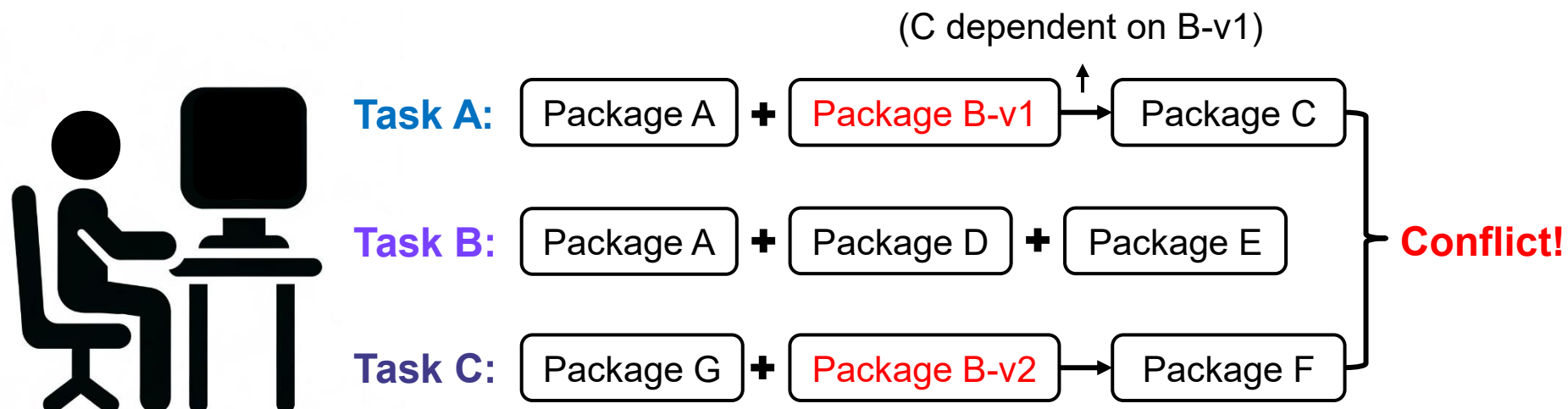
Data Analysis Web Development Machine Learning

Free books on using Python

<https://bcrf.biochem.wisc.edu/2022/08/23/free-data-science-books/>



Package management in Python



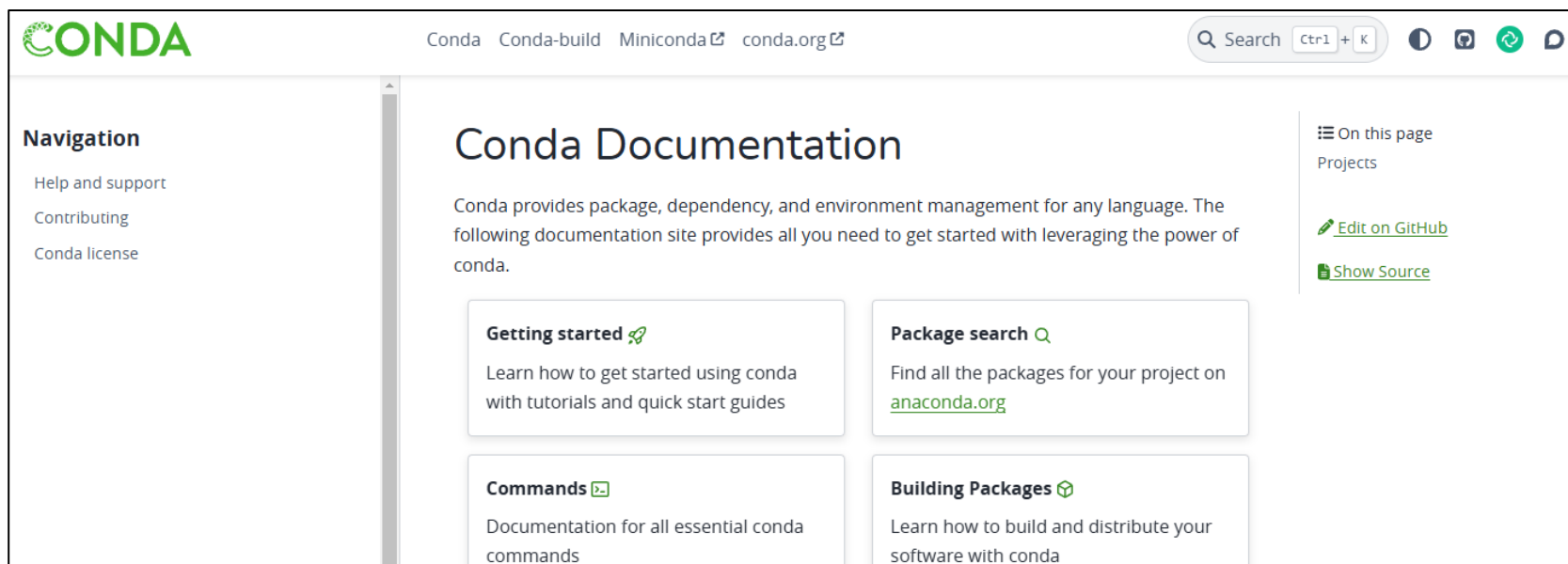
- Instead ...

We aim to create multiple isolated Python environments on our local desktop, each with its own dedicated packages installed for specific tasks.



Conda: A platform for Python package management and creating isolated Python environments

- Conda is a powerful command line tool for package and environment management that runs on Windows, macOS, and Linux



<https://docs.conda.io/projects/conda/en/stable/user-guide/getting-started.html>

<https://docs.conda.io/en/latest/>

Part 1:

Install Anaconda, a distribution that includes conda

Step1: Download Anaconda installers

- Visit the website <https://www.anaconda.com/download/success>, and download the appropriate installer for your computer's operating system

The screenshot shows the Anaconda download page with the following annotations:

- macOS (Apple silicon):** A red box at the top center contains the text: "If you are using macOS, please check your chip is apple or intel chips".
- Windows:** A box on the left points to the "Windows" section. Under "Python 3.12", the "64-Bit Graphical Installer (912.3M)" is highlighted with a purple box.
- Mac:** A box in the center points to the "Mac" section. Under "Python 3.12", the "64-Bit (Apple silicon) Command Line Installer (707.3M)" and "64-Bit (Intel chip) Command Line Installer (731.2M)" are highlighted with purple boxes.
- Linux:** A box on the right points to the "Linux" section. Under "Python 3.12", the "64-Bit (x86) Installer (1007.9M)" is highlighted with a purple box.
- macOS (Intel chip):** A box at the bottom center points to the "Mac" section, specifically to the Intel chip installers.

The Anaconda website header includes the logo, navigation links (Products, Solutions, Resources, Partners, Company), and "Sign Up" / "Sign In" buttons. The main heading is "Anaconda Installers" with a "Download" button.

