

# SETTING UP YOUR COMPUTER TO DOWNSCALE ECOSTRESS LST WITH PYDMS

## ECOSTRESS TUTORIALS

This tutorial will show you how to prepare and download everything for the good use of the downscaling algorithm on Windows.

Questions or comments: [quentindehaene@gmail.com](mailto:quentindehaene@gmail.com) or [glynn.hulley@jpl.nasa.gov](mailto:glynn.hulley@jpl.nasa.gov)

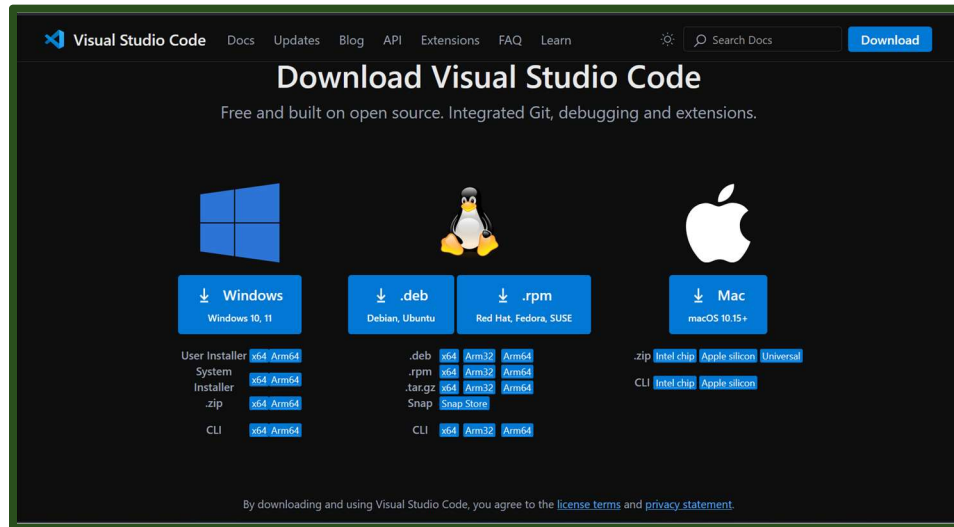
### Table of Contents

How to install Visual Studio Code .....	2
<b><i>What is Conda?</i> .....</b>	<b>4</b>
How to install Conda.....	4
<b><i>What is an Environment?</i>.....</b>	<b>6</b>
How to create an environment .....	6
How to prepare an environment for pyDMS .....	8
How to install and setup pyDMS .....	11

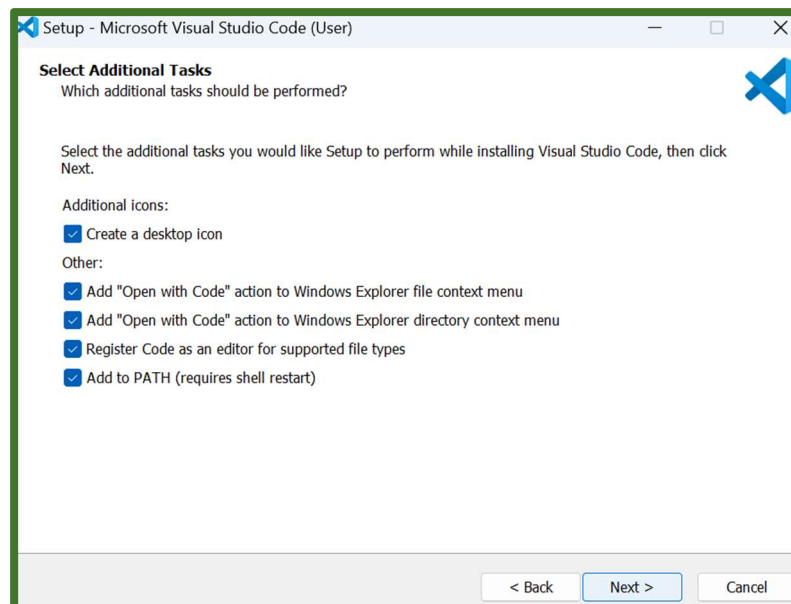


## HOW TO INSTALL VISUAL STUDIO CODE

Start by going to <https://code.visualstudio.com/download> or by searching the web for Visual Studio Code (VS Code). Click the appropriate icon for your OS. In our case, Windows.



