## **Pre-requisites setup**

### **1. WSL setup**

* Install wsl using Command line (running windows PowerShell as administrator),
  + wsl –install (Don’t run this command as this will install latest Ubuntu version)
  + wsl --install -d Ubuntu-22.04 (Run this command)
* Open ubuntu wsl
  + if WslRegisterDistribution failed with error: 0x8004032d ubuntu Error: 0x8004032d (null)
  + In Windows Search Bar, look for "Turn Windows features on or off". Launch it and scroll down until you find "Windows Subsystem for Linux". You'll find it not checked, so you need to check it. Then, your PC will ask your for a reboot. After this, try to launch Ubuntu again and it should work just fine.

### **Close PowerShell**

### **Ubuntu will be installed, run Ubuntu as administrator and follow the below steps**

### **2. Anaconda/Miniconda setup in WSL**

### Create a directory to install minicaonda in

* + mkdir -p ~/miniconda3
* Download latest miniconda version
  + wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh -O ~/miniconda3/miniconda.sh
* Run the install script
  + bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
* Delete the intall script
  + rm -rf ~/miniconda3/miniconda.sh
* Add a conda initialize to your bash
  + ~/miniconda3/bin/conda init bash

(If asked to restart, then restart Ubuntu)

* Verify the installaton
  + conda list

### **3. Apt-get install in WSL for media codecs support**

* sudo apt update
* sudo apt install ffmpeg
* sudo apt install gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly
* Check if ffmpeg code supports AV1: ffmpeg -codecs | grep av1

### **4. Other apt-get installs in WSL**

* sudo apt update
* sudo apt install git-lfs for git large files support. Huggingface needs this. <https://git-lfs.com/>
* Then do git lfs install
* The files tracked by git-lfs are pattern based. If you have a git-lfs repo, you can issue a command git lfs track and find. To add a new file pattern to track, say a zip, you can do git lfs track \*.zip. Then add the .gitattributes file that git lfs track modifies to your repository as follows: git add .gitattributes. In our case this step is not essential as the hugging face lfs repo setup automatically with zip as one of the tracked large file patterns (This is relevant when storing ML models in hugging face repository. You can ignore it for now)

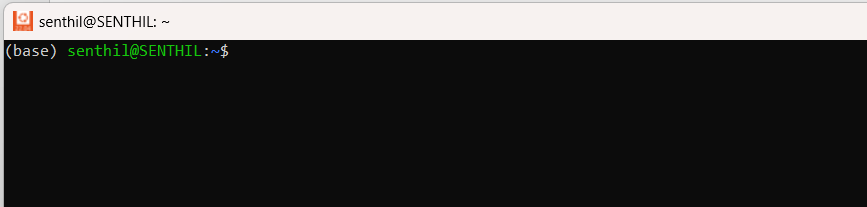
**Nginx Installation:**

Follow the instructions given in the site below and install Nginx in your WSL.

<https://ubuntu.com/tutorials/install-and-configure-nginx#1-overview>

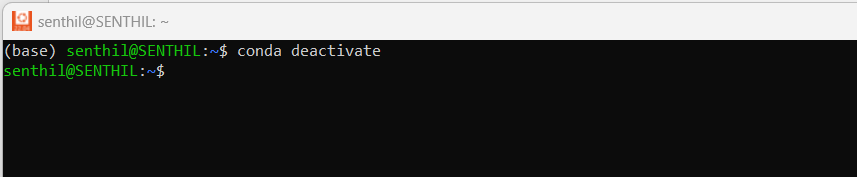
**Setting Up Conda Python environment:**

Open the Ubuntu terminal.



Deactivate the conda environment.

* conda deactivate



Check for any python installed outside the conda environment.

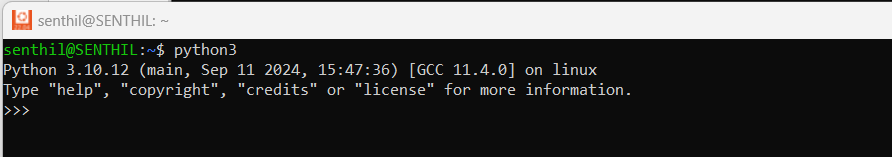
* python
* python3

both should be checked

A screen shot of a computer

Description automatically generated

In my case python3 was there.



To come out of the python script which will show by this >>>

* quit()

this will take you out of the python.