Step2: Install Anaconda - Linux

- In your “/home/your_account_name” directory, create a “bin” folder for installing Anaconda later. (This step is optional, depending on your preference)

```
bojunliu@chem68-25dhcp: ~  
bojunliu@chem68-25dhcp:~$ pwd  
/home/bojunliu  
bojunliu@chem68-25dhcp:~$ mkdir bin  
bojunliu@chem68-25dhcp:~$ ls  
bin Desktop Documents Downloads Music Pictures Public Templates Videos  
bojunliu@chem68-25dhcp:~$
```

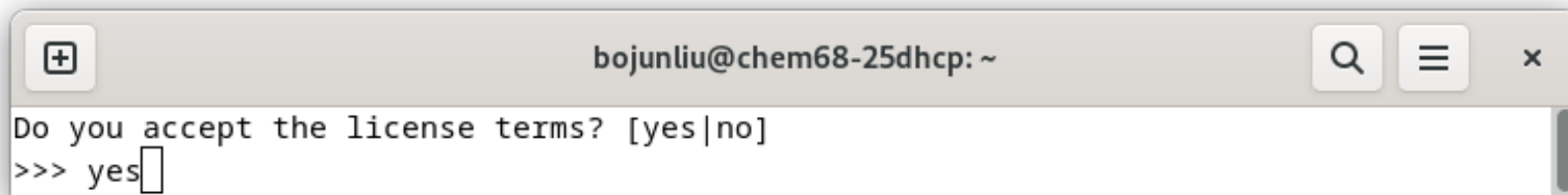
- Find your downloaded Anaconda installer (for me, its in my “./Downloads” directory) and execute the following commands.

```
bojunliu@chem68-25dhcp:~$ chmod -x ./Downloads/Anaconda3-2024.10-1-Linux-x86_64.sh  
bojunliu@chem68-25dhcp:~$ bash ./Downloads/Anaconda3-2024.10-1-Linux-x86_64.sh  
  
Welcome to Anaconda3 2024.10-1  
  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>>
```

If you're not using the Bash shell, make sure to use the appropriate shell instead.

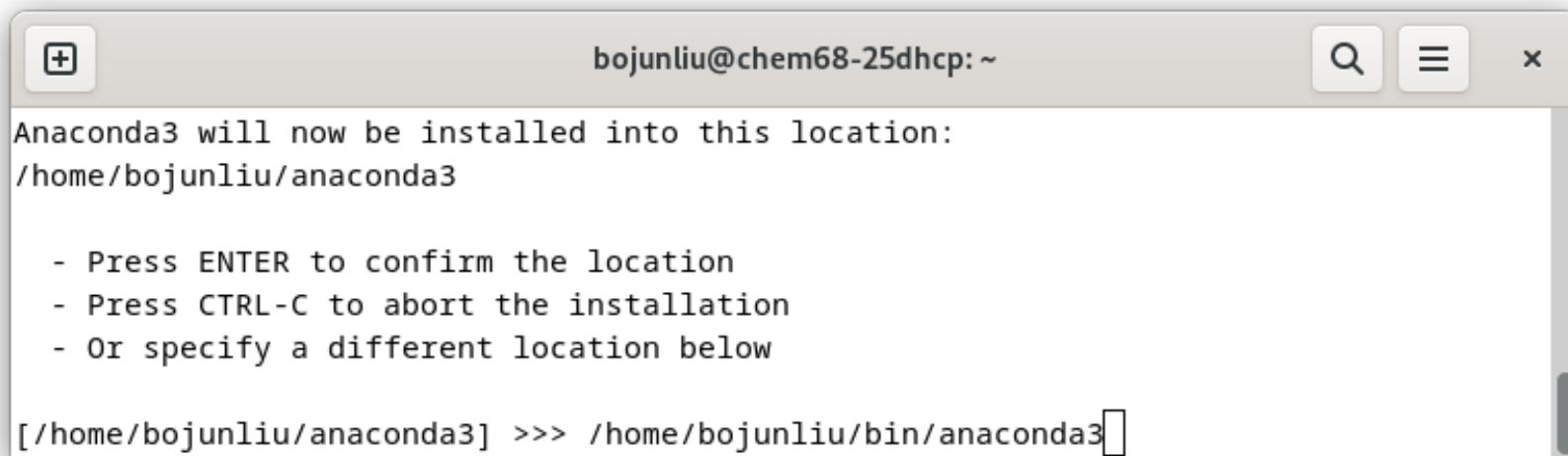
Step2: Install Anaconda - Linux

- Follow the instructions (press “q” to continue when you finish reading the license”).



```
bojunliu@chem68-25dhcp: ~  
Do you accept the license terms? [yes|no]  
>>> yes
```

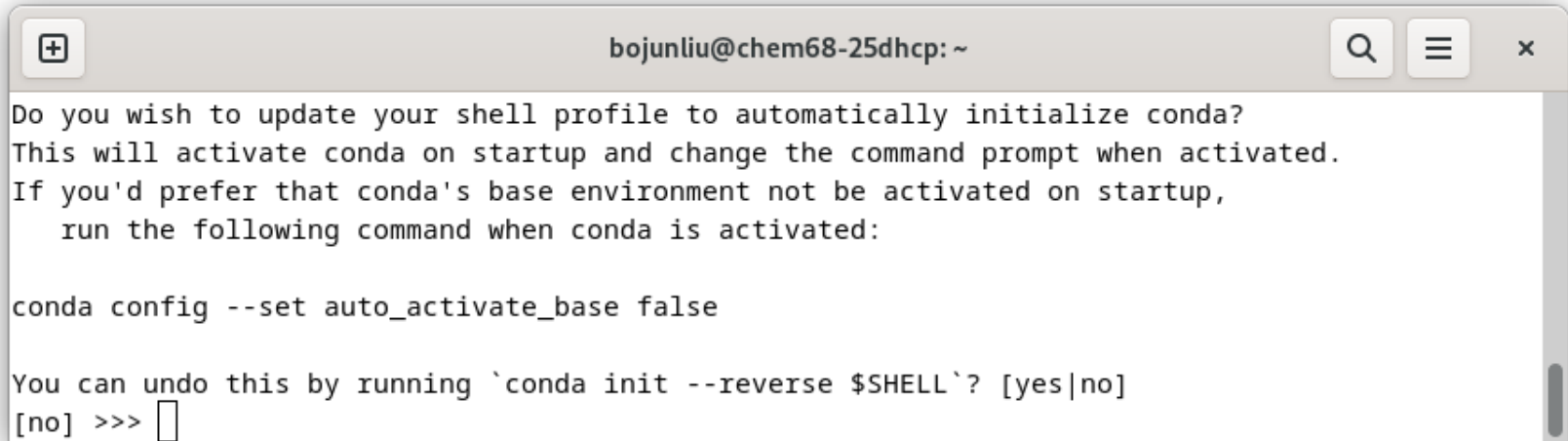
- By default, Anaconda is installed in your “/home/your_account_name” directory. Previously, we have created a “bin” folder, and we can specify our installation location in that folder.



```
bojunliu@chem68-25dhcp: ~  
Anaconda3 will now be installed into this location:  
/home/bojunliu/anaconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/bojunliu/anaconda3] >>> /home/bojunliu/bin/anaconda3
```

Step2: Install Anaconda - Linux

- Last step:



```
bojunliu@chem68-25dhcp: ~  
Do you wish to update your shell profile to automatically initialize conda?  
This will activate conda on startup and change the command prompt when activated.  
If you'd prefer that conda's base environment not be activated on startup,  
    run the following command when conda is activated:  
  
conda config --set auto_activate_base false  
  
You can undo this by running `conda init --reverse $SHELL`? [yes|no]  
[no] >>> 
```