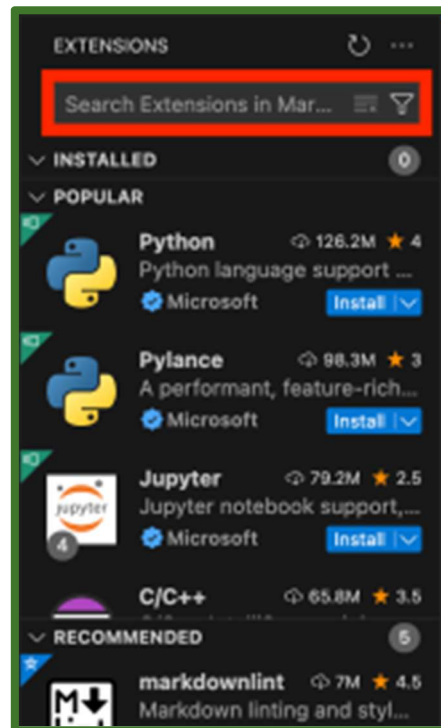
1. Now let's execute the newly downloaded VSCodeUserSetup-xxxxxx.exe file. Click next on the first pages, then agree to the license. Click next until you see this window. I advise you to check all these boxes (the first 3 are optional but they come in handy).



2. Open the newly installed application Visual studio Code.
3. Once the application is open, let's install some extensions to allow us to use Python and Jupyter notebooks in Visual Studio Code. On the left side of the application, look for the **extensions** icon and **click** on it.



4. A new **panel** should open prompting you to search for extensions.



3. In the search bar, type Python. Once you have found the extension, click the blue install button. It should say "installing" for a moment, and then it will be installed.




5. Do the same installation with Jupyter. You can also add other extensions such as formatters (Black formatter, Prettier) or the Python Debugger. There are plenty of resources online for extra extensions that make your coding experience smoother, an example [here](#).

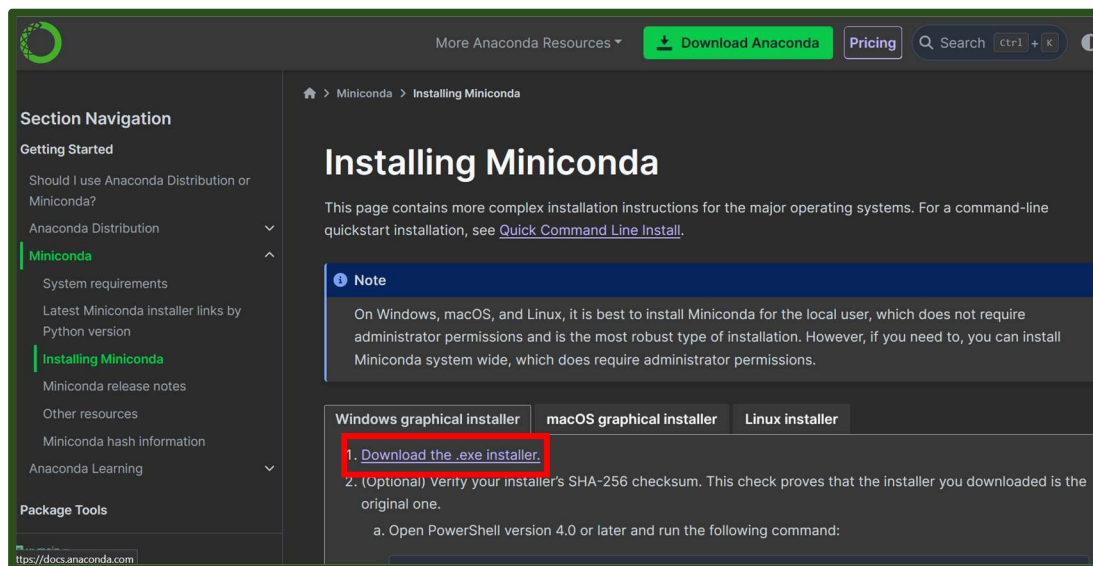
## What is Conda?

