DEPARTMENT OF CIVIL ENGINEERING, IIT MADRAS

CV 5100 Modeling, Uncertainty and Data for Engineers (MUDE)
July - November 2025

Assignment #0: Setting the System Up

This course is inspired by and based on its homonymous course developed by 50+ educators at TU Delft over the last five years. We are using a lot of their hardwork with permission to bring this to you.

Instructions before the session:

You should have completed this section before coming to the tutorial.

1. Make sure to sign up for a new GitHub account using your *@smail.iitm.ac.in* email before coming for the tutorial. **Github.com → Sign up**.
2. *Only after* successfully signing up for GitHub, add your name and email address on the Google sheet here https://tinyurl.com/github-email-mude.

Instructions during the session:

1. You must complete this assignment during the live tutorial session. Note that this assignment sets up the programming system for all the upcoming assignments

Objectives:

At the end of this assignment, you will have learned how to:

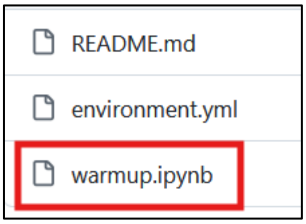
1. Show hidden files on your computer.
2. Install *miniconda* software package.
3. Install *Visual Studio Code*.
4. Set up MUDE environment.
5. Execute a *Python* code in your environment.
6. Set up *GitHub*; install *GitHub Desktop*, and try basic git operations.
7. Instructions for all Assignments.
8. Instructions for Assignment-0.

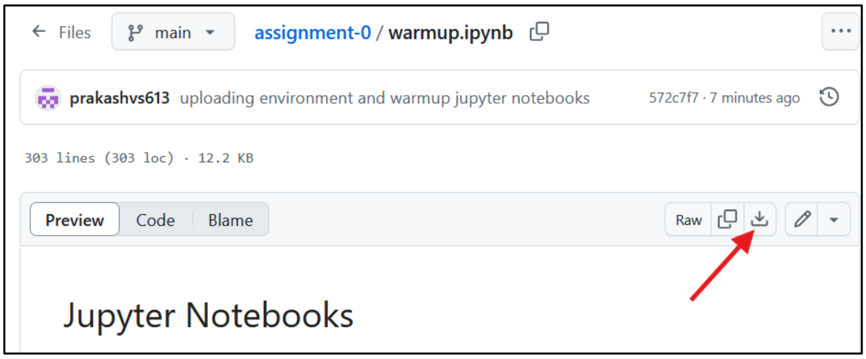
Downloads:

We will use these files later in the assignment. Download the following files from https://github.com/mude-sel-2025/assignment-0

1. environment.yml
2. warmup.ipynb

To download, click on the file and then click on the download icon as shown below.





prakashvs613 uploading environment and warmup jupyter notebooks 572c7f7 · 7 minutes ago

303 lines (303 loc) · 12.2 KB

Software in brief:

- *Miniconda* is a free, miniature installation of Anaconda Distribution that includes
  - Python programming language,
  - conda: an environment manager, and
  - a small number of useful packages, including Python package manager (pip).
- *Visual Studio Code* (VS Code or VSC) is an integrated development environment (IDE) of choice because of its ease-of-use and flexibility.
  - Note that VS Code is an alternative to Jupyter Notebook or Jupyter Lab IDE's, which you may have used in the past.
- *Jupyter* is a Python-based software
  - Jupyter Notebooks are a file format (extension .ipynb); Jupyter Notebook is also an IDE.
  - We will install Jupyter in our computing environment using conda and interact with Jupyter Notebooks (*.ipynb) using VS Code as our IDE.
- *git* is a version control system, used by a wide variety of engineers and software developers to work on projects in parallel together. It provides benefits such as:
  - tracking changes to files,
  - working side by side with other people, and
  - ability to rollback to previous versions without losing track of newer changes.
- *GitHub* is a cloud-based version control system built around git. It provides additional features to seamlessly create and share repositories. We will use GitHub for assignments.
- *GitHub Desktop* is a GUI-based open source application that helps you to work with code hosted on GitHub.

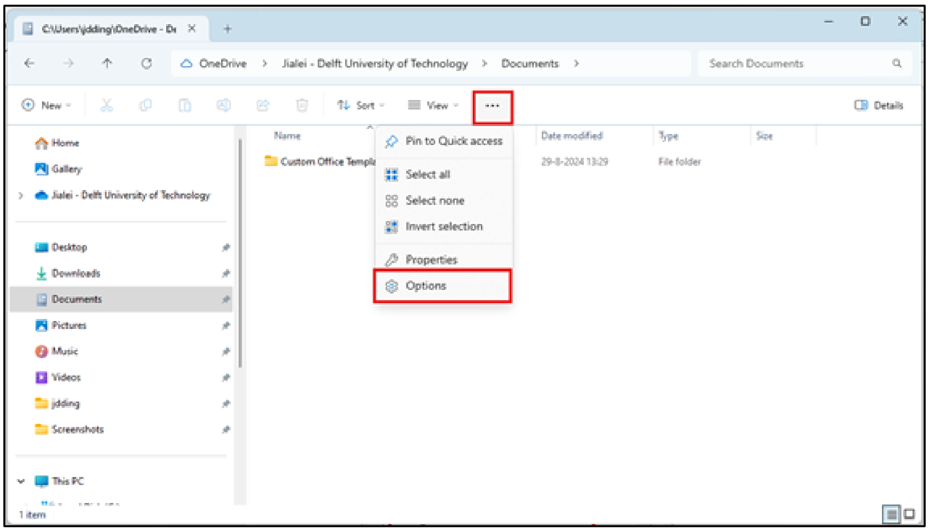
Objective 1: Show hidden files

Mac OS: Simply press three buttons simultaneously, CTRL + SHIFT + .