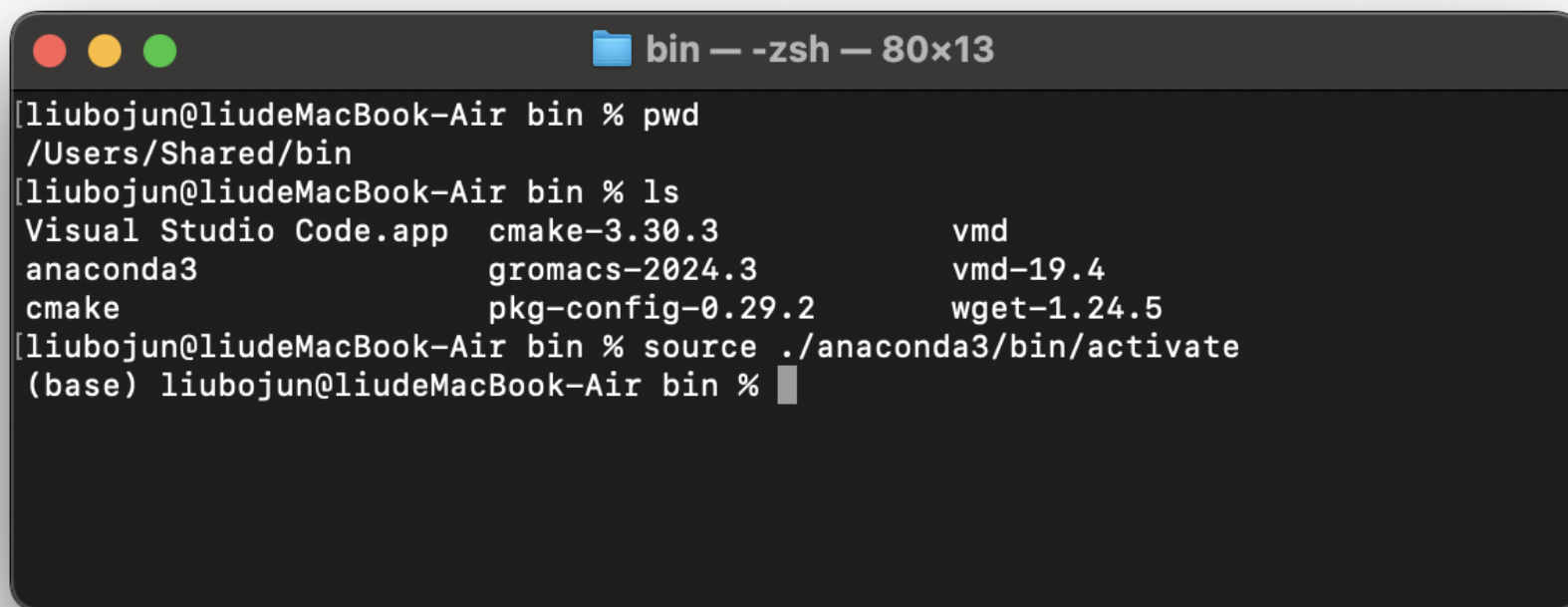
- If you press yes, you are all set! Conda will automatically initialize every time you open a shell.
- If you press no, you will need to activate conda manually using the **source** command. See below. Be sure to adjust the path based on your anaconda3 directory.



```
bojunliu@chem68-25dhcp: ~  
bojunliu@chem68-25dhcp:~$ source /home/bojunliu/bin/anaconda3/bin/activate  
(base) bojunliu@chem68-25dhcp:~$ 
```

Step2: Install Anaconda - macOS

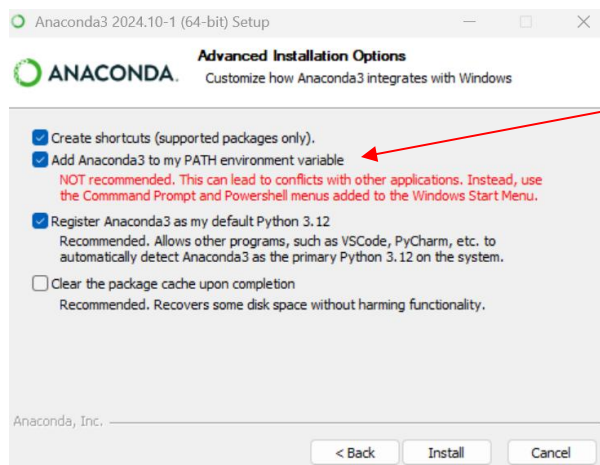
- On macOS, you can follow the same steps in the Terminal as on Linux. The only difference is where you install Anaconda. Personally, I create a “Shared” folder in the “/Users” directory, then a “bin” folder inside it, and install Anaconda there. This is optional and depends on your preference.



```
bin — -zsh — 80x13
[liubojun@liudeMacBook-Air bin % pwd
/Users/Shared/bin
[liubojun@liudeMacBook-Air bin % ls
Visual Studio Code.app  cmake-3.30.3          vmd
anaconda3               gromacs-2024.3        vmd-19.4
cmake                   pkg-config-0.29.2     wget-1.24.5
[liubojun@liudeMacBook-Air bin % source ./anaconda3/bin/activate
(base) liubojun@liudeMacBook-Air bin %
```

Step2: Install Anaconda - Windows

- Double click the downloaded executive file “Anaconda3-2024.10-1-Windows-x86_64.exe” for installation
- Proceed installation but pay attention to the following step:
 - Select “Add Anaconda3 to my PATH environment variable”, especially when new to Python.



**PLEASE
SELECT
!!!**

- After installation: open a Command Prompt terminal (via typing “cmd” in your search bar), and enter “conda activate” in terminal.

```
Command Prompt
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\24153>conda activate

(base) C:\Users\24153>
```

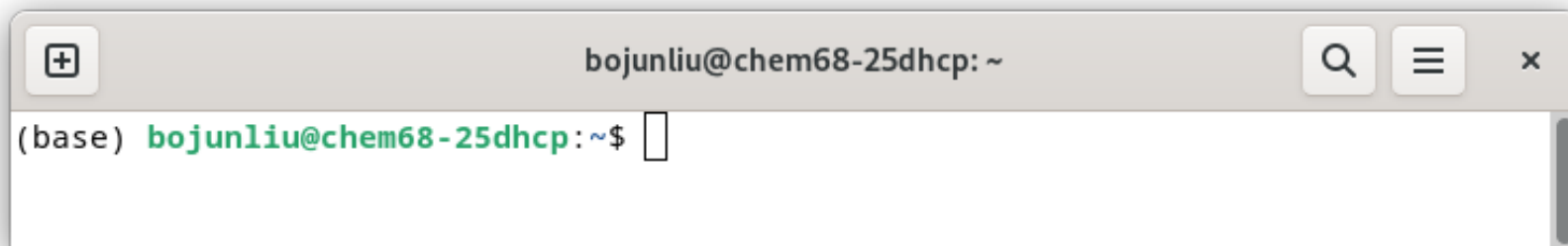
“base” environment will be activated after the “conda activate” command, if installed correctly. Otherwise, let the TA know.