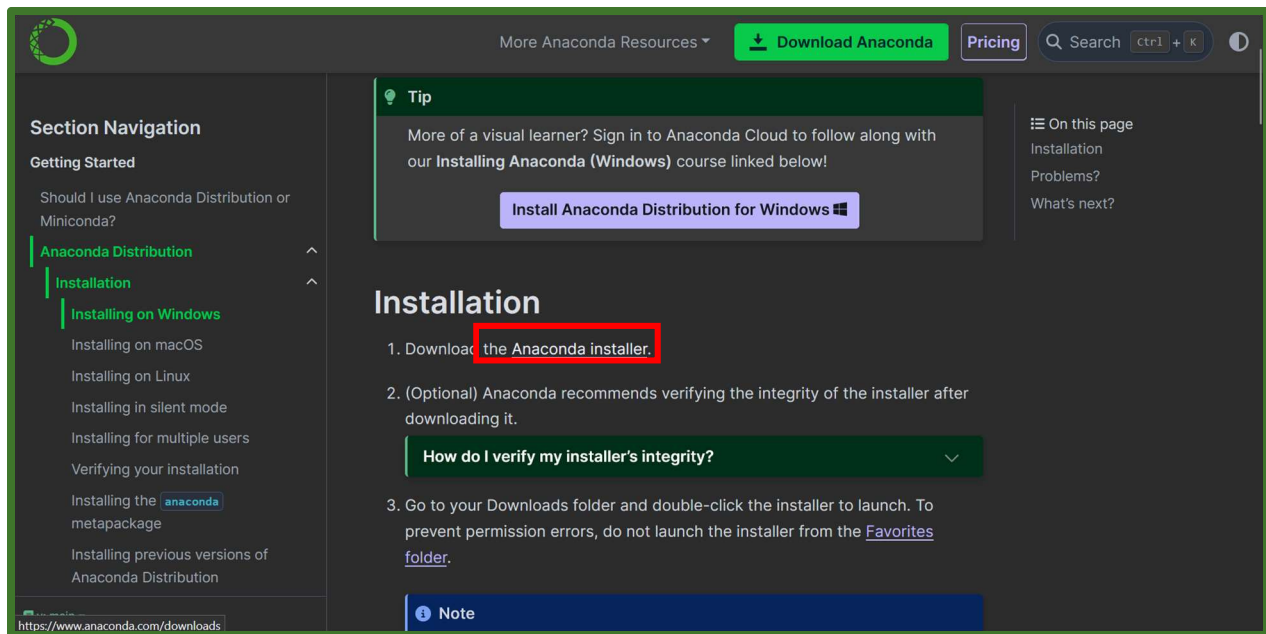
## HOW TO INSTALL CONDA

## HOW TO INSTALL CONDA

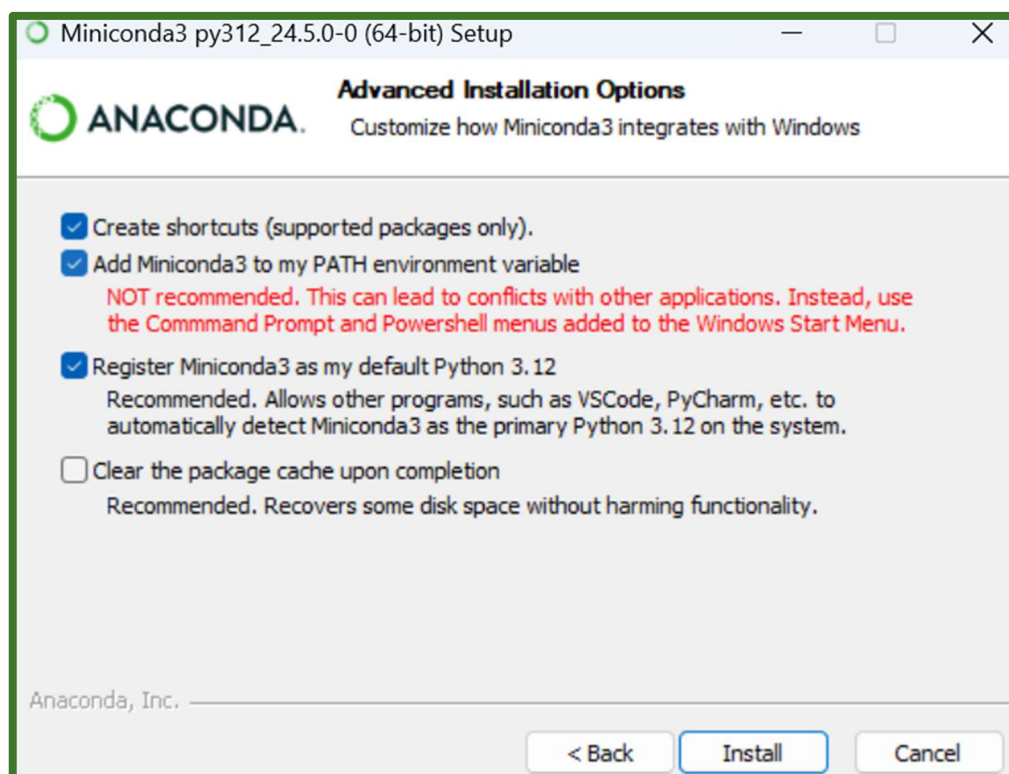
1. Go to <https://docs.anaconda.com/miniconda/miniconda-install/> and click on the “Download