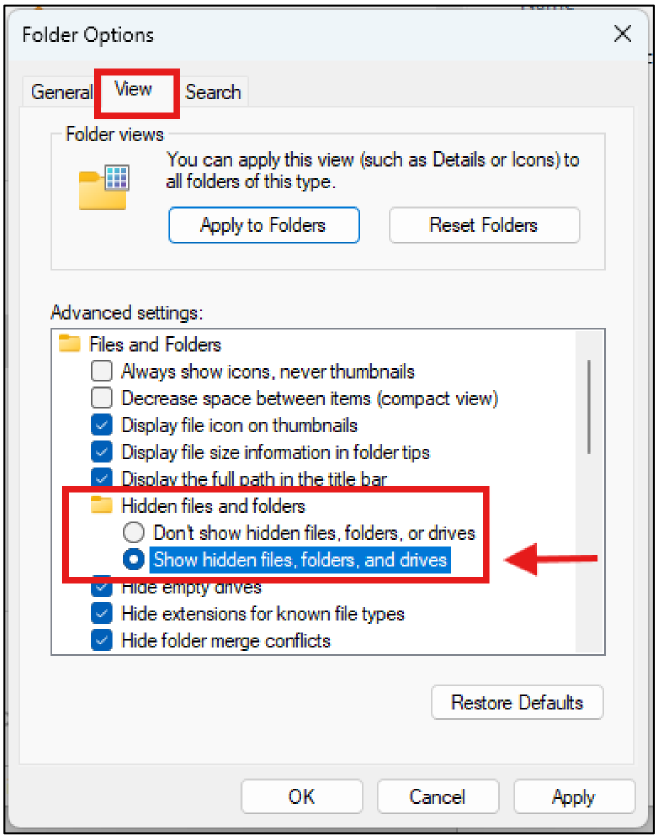
If that does not work, try a Google search on “Finding hidden files on Mac <<*enter your OS version*>>”

Windows OS:

Open Windows Explorer (file explorer/my computer). From the menu bar, find “Options” or “Settings” as follows. Press Alt if you do not see the menu bar.



Navigate to “View” tab and then select the option for “Show hidden files, folders, …”.



You can apply this view (such as Details or Icons) to all folders of this type.

Advanced settings:

Files and Folders
- □ Always show icons, never thumbnails
- □ Decrease space between items (compact view)
- ☑ Display file icon on thumbnails
- ☑ Display file size information in folder tips
- ☑ Display the full path in the title bar

Hidden files and folders
- ○ Don't show hidden files, folders, or drives
- ● Show hidden files, folders, and drives

- ☑ Hide empty drives
- ☑ Hide extensions for known file types
- ☑ Hide folder merge conflicts

Objective 2: Install miniconda

Step 2.1: Download and install miniconda.

Step 2.2: Check whether the installation was successful.

Step 2.3: Use Anaconda prompt.

Note: Unless you installed anaconda or miniconda recently (e.g., within the last year), we recommend you completely remove older versions of these software distributions from your computer and start from scratch.

If you are on macOS, skip Windows-specific instructions.

Step 2.1, 2.2, and 2.3 for Windows

2.1 Download and install miniconda on Windows

Open the command prompt by searching for `cmd` in the Start menu.

Note: The cmd is a kind of Command Line Interface (CLI), a long-serving interface to interact with a computer. The CLI is contrasted against the Graphical User Interface (GUI).

Type in the following three commands *one-by-one* to:

- ❖ quickly and quietly download Windows Installer,
- ❖ rename it to a shorter file name, and then
- ❖ delete the installer.

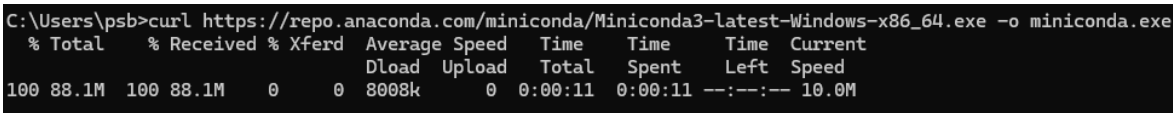
```
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o miniconda.exe
```

```
start /wait "" miniconda.exe /S
```

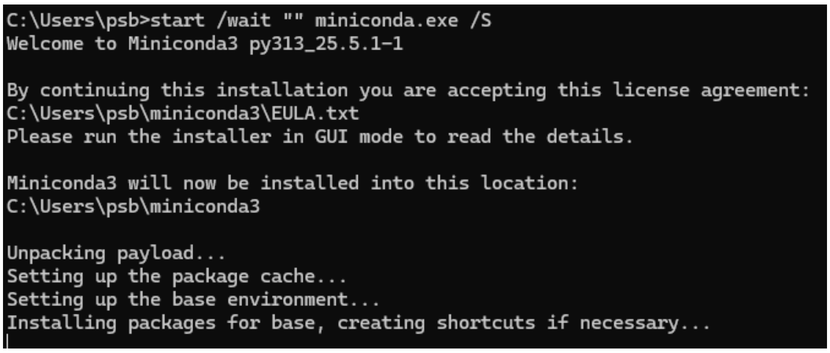
```
del miniconda.exe
```

Notes:

1. If the first command throws error, add --ssl-no-revoke before -o in the first command.
2. If that does not work, download miniconda from https://www.anaconda.com/download, make sure to *Skip registration*.



```
C:\Users\psb>curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o miniconda.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 88.1M  100 88.1M    0     0  8008k      0  0:00:11  0:00:11 --:--:-- 10.0M
```