Part 2:

Set up Python environment using Conda

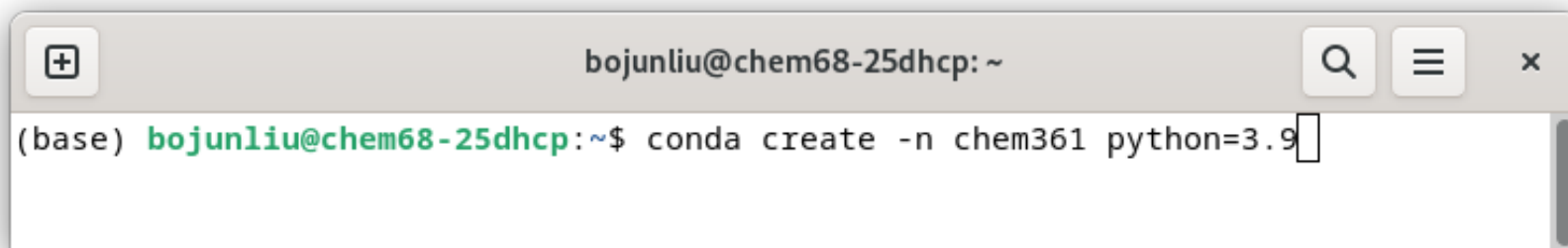
Step3: Create Python (conda) environment

- Now you have successfully installed anaconda3 and opened terminal with “conda” activated.



```
bojunliu@chem68-25dhcp: ~  
(base) bojunliu@chem68-25dhcp:~$
```

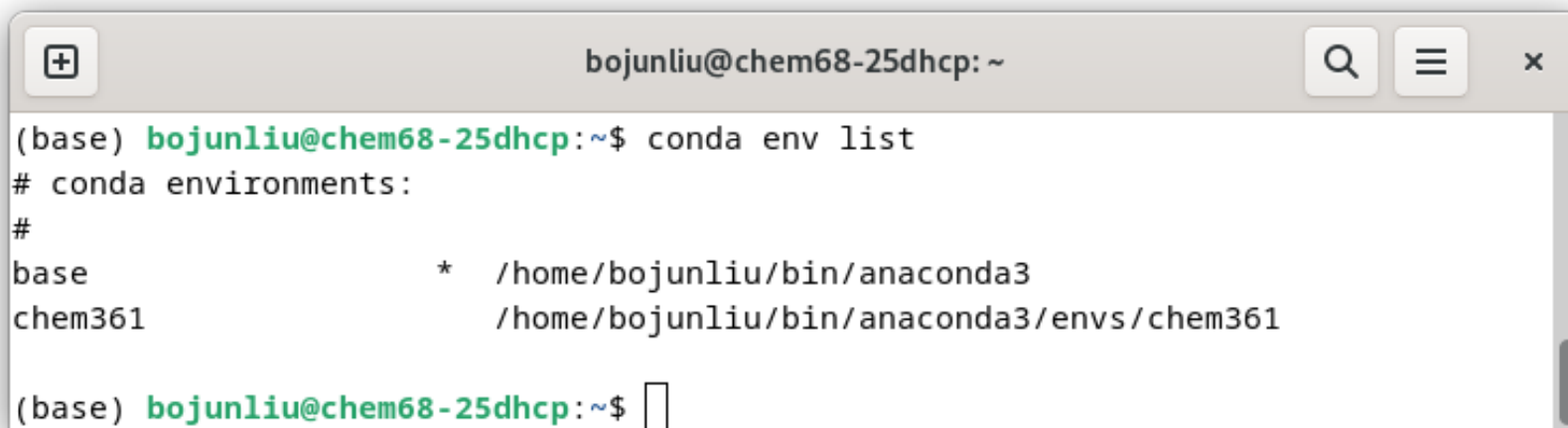
- “(base)” indicates you’re in the base environment. We typically avoid modifying the base environment directly and instead create new Python environments for different tasks.
- To create a Python environment, use the **conda create** command. You can specify the environment’s name with the **-n** flag and set the Python version. For example, the command below creates an environment named “chem361” with Python 3.9 in it.



```
bojunliu@chem68-25dhcp: ~  
(base) bojunliu@chem68-25dhcp:~$ conda create -n chem361 python=3.9
```

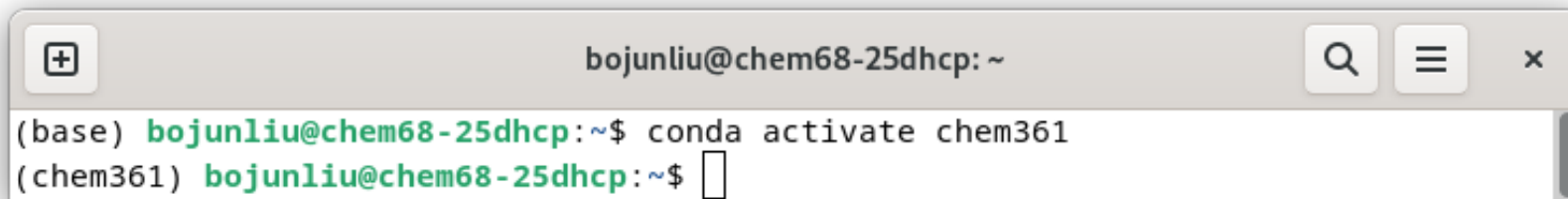
Step3: Create Python (conda) environment

- Now you have successfully created a new Python environment named “chem361”. You can use the **conda env list** command to see your newly created environment.



```
bojunliu@chem68-25dhcp: ~  
(base) bojunliu@chem68-25dhcp:~$ conda env list  
# conda environments:  
#  
base * /home/bojunliu/bin/anaconda3  
chem361 /home/bojunliu/bin/anaconda3/envs/chem361  
  
(base) bojunliu@chem68-25dhcp:~$
```

- However, you're still in the base environment. Run **conda activate chem361** to switch to the newly created environment “chem361”. The prompt “(chem361)” indicates you are now in that environment.



```
bojunliu@chem68-25dhcp: ~  
(base) bojunliu@chem68-25dhcp:~$ conda activate chem361  
(chem361) bojunliu@chem68-25dhcp:~$
```

Step4: Install Python packages

- Now that you've created a new Python environment, it only includes Python by default. To use it effectively for data analysis or machine learning, you'll need to install additional Python packages. For example, ...