- 
[More Anaconda Resources](#)
[Download Anaconda](#)
[Pricing](#)





2. Execute the downloaded installer and follow the process until you see this window. Here it is **very important** that you check the second box. If you don't VS Code won't recognize conda and you'll have to add it to PATH manually.



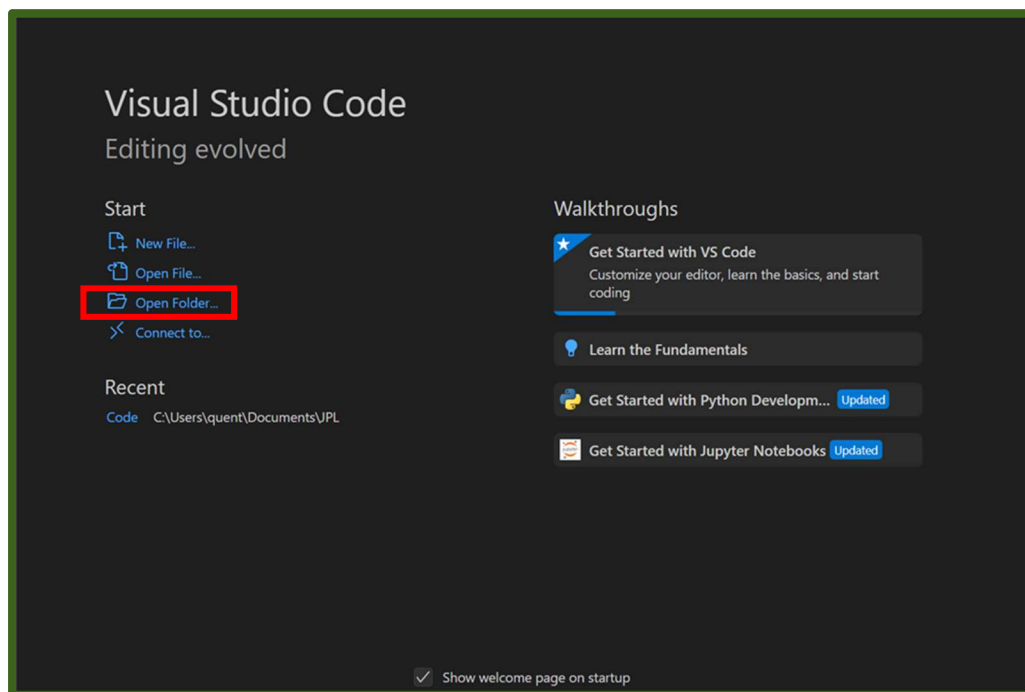
## What is an Environment?

An environment is a separate place on your computer where you can install software and libraries specific to the project you are working on. This allows you to have multiple projects all with their unique requirements. We need to create an environment that has all the tools we need to work with ECOSTRESS data.