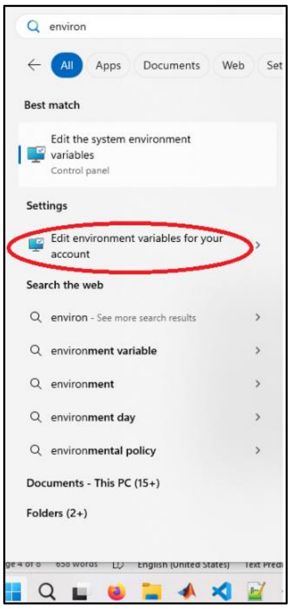


```
C:\Users\psb>start /wait "" miniconda.exe /S
Welcome to Miniconda3 py313_25.5.1-1

By continuing this installation you are accepting this license agreement:
C:\Users\psb\miniconda3\EULA.txt
Please run the installer in GUI mode to read the details.

Miniconda3 will now be installed into this location:
C:\Users\psb\miniconda3

Unpacking payload...
Setting up the package cache...
Setting up the base environment...
Installing packages for base, creating shortcuts if necessary...
```

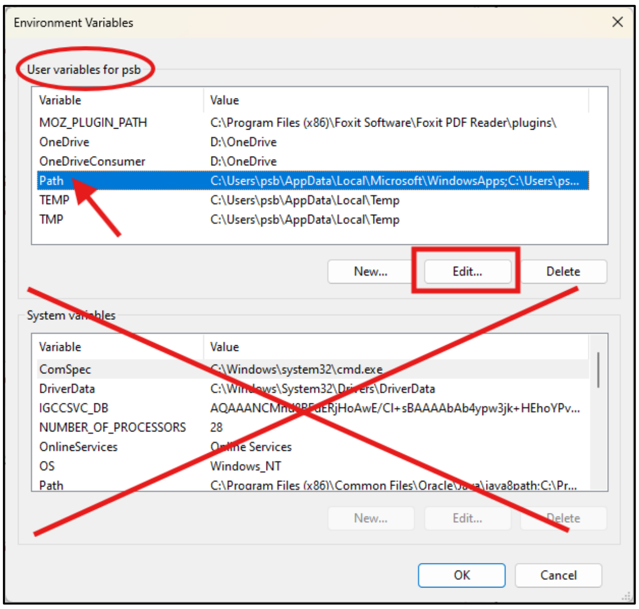
Note: if CTRL + V does not work; you may need to use the menu bar of the terminal window or right-click to select paste.

2.1.1 Add conda.exe on your PATH

1. Find the path of conda.exe. It looks like the following:
   `C:\Users\<<username>>\miniconda3\Scripts\conda.exe`
2. Tell your computer how to find conda.exe, by adding it to ths user-specific *environment variable* called PATH as follows:
   a. Search for "environment " on windows. Select "Edit environment variables for your account".
   b. As a general rule, you should only adjust the PATH variable for the user, not the system. Pay attention to the screenshots below.
   c. Edit the PATH variable by adding the path to conda.exe. If there are no values in PATH, directly enter the conda.exe directory, else click on New as shown below then add the conda.exe directory.



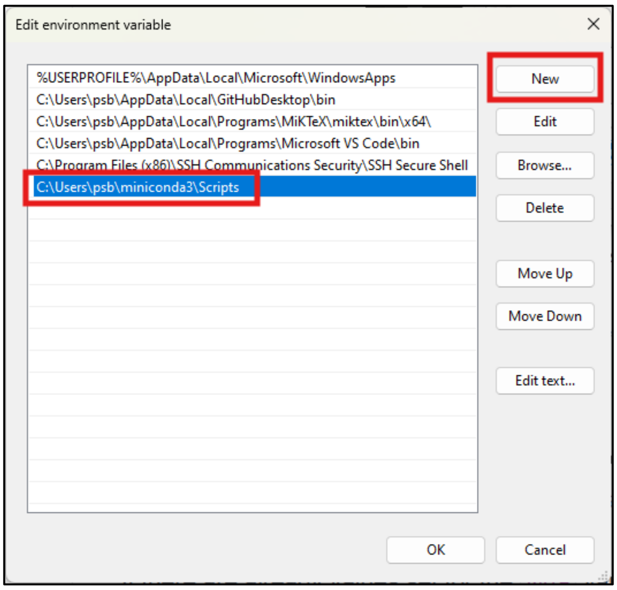
(a)



Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit PDF Reader\plugins\
OneDrive	D:\OneDrive
OneDriveConsumer	D:\OneDrive
Path	C:\Users\psb\AppData\Local\Microsoft\WindowsApps;C:\Users\ps...
TEMP	C:\Users\psb\AppData\Local\Temp
TMP	C:\Users\psb\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
IGCCSVC_DB	AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAbAb4ypw3jk+HEhoYPv...
NUMBER_OF_PROCESSORS	28
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\java8path;C:\Pr...

(b)



```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\psb\AppData\Local\GitHubDesktop\bin
C:\Users\psb\AppData\Local\Programs\MiKTeX\miktex\bin\x64\
C:\Users\psb\AppData\Local\Programs\Microsoft VS Code\bin
C:\Program Files (x86)\SSH Communications Security\SSH Secure Shell
C:\Users\psb\miniconda3\Scripts
```

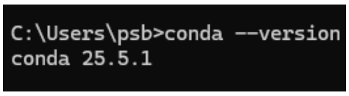
(c)

2.2 Check whether the installation was successful

The easiest way to check if the conda is (1) successfully installed and (2) accessible to the command line interface (CLI).

Run `conda --version` in cmd.

Restart cmd, if it does not work.



```
C:\Users\psb>conda --version
conda 25.5.1
```

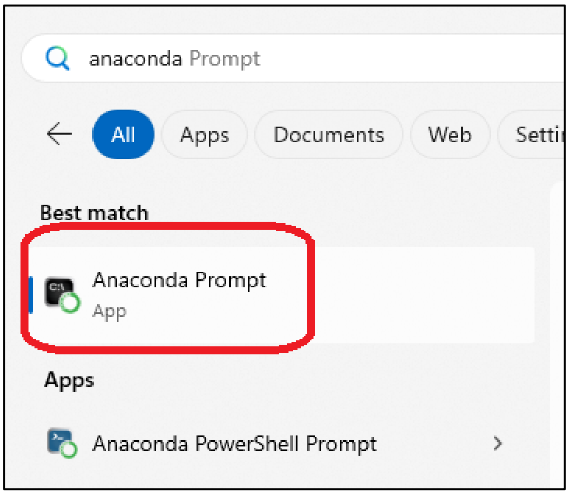
2.3 Use Anaconda prompt

Now that Miniconda is installed on your system, we will use the Anaconda prompt instead of the Windows Command Prompt. Thus, we will bypass the distinction between Windows/Mac for good.