A computer screen with text

Description automatically generated

In my case, since python3 was there so, I will check for python3

* which python3

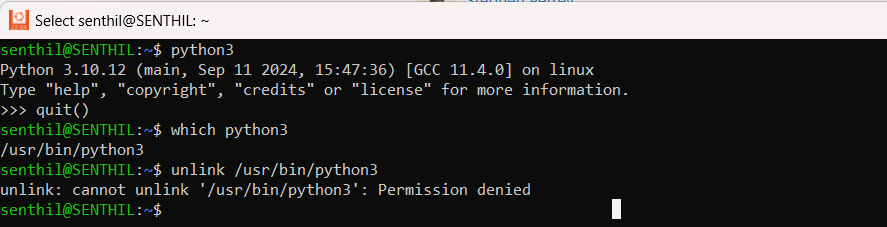
A screenshot of a computer

Description automatically generated

Now unlink the python using the command unlink along with the directory which you got using the which python3 command.

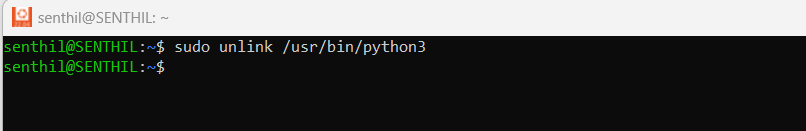
So, in my case it is

* unlink /usr/bin/python3

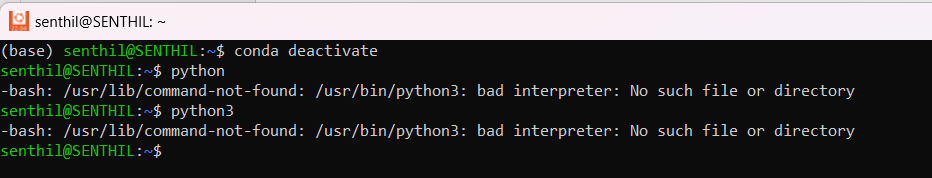


If permission is denied, then use sudo in front of the same command.

* sudo unlink /usr/bin/python3



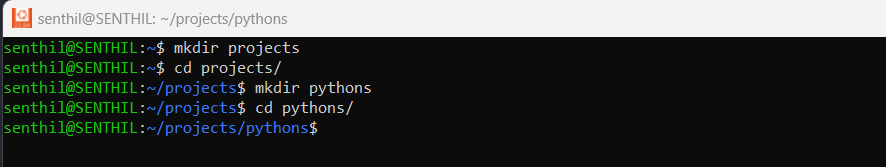
Restart ubuntu, come out of the conda environment and check for python again.



If you don’t get any python then you’re good to go.

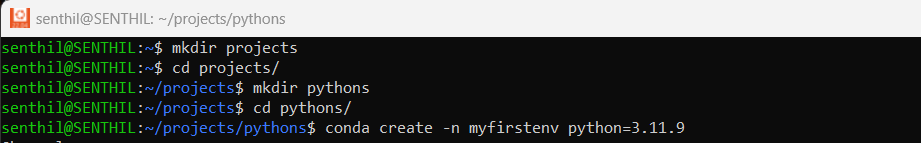
Now, we will create our own conda environment, for that we need to create a folder.

* mkdir projects
* cd projects/
* mkdir pythons
* cd pythons

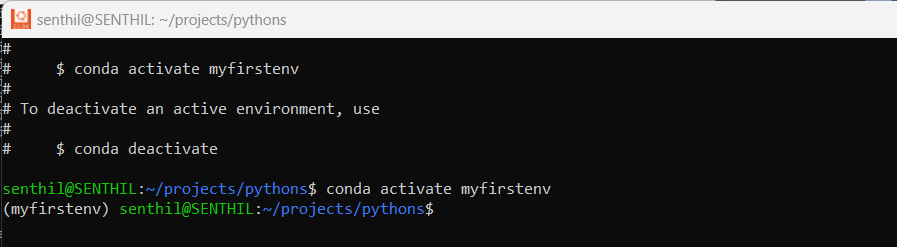


Now create conda environment.