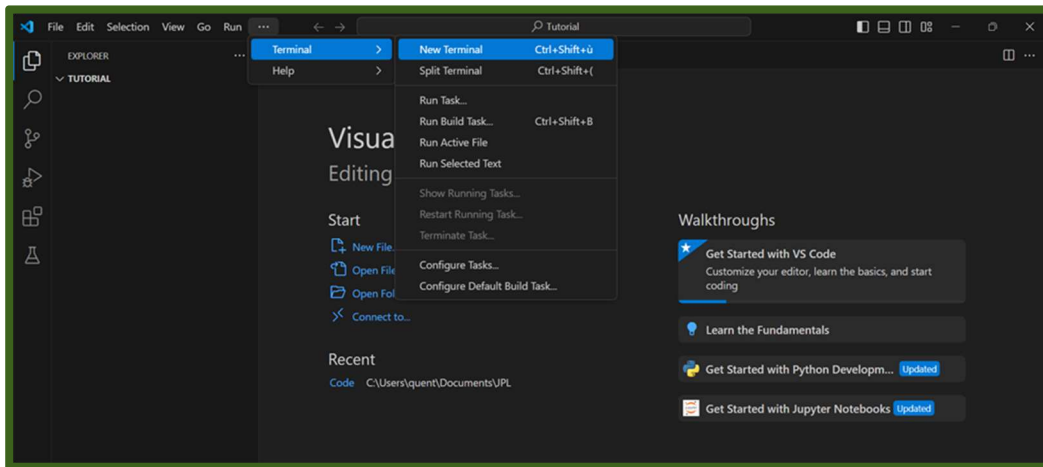
## HOW TO CREATE AN ENVIRONMENT

You might have create other environments in the past for other projects, there are many ways to create and set up you environment. I personally like to use this because it doesn't require to leave VS Code, but do what you are comfortable with.

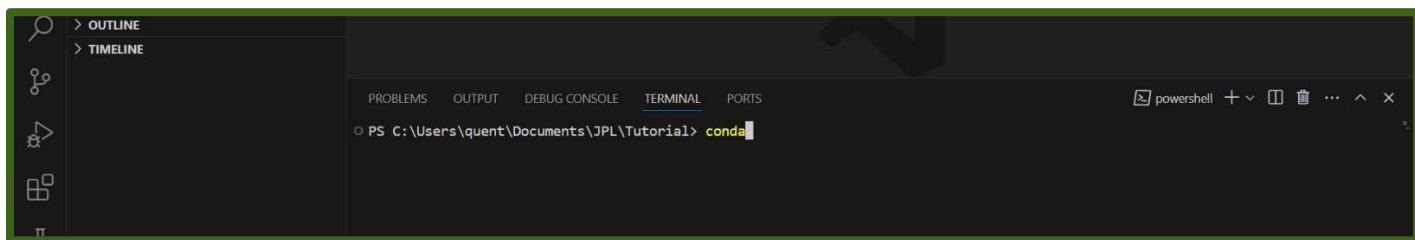
1. Open Visual Studio Code with the desktop icon if you created one while installing or search for the application in the Windows search bar. Click on Open Folder. Select the folder of your choosing, here I have previously created a folder especially for this tutorial.



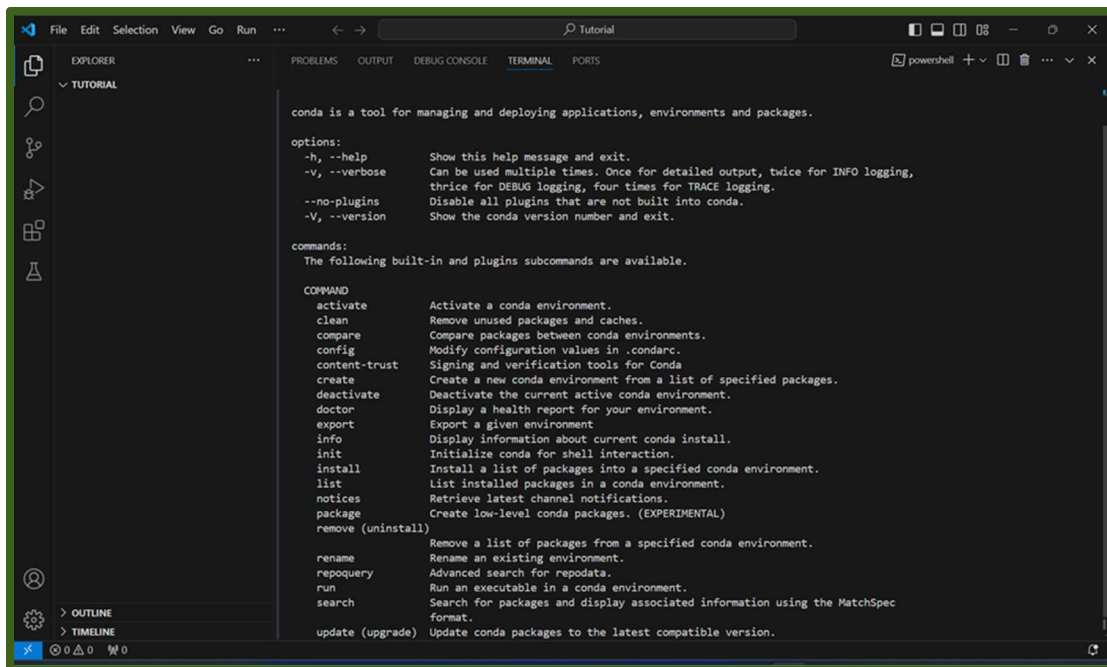
2. Open a new terminal in VS Code as such.