Array-based
computing



Scientific
computing

Machine
Learning



Deep
Learning

Data
Visualization



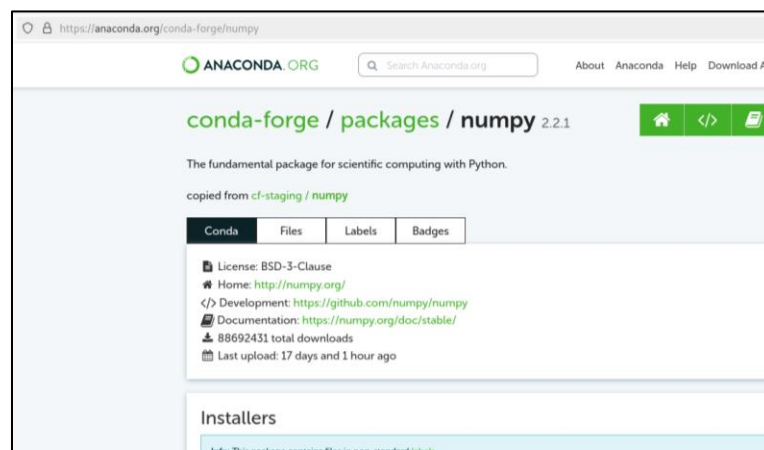
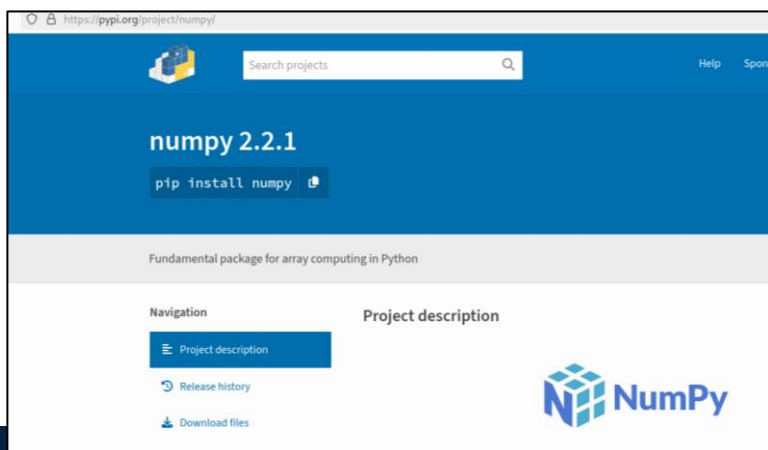
Chem-
Informatics

Open-Source Cheminformatics
and Machine Learning

Step4: Install Python packages

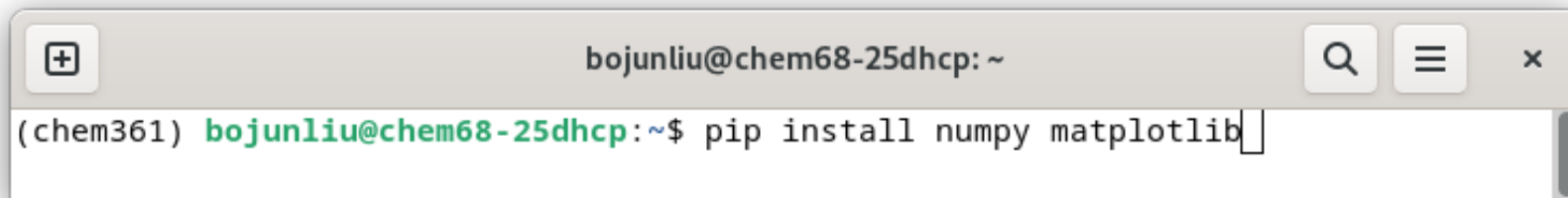
- Before we begin, let me introduce two standard methods for installing Python packages.
 - **conda install <pkg_name> = <pkg_version>**
 - **pip install <pkg_name> == <pkg_version>**
- Both methods can be used to install Python packages. While **pip install** is often simpler and faster, **conda install** can automatically handle dependencies within your environment.
- When you use **pip install**, packages are retrieved from PyPI at <https://pypi.org>. In contrast, **conda install** pulls packages from the Anaconda repository at <https://anaconda.org/anaconda/repo>. If you specify **conda install -c conda-forge**, it also searches for packages in the conda-forge channel at <https://conda-forge.org/packages/>.
- For instance, the package “numpy” can be found in both PyPI and the Anaconda repository.

If you don't specify the pkg_version, the newest version will be automatically selected.



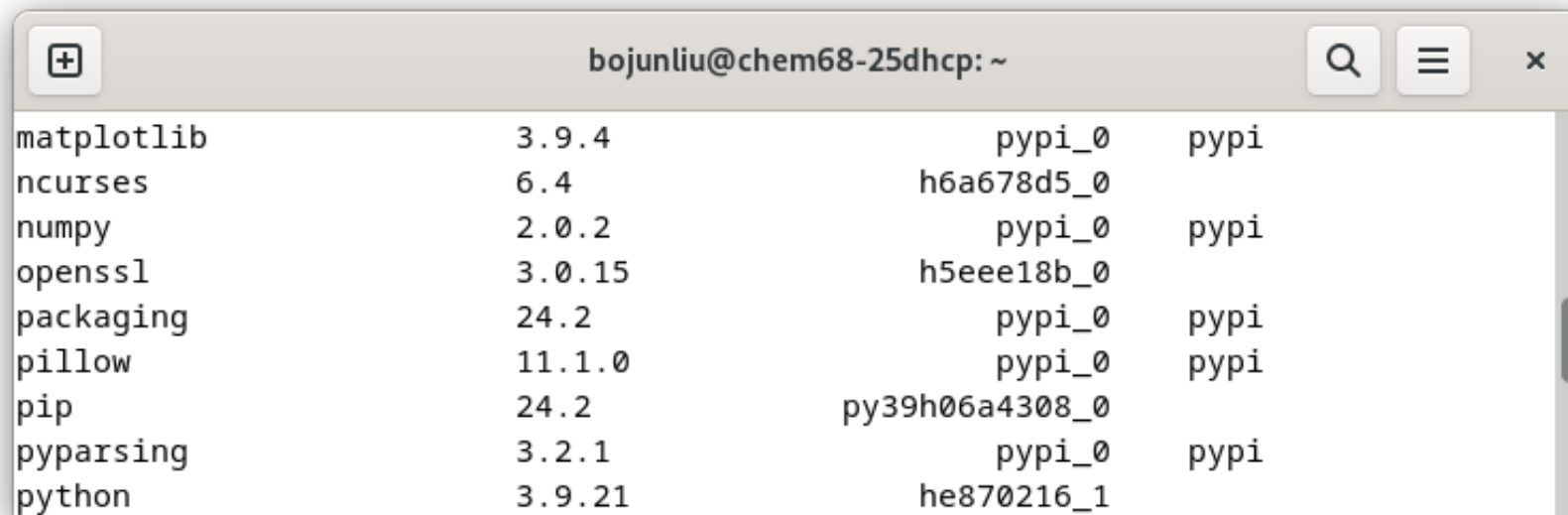
Step4: Install Python packages

- Now let's start installing Python packages. For example, “numpy” and “matplotlib”:



```
bojunliu@chem68-25dhcp: ~  
(chem361) bojunliu@chem68-25dhcp:~$ pip install numpy matplotlib
```