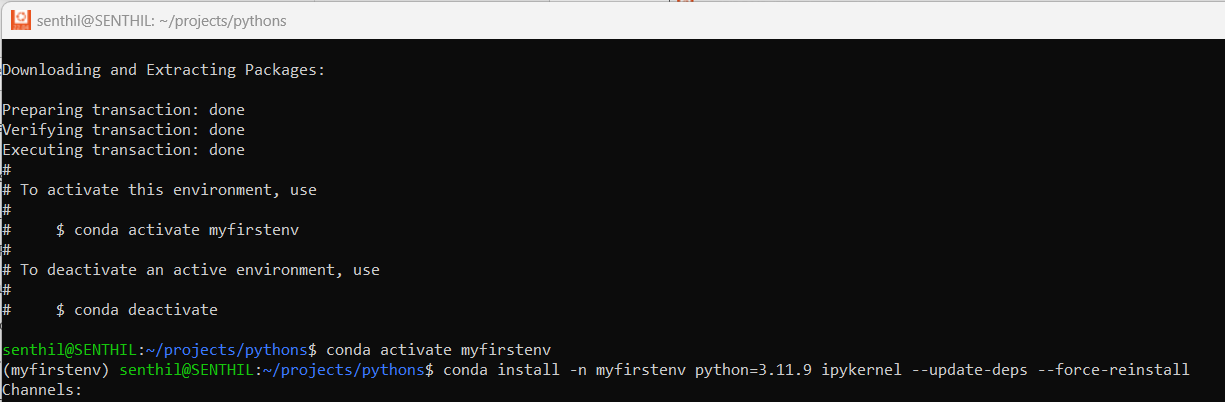
* conda create -n myfirstenv python=3.11.9



* conda deactivate



* conda activate myfirstenv

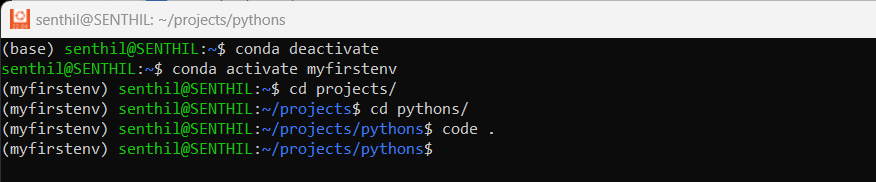


This will activate our own conda environment, we can check the environment as it will show up before our username. In my case it is (myfirstenv) senthil@SENTHIL.

Now we start our V S Code, before that make sure your V S code is connected to WSL

We can start vs code, using the terminal command.

* code .



Now, create a requirements.txt file and add below content to it.

# computation libraries

numpy==1.26.0

scipy==1.11.3

pandas==2.1.1

# plotting and visualization libraries

matplotlib==3.8.0

seaborn==0.13.0

And save the file



Now, open the terminal in V S Code by going to

View -> terminal

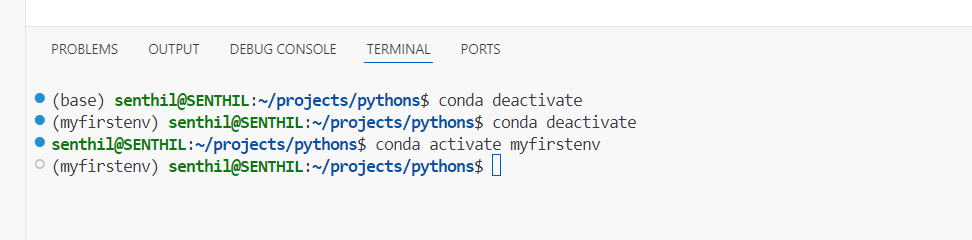
Once the terminal is opened deactivate the conda (base) environment

* conda deactivate

use twice or thrice till you exit from every environment

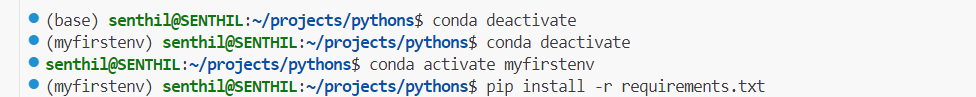
then activate our own environment.