3. As a first test, type conda and run this command.

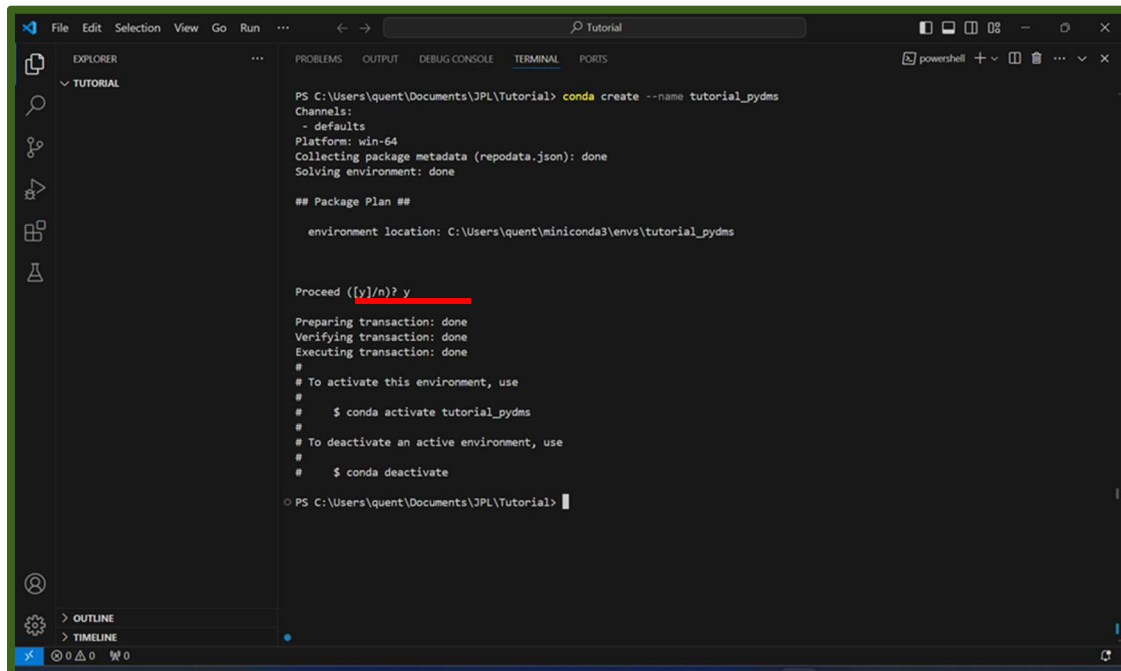


You should see:



If you've returned an error saying that the command is unknown, it's most likely because it wasn't added to PATH during the installation.