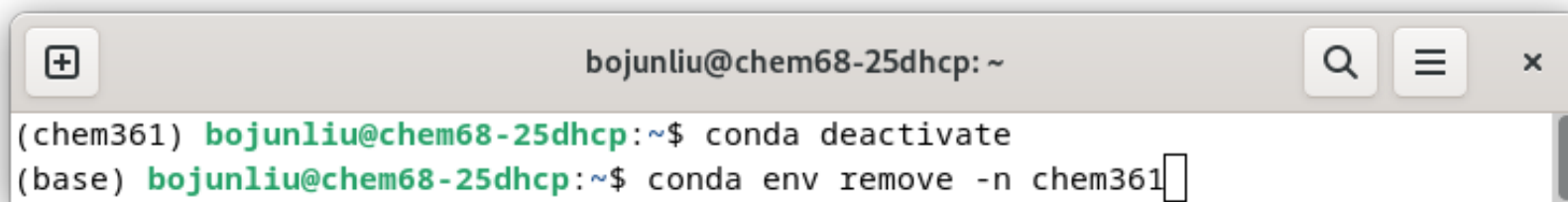
- Use the command **conda list** , you will see “numpy” and “matplotlib” have been installed in your Python environment.



```
bojunliu@chem68-25dhcp: ~  
matplotlib      3.9.4          pypi_0    pypi  
ncurses         6.4            h6a678d5_0  
numpy           2.0.2          pypi_0    pypi  
openssl        3.0.15         h5eee18b_0  
packaging       24.2           pypi_0    pypi  
pillow          11.1.0         pypi_0    pypi  
pip             24.2           py39h06a4308_0  
pyparsing       3.2.1          pypi_0    pypi  
python          3.9.21         he870216_1
```

Step4: Install Python packages

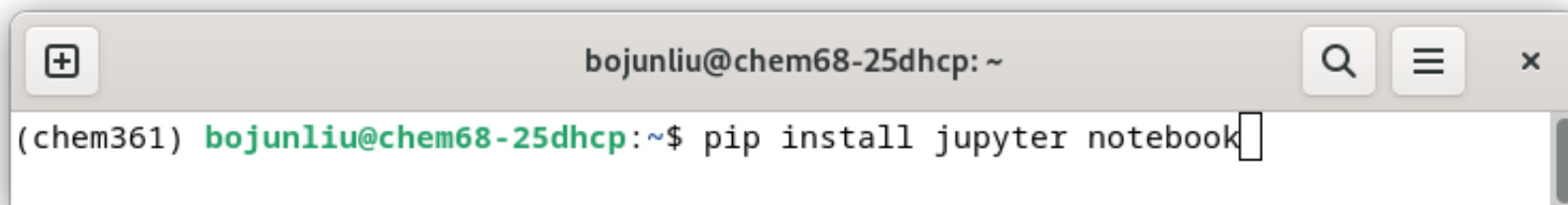
- In the future, we will release the required Python dependencies for each homework project.
- Be mindful when setting up your Python environment to ensure compatibility among different packages. If you encounter any issues, you can use **pip uninstall <pkg_name>** or **conda uninstall <pkg_name>** (depending on how the package was installed). However, we recommend creating a new environment if problems persist, as this approach often simplifies resolving conflicts.
- When creating another Python environment, remember to run **conda deactivate** to return to the base environment. To remove an environment you previously created, use **conda env remove -n <env_name>** .

A terminal window with a title bar showing 'bojunliu@chem68-25dhcp: ~'. The window contains two lines of text: '(chem361) bojunliu@chem68-25dhcp:~\$ conda deactivate' and '(base) bojunliu@chem68-25dhcp:~\$ conda env remove -n chem361' followed by a cursor.

```
bojunliu@chem68-25dhcp: ~  
(chem361) bojunliu@chem68-25dhcp:~$ conda deactivate  
(base) bojunliu@chem68-25dhcp:~$ conda env remove -n chem361
```

Step5: Run Python code: jupyter notebook

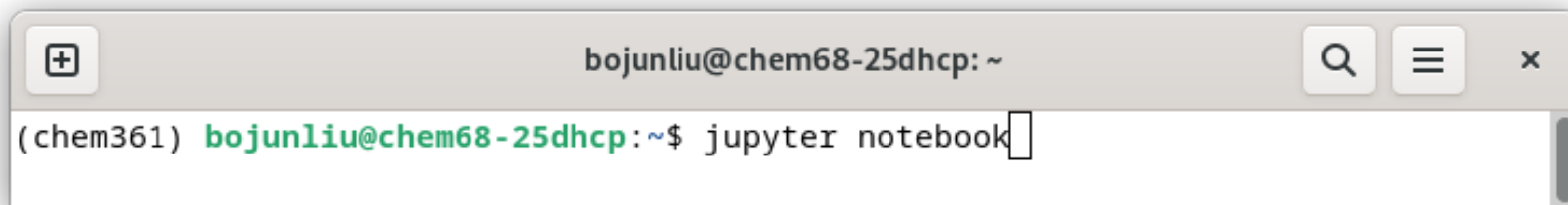
- We recommend using jupyter notebooks to write and run Python code for your projects.
- Firstly, you need to install jupyter, notebook in your Python environment.



A terminal window with a title bar showing 'bojunliu@chem68-25dhcp: ~'. The prompt is '(chem361) bojunliu@chem68-25dhcp:~\$' and the command 'pip install jupyter notebook' is entered. The window includes search, menu, and close icons in the top right corner.

```
(chem361) bojunliu@chem68-25dhcp:~$ pip install jupyter notebook
```

- Then, type **jupyter notebook**.



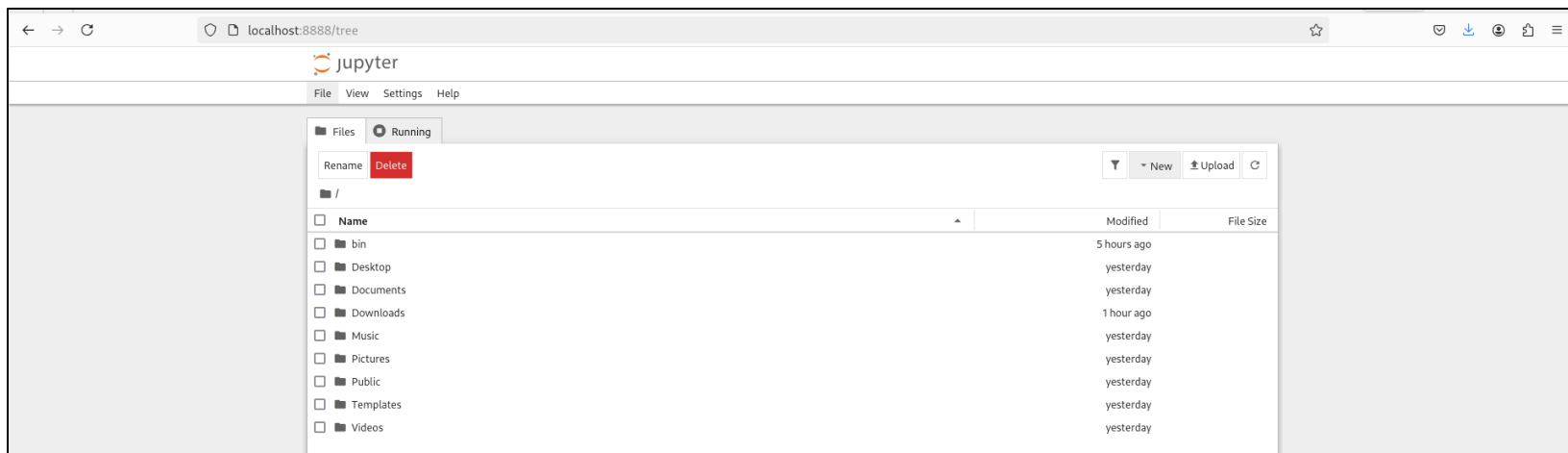
A terminal window with a title bar showing 'bojunliu@chem68-25dhcp: ~'. The prompt is '(chem361) bojunliu@chem68-25dhcp:~\$' and the command 'jupyter notebook' is entered. The window includes search, menu, and close icons in the top right corner.

```
(chem361) bojunliu@chem68-25dhcp:~$ jupyter notebook
```

- The Jupyter server will open in your web browser (press **Ctrl + C** in your terminal to stop it).