* conda activate myfirstenv

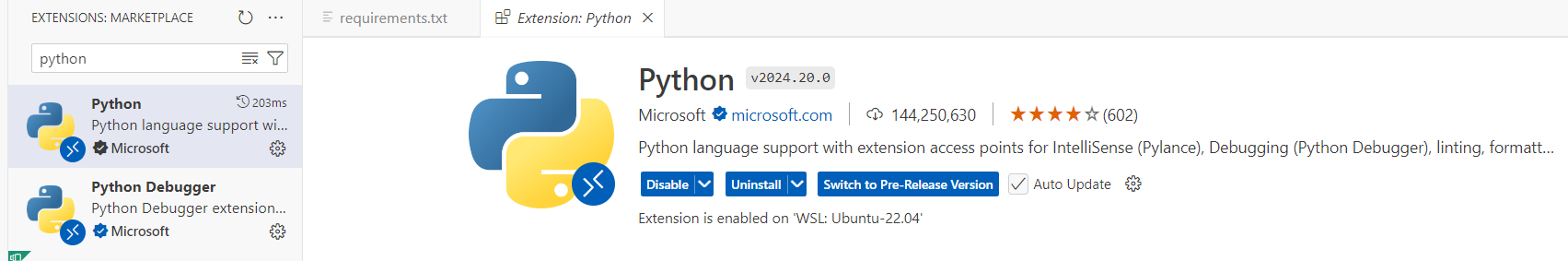


Now, install all the packages from the requirements.txt file

* pip install -r requirements.txt

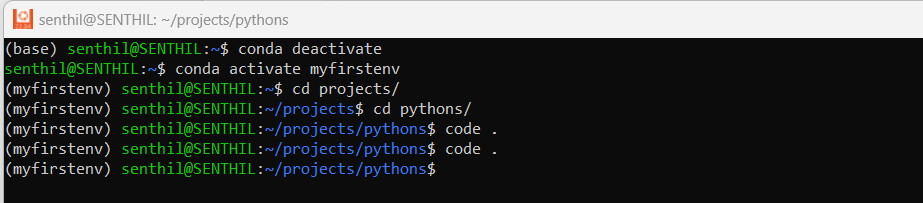


Make sure you have python extension installed in the VS code. If not, go to extension and search for python and install it.



Once done, close VS Code and restart it using ubuntu terminal using the same command.

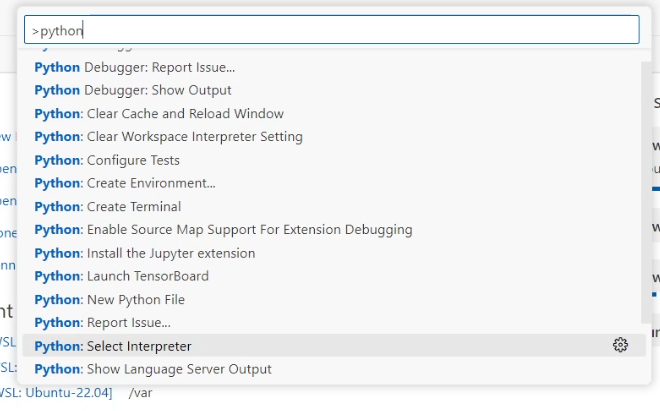
* code .



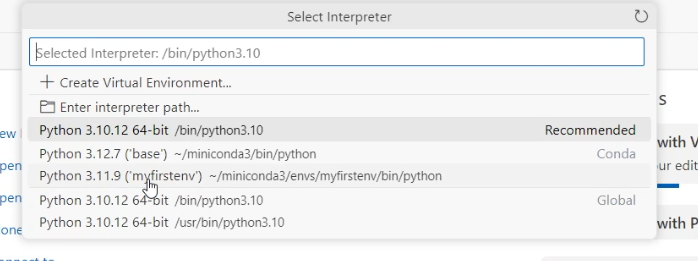
Once, VS Code opens up

Press Ctrl + shift + p

And choose python: select interpreter



And, then select python 3.11.9(‘myfirstenv’)



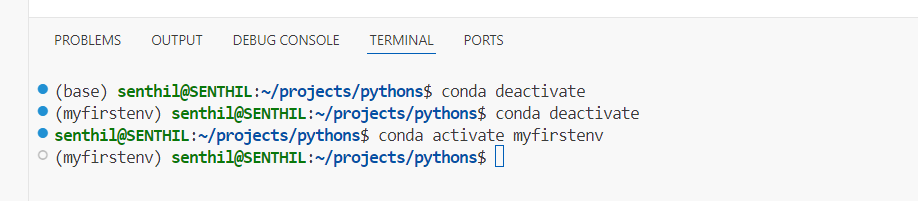
Get back to the terminal and go to our environment by repeating the same steps.