4. Once we checked that conda was properly installed. We can create an environment. To do so, run the command: **conda create --name nameofyourenvironment**  
You can add the option `python=3.x` where you precise the version of python you want to use for you environment. This can be helpful if you have different versions of python installed and that you don't want to confuse your environments.



```
PS C:\Users\quent\Documents\JPL\Tutorial> conda create --name tutorial_pydms
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\quent\miniconda3\envs\tutorial_pydms

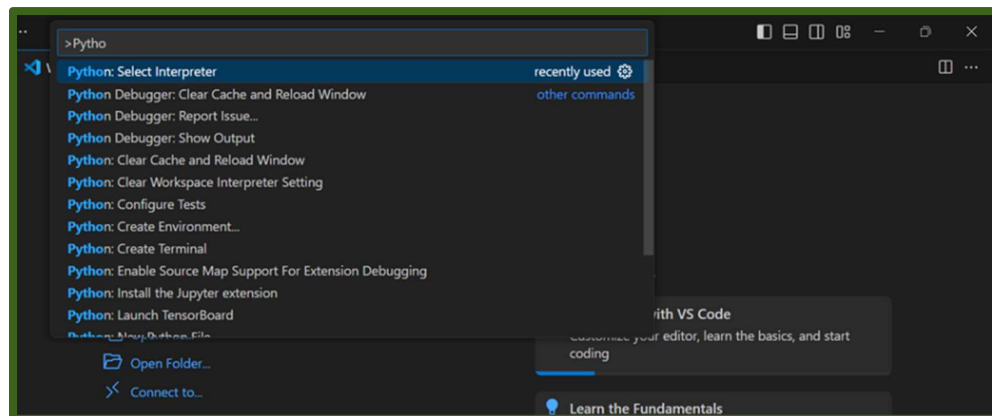
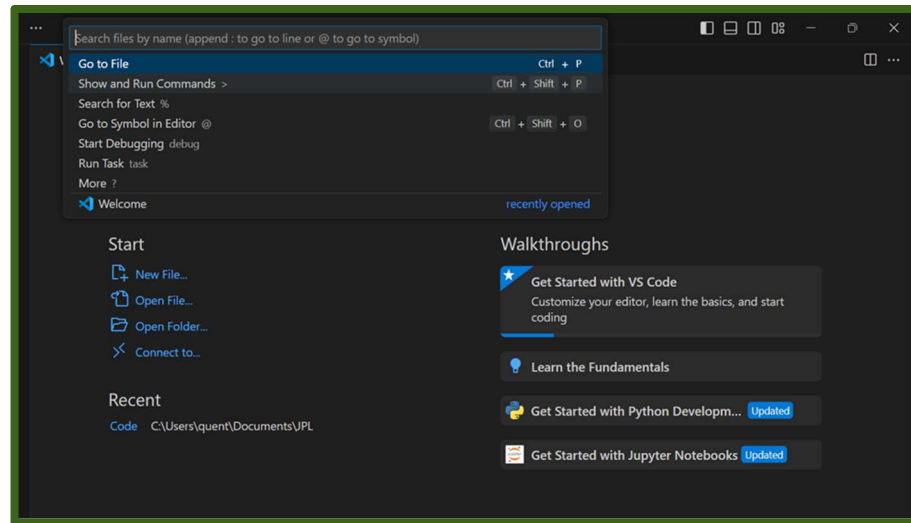
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate tutorial_pydms
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
PS C:\Users\quent\Documents\JPL\Tutorial>
```

Most of the time when you create, delete, install or uninstall, conda will ask you to confirm by typing y in the terminal as underlined here.

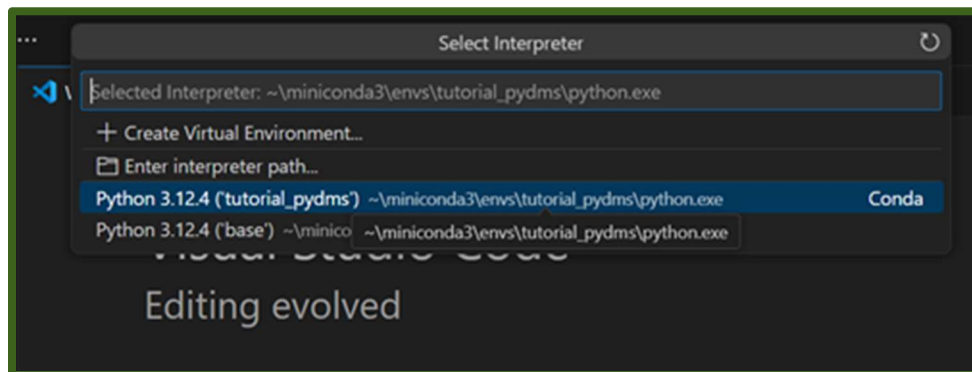
## HOW TO PREPARE AN ENVIRONMENT FOR PYDMS

1. Now, to use this newly created environment as our interpreter for our codes and commands. Click on the search bar on top of the window and type: **> Python: Select Interpreter.**

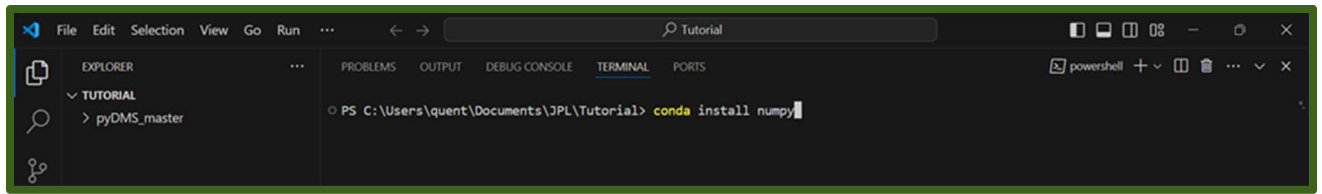




2. Select the newly created environment and open a new terminal.



3. It's now time to install the libraries needed to run the downscaling notebook. To install basic packages simply run: **conda install nameofthepackage**. For example, here with the package numpy.