To access Anaconda prompt, search "Anaconda" in windows search. You would see two options:

1. "Anaconda Prompt"
2. "Anaconda PowerShell Prompt"

These are two versions of CLI on Windows. We will stick to "Anaconda Prompt", if needed.



Step 2.1, 2.2, and 2.3 for macOS

2.1 Download and install miniconda on Mac

Open the terminal by either navigating to *Applications → Utilities → Terminal*, or by searching for "Terminal" using `Cmd + Space.`

Type in the following four commands one-by-one to:

- Create directory,
- quickly and quietly download miniconda installer using curl,
- install the miniconda, and then
- delete the installer file.

```
mkdir -p ~/miniconda3
```

```
curl     https://repo.anaconda.com/miniconda/Miniconda3-latest-
MacOSX-arm64.sh -o ~/miniconda3/miniconda.sh
```

```
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
```

```
rm ~/miniconda3/miniconda.sh
```

2.2 Check whether the installation was successful

The easiest way to check if the conda is (1) successfully installed and (2) accessible in the terminal.

Run `conda --version` in terminal.

Restart terminal, if it does not work.

If restarting did not work, add conda.exe to the PATH, with following commands.

```
export PATH="$HOME/miniconda3/bin:$PATH"
```

```
source ~/.zshrc
```

2.3 Use Anaconda prompt

Use terminal to interact with Anaconda. To activate your conda environment, use the following command.

```
conda activate
```

Objective 3: Visual Studio Code

3.1 Download VS Code

Download and install VS Code from https://code.visualstudio.com/Download

---

3.2 Install extensions

Install:

- Python extension: https://marketplace.visualstudio.com/items?itemName=ms-python.python
- Jupyter extension: https://marketplace.visualstudio.com/items?itemName=ms-toolsai.jupyter

Alternatively, you can also use the "Extensions" box on the left side of VS Code, then search for these extensions using their IDs.

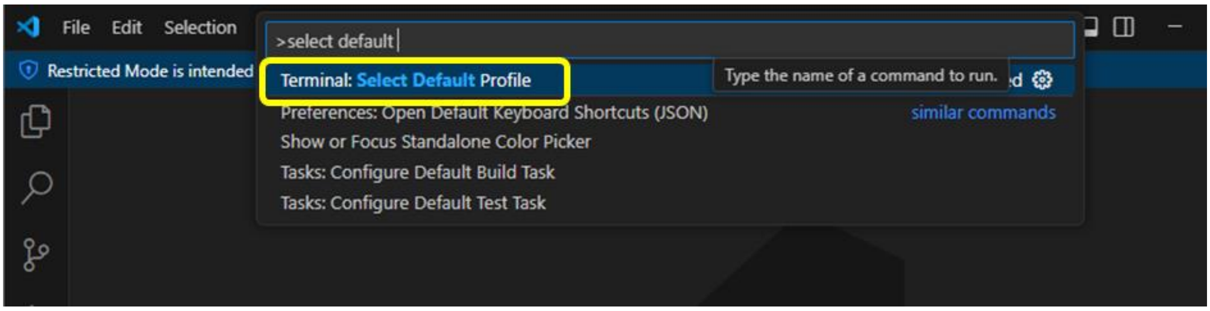
- Python (Extension ID: `ms-python.python`)
- Jupyter (Extension ID: `ms-toolsai.jupyter`)

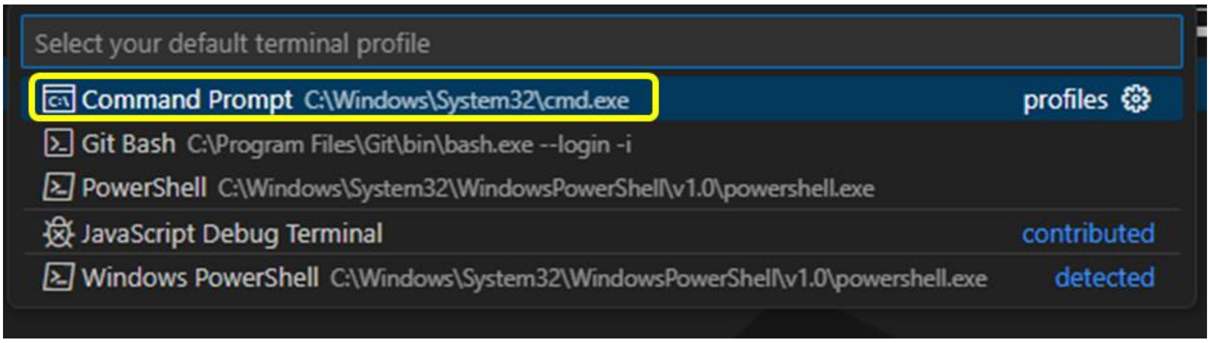
3.3 VS Code Terminal and Command Palette

Familiarize yourself with VS Code. You can customize it for your use after the session. For now, learn the following two:

1) **VS Code Terminal**. VS Code can open all sorts of CLI's. From menu bar, click Terminal → New Terminal.
2) **Command Palette**. Command palette governs all functionality of VS Code, and can be accessed with CTRL + SHIFT + P for Windows and SHIFT + CMD + P for Mac. *This is the single most important VS Code command to remember.*

If your terminal was powershell, you can change the default terminal by opening the command paletter and searching "Terminal: Select Default Profile" and then selecting your preferred CLI from the list.