* conda deactivate

use twice or thrice till you exit from every environment

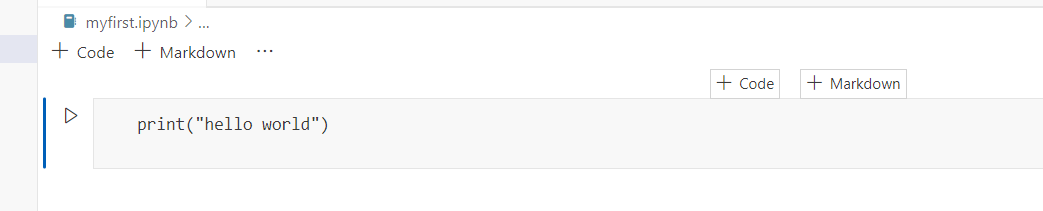
then activate our own environment.

* conda activate myfirstenv

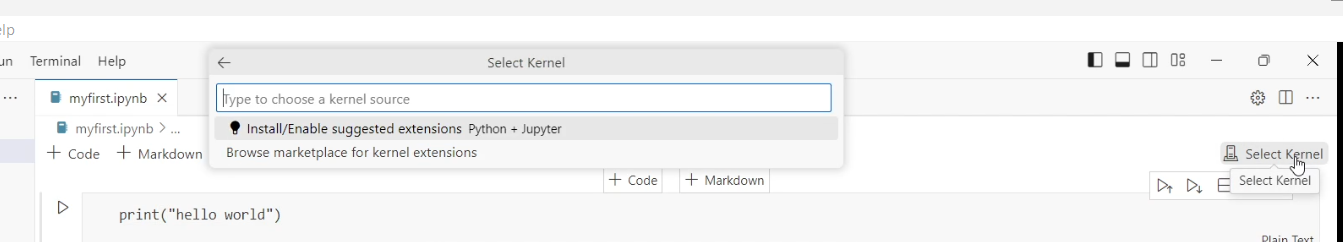


Create a new myfirst.ipynb file

And type print(“hello world”)



If it doesn’t work, go to the select kernel button.



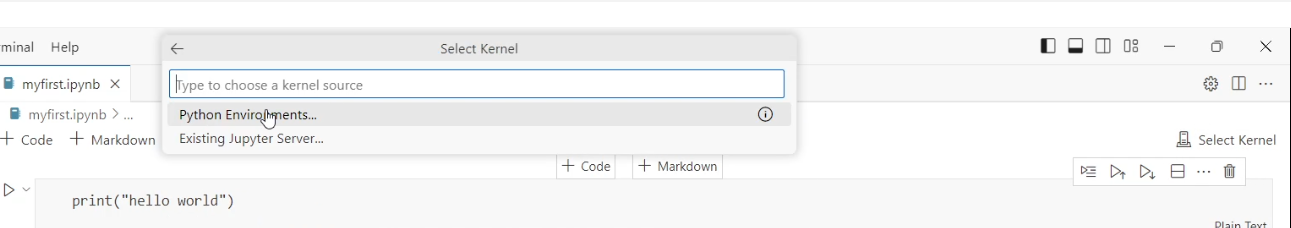
It will open the box, select install/enable suggested extension

It will install the jupyter notebook extension.

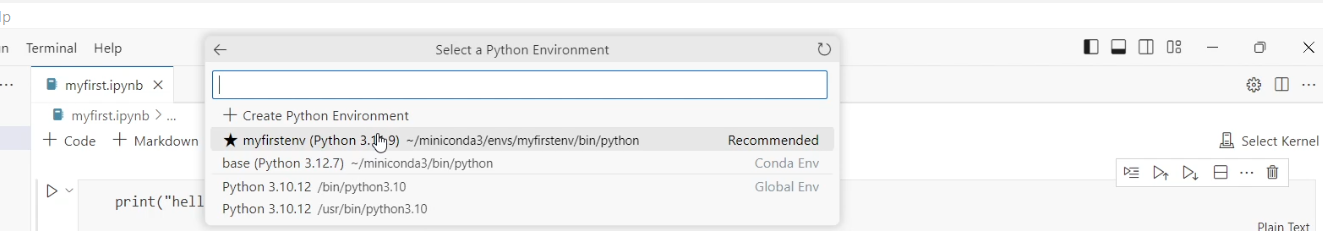
Once, jupyter notebook extension is installed