Step5: Run Python code: jupyter notebook

- Your home directory will be shown here.

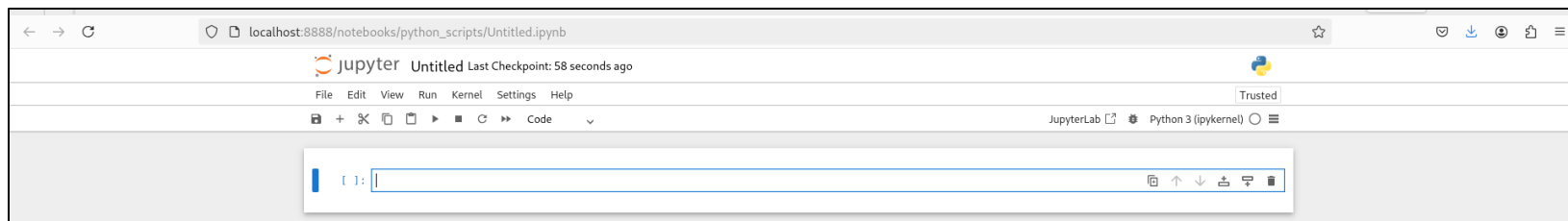


- Create a Python 3 (ipykernel) file.

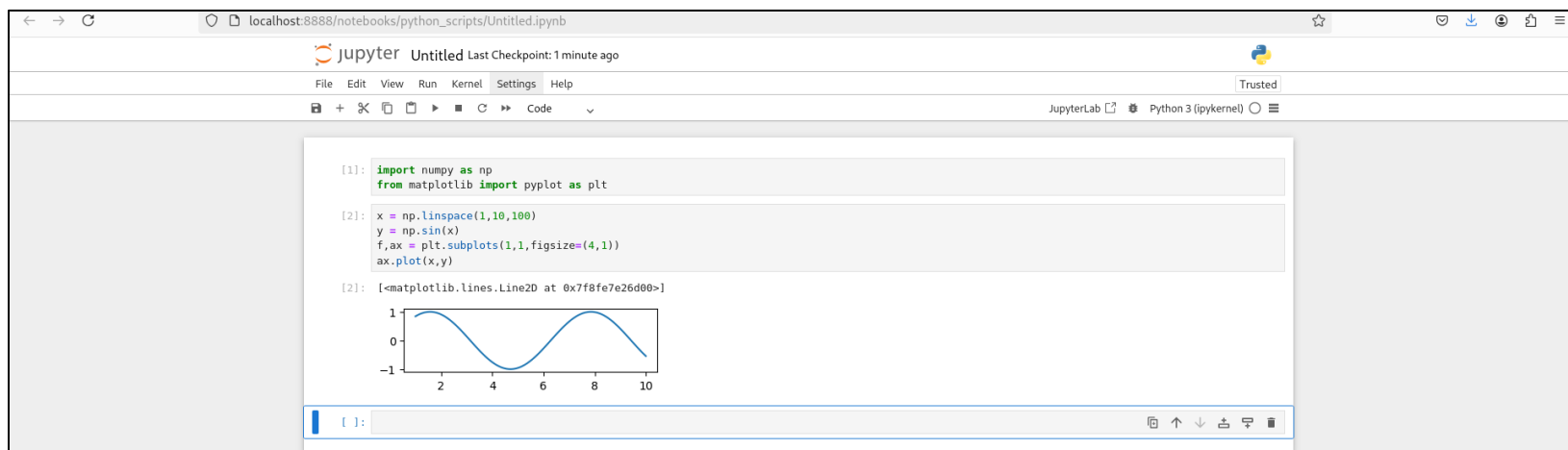


Step5: Run Python code: jupyter notebook

- You can start running Python code in this jupyter notebook!



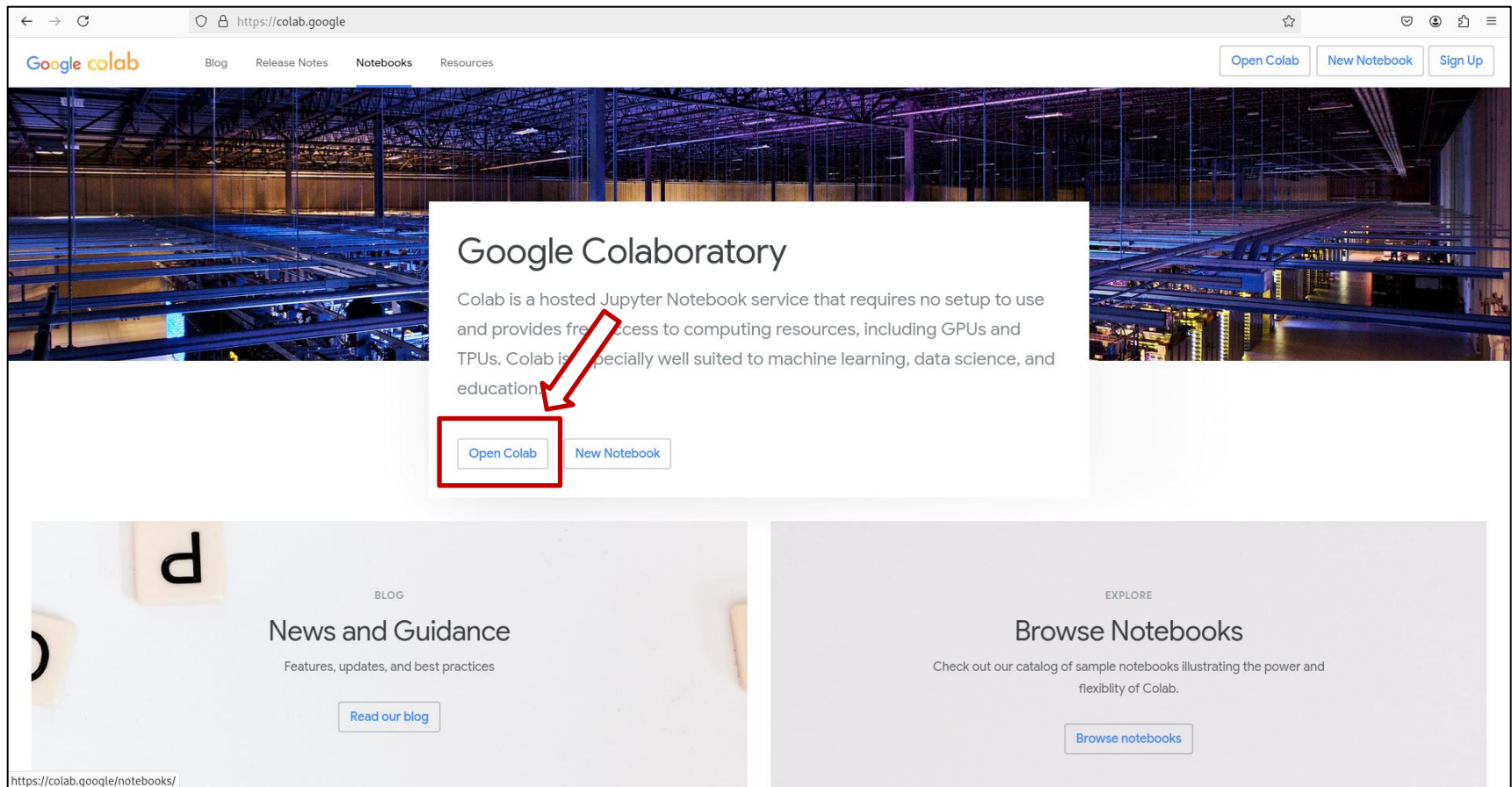
- Visit the official Jupyter Notebook documentation at <https://jupyter-notebook.readthedocs.io/en/latest/notebook.html> to learn about its full range of features.
- Let's try it out by plotting a sine function. Press **Shift + Enter** to execute each cell.