File
Edit
Selection
>select default
Restricted Mode is intended
Terminal: Select Default Profile
Type the name of a command to run.
Preferences: Open Default Keyboard Shortcuts (JSON)
similar commands
Show or Focus Standalone Color Picker
Tasks: Configure Default Build Task
Tasks: Configure Default Test Task


Select your default terminal profile
Command Prompt C:\Windows\System32\cmd.exe
profiles
Git Bash C:\Program Files\Git\bin\bash.exe --login -i
PowerShell C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
JavaScript Debug Terminal
contributed
Windows PowerShell C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
detected

Objective 4: Set up MUDE environment

4.1 Create/download environment.yml file

`.yml` file is used to create an environment. It tells `conda` to install specific versions of `Python` and then a list of packages, compatible with the Python version. You must have downloaded it following the instructions on page 1.

Alternatively, you can create a new text file in VS Code. Type/paste the following:

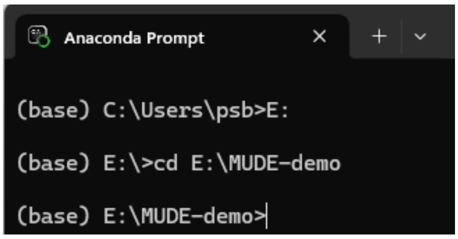
```
name: mude-base

dependencies:

  - python=3.12

  - numpy

  - scipy

  - pandas

  - matplotlib
```

Do not remove spacebars. Save the file as `environment.yml` in a new *working directory* where you will save all your programs for the MUDE course.

4.2 Create an environment for MUDE

Open "Anaconda Prompt". Navigate to the directory you create for all your programs.

Use `cd` to change directory. If your file is on a different drive, use the drive letter followed by a colon (:) to change drive. Then `cd address-to-new-path`.

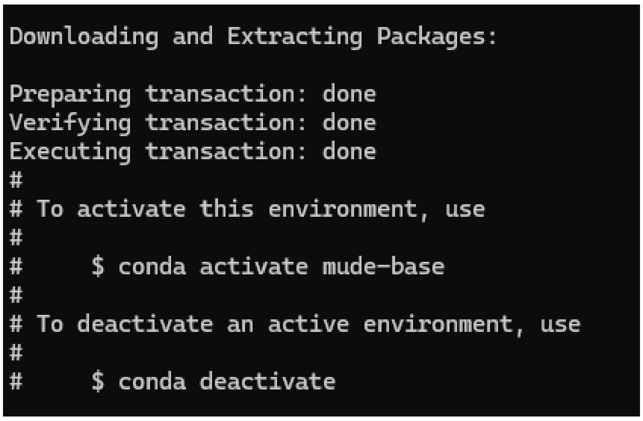


```
(base) C:\Users\psb>E:

(base) E:\>cd E:\MUDE-demo

(base) E:\MUDE-demo>
```

You should already have saved environment.yml in your *working directory* during the previos step. To create a new environment dedicated to the MUDE course, run:

```
conda env create -f environment.yml
```

The switch -f tells conda to use the file environment.yml for the list of desired software and packages. Accept terms of service thrice. Once successfully done, you would see the following message:



```
Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate mude-base
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

The above message indicates that conda successfully created the environment. You need to activate the environment now by running:

```
conda activate mude-base
```

Objective 5: Execute a python code in your environment

5.1 Activate mude-base Python Environment

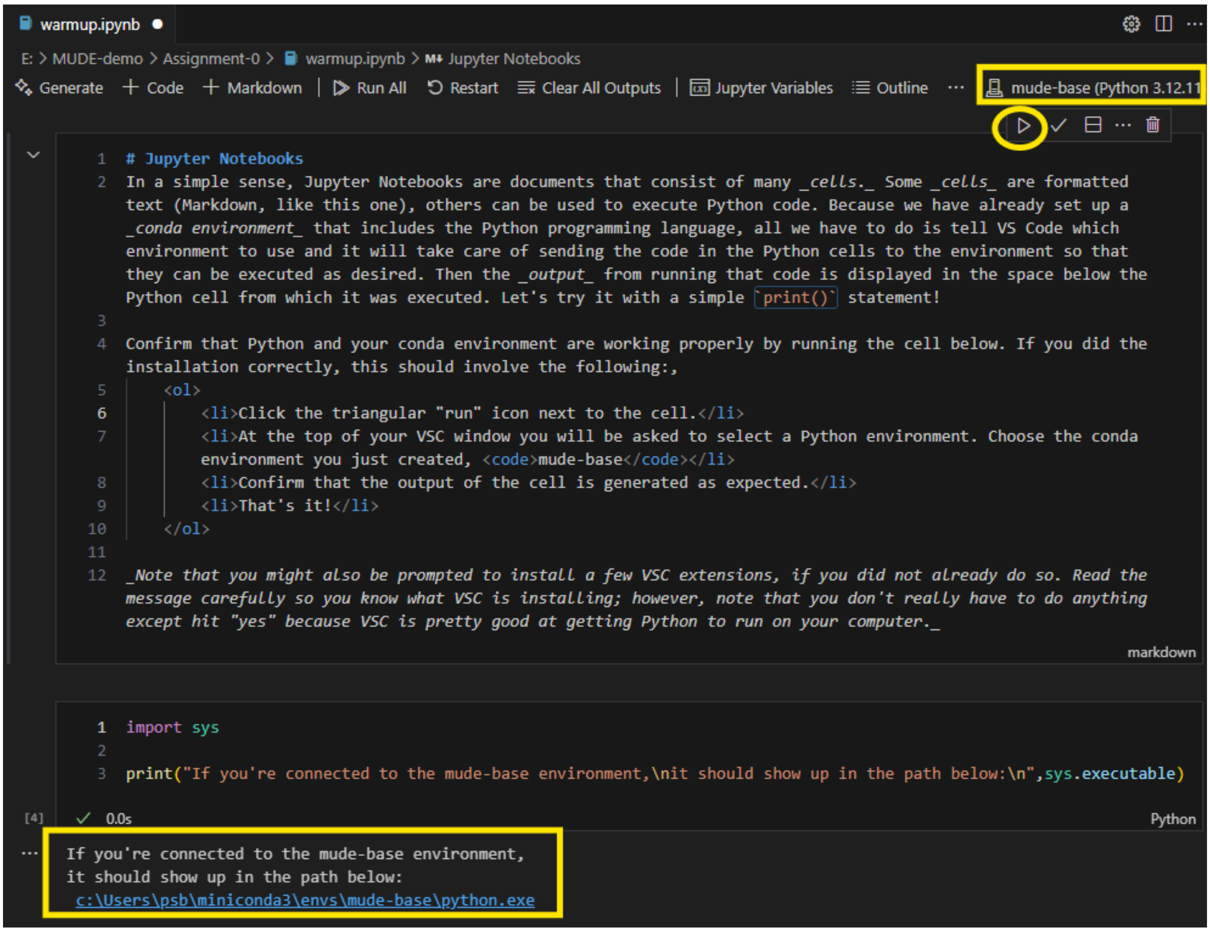
Open `warmup.ipynb` file in VS Code (*download instructions on page 1*).

Read the file.

Make sure to set your environment as `mude-base`.

VS Code may prompt to install additional package such as ipykernel.

Here is the screenshot of a successful set up.