Then select python environments

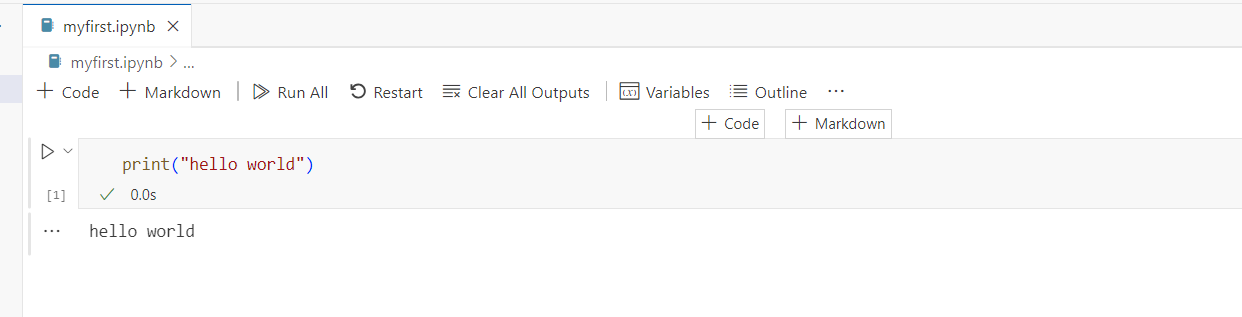
If it gets closed, then click on select kernel again it will again open up



Select myfirstenv(python3.11.9)



Now run the code, it should work

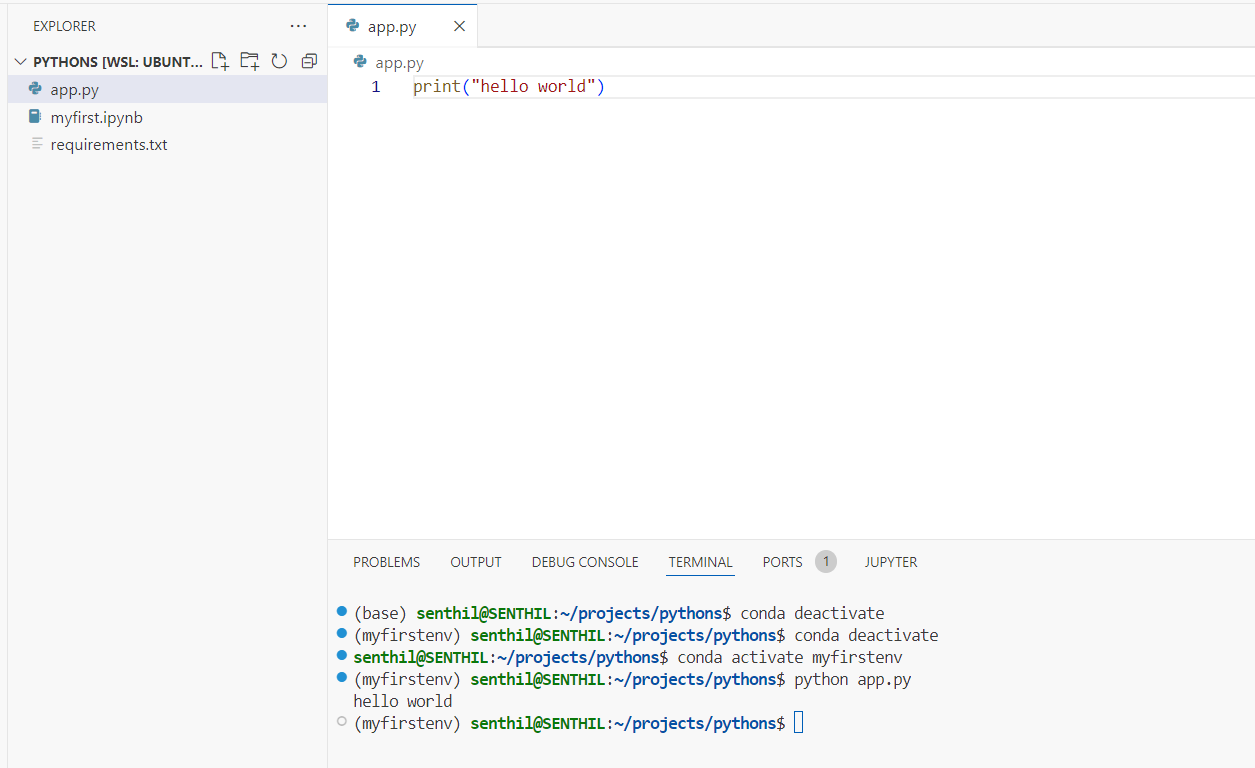


Create app.py file and type print(“hello world”)

Save it and run the using the terminal.

* python app.py

it should give you the output



Congratulations, you are done with building your custom Conda environment.