Part 3:

Use Python in Google Colab

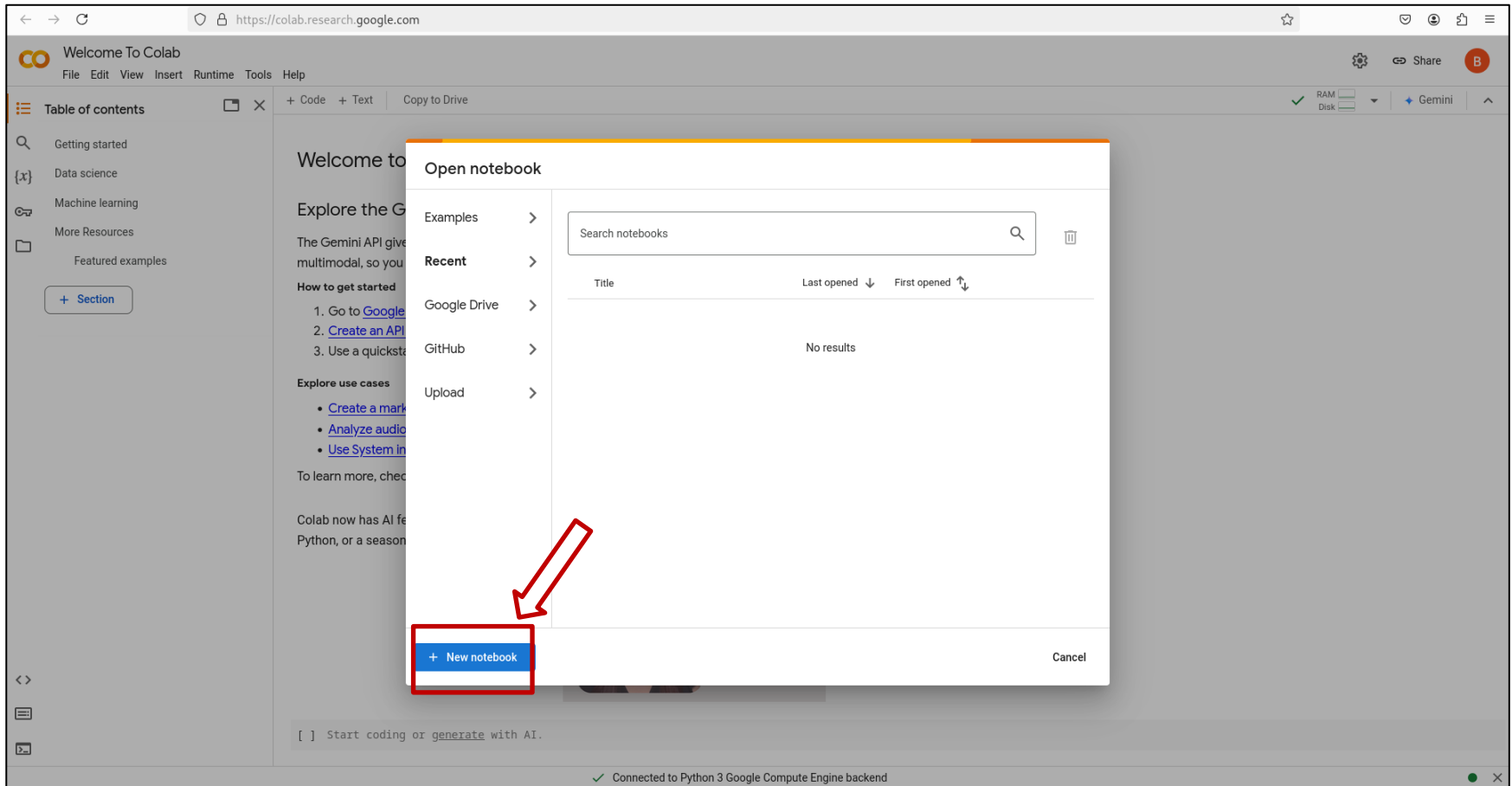
Google Colab: A cloud-based platform for running Python code and collaborating on data science projects in Jupyter notebooks



<https://colab.google/>

Open a notebook

- Open new notebook.



Check the available Python packages

- The Google Colab server usually comes pre-installed with a variety of Python packages. You can use **!pip list** command to check the already installed Python packages. For example, you can see numpy and matplotlib in our previous example are already installed!

https://colab.research.google.com/drive/1p2TsGDfTeCvApCyTUG4SCeclGtnXGytlscrollToTTRRIvZ2iHS

Untitled2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Code Text

[20] pip list

Markdown	3.7
markdown-it-py	3.0.0
MarkupSafe	3.0.2
matplotlib	3.10.0
matplotlib-inline	0.1.7
matplotlib-venn	1.1.1
mdit-py-plugins	0.4.2
msdml	0.1.2
minikanren	1.0.3
missingno	0.5.2
mistune	3.1.0
mizani	0.13.1
nkl	2025.0.1
nl-dtypes	0.4.1
nlxtend	0.23.3
note-iter-tools	10.5.0
noteipy	1.0.3
openath	1.3.0
nsgpack	1.1.0
multidict	6.1.0
multipledispatch	1.0.0
multitasking	0.0.11
murmurhash	1.0.11
music21	9.3.0
nameex	0.0.8
natrwhals	1.21.1
natsort	8.4.0
nbclassic	1.1.0
nbclient	0.10.2
nbconvert	7.16.5
nbformat	5.10.4
ndindex	1.9.2
nest-asyncio	1.6.0
networkx	3.4.2
nibabel	5.3.2
nlTK	3.9.1
notebook	6.5.5
notebook_shim	0.2.4
numba	0.60.0
numba-cuda	0.0.0
numpy	1.26.4
nvidia-cublas-cu12	12.0.4.1
nvidia-cuda-cupti-cu12	12.6.10
nvidia-cuda-nvcc-cu12	12.6.85

1s completed at 4:32 PM

- If you want to install Python packages not being included, you can use **!pip install <pkg_name>** command. The "!" at the beginning is specific to Jupyter Notebook or IPython environments. It allows you to run shell commands directly from a notebook cell.
- You can try our previous example to plot a sine function in Google Colab!