```
# Jupyter Notebooks
In a simple sense, Jupyter Notebooks are documents that consist of many _cells._ Some _cells_ are formatted text (Markdown, like this one), others can be used to execute Python code. Because we have already set up a _conda environment_ that includes the Python programming language, all we have to do is tell VS Code which environment to use and it will take care of sending the code in the Python cells to the environment so that they can be executed as desired. Then the _output_ from running that code is displayed in the space below the Python cell from which it was executed. Let's try it with a simple `print()` statement!

Confirm that Python and your conda environment are working properly by running the cell below. If you did the installation correctly, this should involve the following:,
    <ol>
        <li>Click the triangular "run" icon next to the cell.</li>
        <li>At the top of your VSC window you will be asked to select a Python environment. Choose the conda environment you just created, <code>mude-base</code></li>
        <li>Confirm that the output of the cell is generated as expected.</li>
        <li>That's it!</li>
    </ol>

_Note that you might also be prompted to install a few VSC extensions, if you did not already do so. Read the message carefully so you know what VSC is installing; however, note that you don't really have to do anything except hit "yes" because VSC is pretty good at getting Python to run on your computer._
```

```python
import sys

print("If you're connected to the mude-base environment,\nit should show up in the path below:\n",sys.executable)
```

```
If you're connected to the mude-base environment,
it should show up in the path below:
 c:\Users\psb\miniconda3\envs\mude-base\python.exe
```

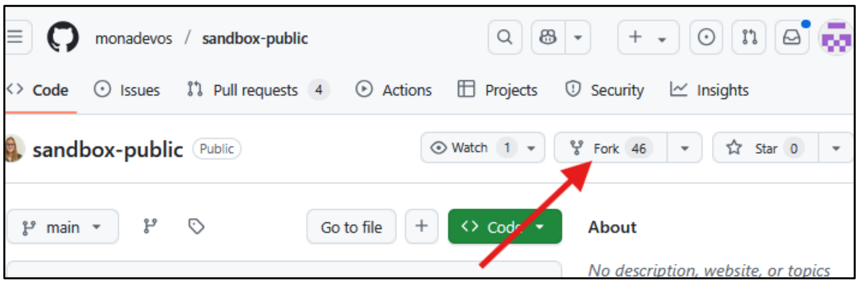
5.2 Go ahead and execute a few cells of the **`warmup.ipynb`**.

Objective 6: Set up GitHub; install GitHub Desktop, and try basic git operations

6.1 Create a sample repository on GitHub

You should already have created an account on GitHub.com with your smail.iitm.ac.in email.

1. Go to https://github.com on the browser.
2. Login.
3. Browse to https://github.com/monadevos/sandbox-public
4. Fork this repository. Name it to something simple.



6.2 Install GitHub Desktop

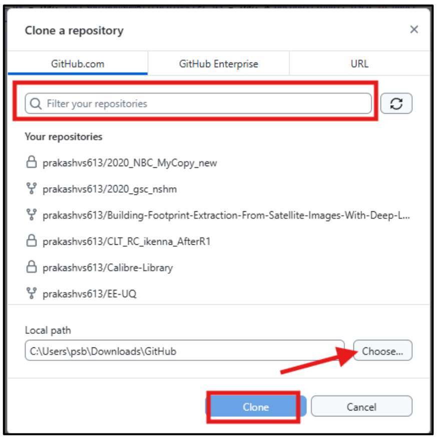
Download and install GitHub Desktop from here https://desktop.github.com/download/.

Login to *GitHub Desktop* account using your browser. This should work seamlessly if GitHub Desktoo directed you to the browser used in the previous step.

6.3 Edit GitHub repository using the local machine

Here are the steps for editing GitHub repo using your local machine. This is the recommended way to update the repo when you want *to make significant changes* (adding files/folders, writing programs, etc.).

1. Clone the sandbox repo on the local computer
2. Create a directory where you wish you store MUDE-related files. *Note: If you are working on a shared computer, you can create a directory in any accessible location (e.g., documents).*
3. Click on “Clone repository”.
4. Search for the sandbox repo by typing first few letters of “sandbox-public”. Select the repo.
5. Click on “Choose” for “Local path”. Navigate to the MUDE-related directory on your local machine.
6. Click “Clone.”



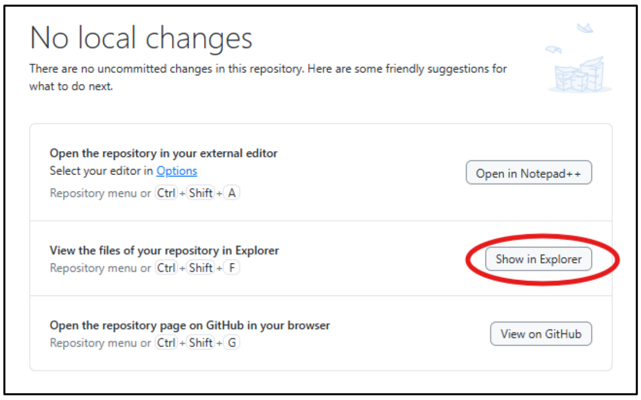
Clone a repository

Your repositories

- prakashvs613/2020_NBC_MyCopy_new
- prakashvs613/2020_gsc_nshm
- prakashvs613/Building-Footprint-Extraction-From-Satellite-Images-With-Deep-L...
- prakashvs613/CLT_RC_ikenna_AfterR1
- prakashvs613/Calibre-Library
- prakashvs613/EE-UQ

Local path: C:\Users\psb\Downloads\GitHub

7. Click on “Show in Explorer” in GitHub Desktop (Alternatively, open file explorer on computer and navigate to the directory where you cloned sandbox repo).



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl + Shift + A

View the files of your repository in Explorer
Repository menu or Ctrl + Shift + F

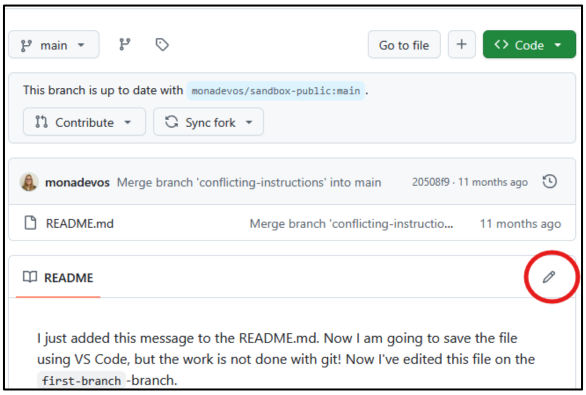
Open the repository page on GitHub in your browser
Repository menu or Ctrl + Shift + G

8. Open ReadMe.md file on your local machine in VS Code. Make some changes. Save the file.
9. Navigate back to GitHub Desktop. You should be able to see the changes you made. GitHub Desktop runs something called “diff” (short for difference) to track all the changes made.
10. Commit the change with a “Summary (required)”. Click on “Commit to main”.
11. A new button with “Push” appears. Click on it.
12. **12.** Check GitHub on browser. If the changes you made on local machine, appears on the web, you have successfully Pushed to the remote.

6.4 Edit GitHub repo using web IDE

Here are the steps for editing GitHub repo using your web browser. This is the recommended way to update the repo when you want *to make minor changes* (updating ReadMe file, a few lines in the program on-the-fly).

1. In the browser, Clone the previously forked Sandbox repo.
2. Either click on the edit button for README directly or click on README.md file and then edit. Screenshots for both are shown below.

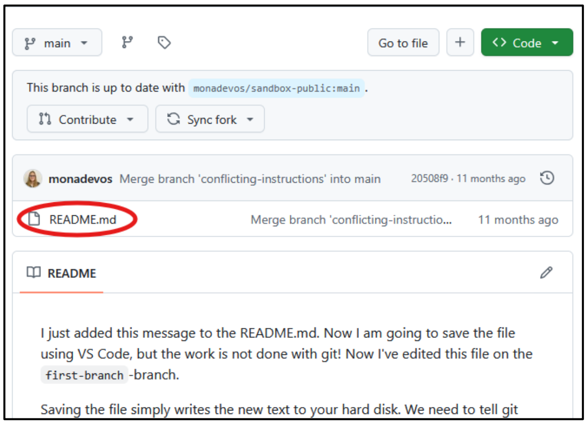


This branch is up to date with monadevos/sandbox-public:main .

Merge branch 'conflicting-instructions' into main

Merge branch 'conflicting-instructio...

I just added this message to the README.md. Now I am going to save the file using VS Code, but the work is not done with git! Now I've edited this file on the first-branch -branch.



This branch is up to date with monadevos/sandbox-public:main .

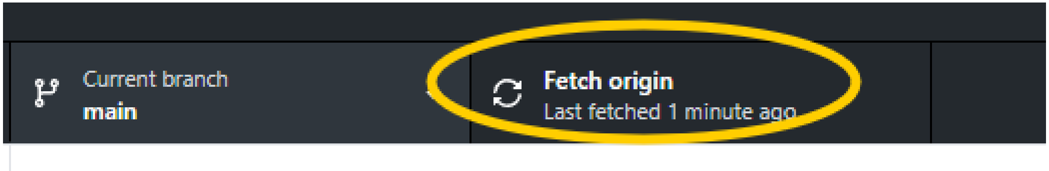
Merge branch 'conflicting-instructions' into main

Merge branch 'conflicting-instructio...

I just added this message to the README.md. Now I am going to save the file using VS Code, but the work is not done with git! Now I've edited this file on the first-branch -branch.

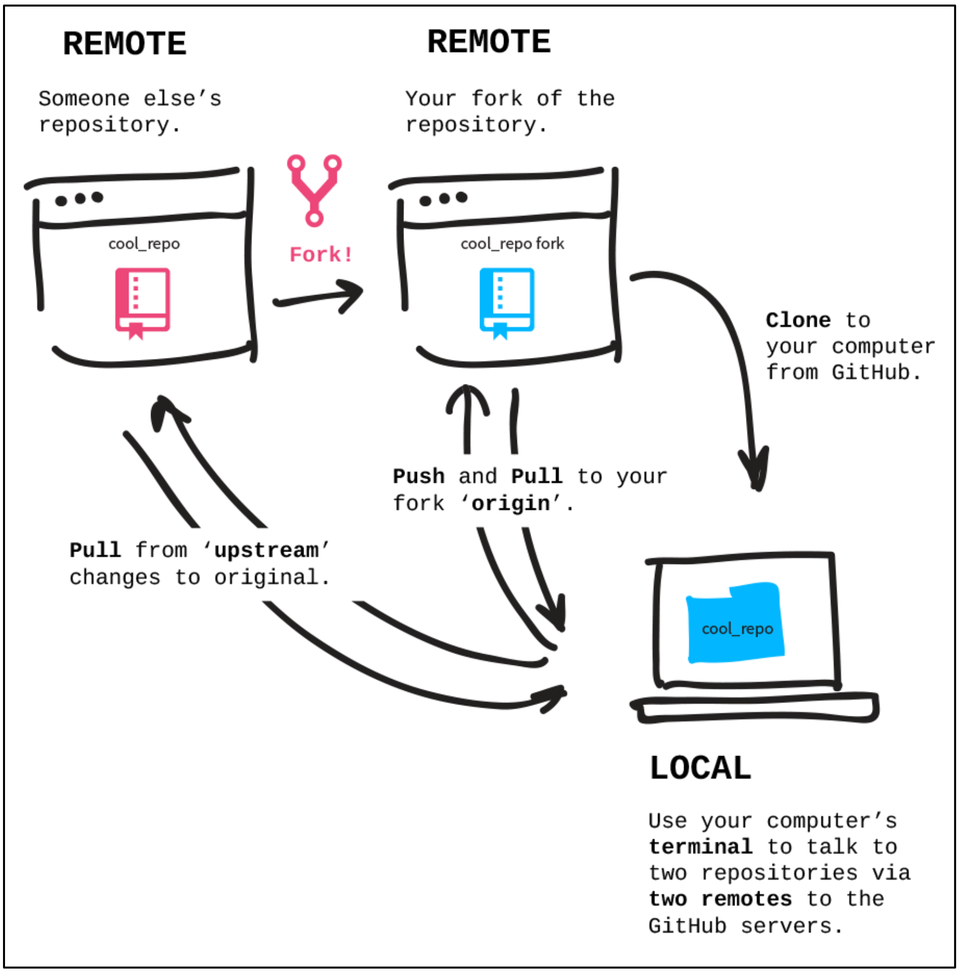
Saving the file simply writes the new text to your hard disk. We need to tell git

3. Make changes to ReadMe.md (markdown file). Commit. Note that commiting on web directly pushes to the main repo.
4. Make sure that you can see the changes made on the main page of the sandbox repo.
5. Come back to GitHub Desktop. Click on "Fetch Origin".



6. You will see a new button for "Pull". Click on it to sync your local repo with remote.
7. Open the ReadMe.md file on your local machine to make sure you can see changes you made on the web.

6.5 What was happening in the background?

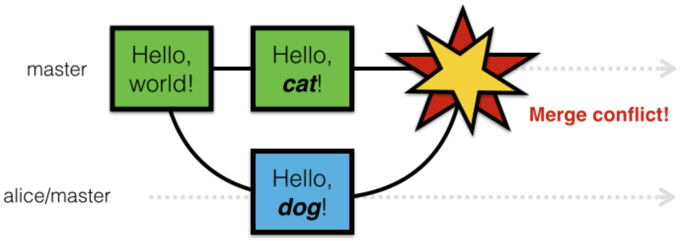

REMOTE
Someone else's repository.
REMOTE
Your fork of the repository.
cool_repo
Fork!
cool_repo fork
Clone to your computer from GitHub.
Push and Pull to your fork 'origin'.
Pull from 'upstream' changes to original.
cool_repo
LOCAL
Use your computer's terminal to talk to two repositories via two remotes to the GitHub servers.

© Jessica Lord

Make sure you understand each of the following:

1. Remote and Local
2. Forking
3. Cloning
4. Push-Pull
5. Commit

In a collaborative project, conflicts may arise, which needs to be manually resolved (not discussed in this session):



Objective 7: Instructions for all assignments

The landing GitHub repo for the course is https://github.com/mude-sel-2025/CV-5100. Whenever in doubt, go to this page. You will find all of the following here:

1. Lecture notes. Stored in different folders. Try to spot already uploaded lecture notes.
2. References.
3. Links to assignments: (a) Click on "Assignment-X repo" to see what does the assignment contain, (b) Click on "Accept Assignment-X (GitHub Classroom)" to accept the assignment. This will fork the Assignment repo and create a personal copy for you.
4. For each assignment, you will see a new link on the landing repo (https://github.com/mude-sel-2025/CV-5100) corresponding to that assignment.
5. Each assignment's repository will contain a pdf with all instructions. Like the one you are reading right now for Assignment-0. You must read the assignment statements entirely and carefully, and then perform all tasks mentioned therein.
6. Submitting the assignment is simply committing and pushing your changes. To be doubly sure, your changes are *pushed*, check the web version of your personal repository.
7. Now, click on "Accept Assignment-0" to accept the assignment. Perform the tasks mentioned below. Once done, submit (push) the assignment.

Objective 8: Instructions for Assignment-0

For Assignment-0, perform the following tasks:

1. Rename the header of the README.md file to reflect your name.
2. Delete the rest of the content in the README.md file and explain what you understand by:
   a) Remote and Local
   b) Forking
   c) Cloning
   d) Push-Pull
   e) Commit
3. Open the *warmup.ipynb* file in VS Code. Run different code cells on your local machine. Push the changes using GitHub Desktop.

(Take-home task) Go through this brief and basic Teach book on Python. This link is also included under References of the landing repo:

https://teachbooks.io/learn-python/01/Theory/01.html