A screenshot of a JupyterLab interface. The left sidebar shows the 'EXPLORER' view with a folder named 'TUTORIAL' containing a file 'pyDMS\_master'. The main area is the 'TERMINAL' tab, which shows a PowerShell prompt at 'PS C:\Users\quent\Documents\JPL\Tutorial>' with the command 'conda install numpy' entered.

Some packages are not available in the default channel so it requires an extra option to be able to install them: **conda install -c conda-forge nameofthepackage**. For instance, with the package gdal.

A screenshot of a JupyterLab terminal window. The terminal shows the output of a previous command: 'Preparing transaction: done', 'Verifying transaction: done', and 'Executing transaction: done'. Below this, a new PowerShell prompt shows the command 'conda install -c conda-forge gdal' being entered.

To be sure I advise you to run **conda install -c conda-forge nameofthepackage** that way you'll always find the package you are looking for.

To be able to run the notebook fully, you'll need to install the packages in requirements.txt. You can run the command "**conda list**" to read the list of the installed packages in your active environment.

Alternatively, you can replicate the environment I've been using on my computer to develop and run the code. For that I provided the file **sharpening\_ecostress\_dev.yml**, and to reproduce the environment run: `conda env create -f sharpening_ecostress_dev.yml`.

There is also the file **sharpening\_ecostress.yml** for a lighter configuration with only the packages needed.

For more information about conda environments visit:

<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

## HOW TO INSTALL AND SETUP PYDMS

1. Download the pyDMS\_master folder from the project, if you haven't already.
2. Move the pyDMS\_master folder to the folder open in VS Code. You should have the notebooks in the same parent folder as the pyDMS\_master folder.

Here is what the folder should contain:

Name	Type	Size
pyDMS	File folder	
.gitattributes	Git Attributes Source ...	1 KB
.gitignore	Git Ignore Source File	1 KB
README.md	Markdown Source File	3 KB
run_pyDMS.py	Python.File	3 KB
setup.py	Python.File	4 KB

3. Go back to your terminal (or open a new one) and change the directory to the newly extracted pyDMS\_master with the command: **cd .\pyDMS\_master\**

To be certain that you are using the correct environment run the command: **conda info**.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
powershell - pyDMS_master + - - - ^ x
PS C:\Users\quent\Documents\JPL\Tutorial> cd .\pyDMS_master\
PS C:\Users\quent\Documents\JPL\Tutorial\pyDMS_master> conda info
```

Check in the response that the active environment is the one where you installed all the packages.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - pyDMS_master + v [ ] [ ] ... ^ x
PS C:\Users\quent\Documents\JPL\Tutorial\pyDMS_master> conda info

      active environment : tutorial_pydms
      active env location : C:\Users\quent\miniconda3\envs\tutorial_pydms
            shell level : 1
        user config file : C:\Users\quent\.condarc
 populated config files :
          conda version : 24.7.1
    conda-build version : not installed
         python version : 3.12.4.final.0
           solver type : libmamba (default)
 virtual packages : __archspec=1=skylake
                   __conda=24.7.1=0
                   __win=0=0

 base environment : C:\Users\quent\miniconda3 (writable)
   conda av data dir : C:\Users\quent\miniconda3\etc\conda
conda av metadata url : None
   channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                  https://repo.anaconda.com/pkgs/main/noarch
                  https://repo.anaconda.com/pkgs/r/win-64
                  https://repo.anaconda.com/pkgs/r/noarch
                  https://repo.anaconda.com/pkgs/msys2/win-64
                  https://repo.anaconda.com/pkgs/msys2/noarch

 package cache : C:\Users\quent\miniconda3\pkgs
                  C:\Users\quent\.conda\pkgs
                  C:\Users\quent\AppData\Local\conda\conda\pkgs
   envs directories : C:\Users\quent\miniconda3\envs
                     C:\Users\quent\.conda\envs
                     C:\Users\quent\AppData\Local\conda\conda\envs
 platform : win-64
```

4. Finally, you can run this last command: **python setup.py install**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - pyDMS_master + v [ ] [ ] ... ^ x
PS C:\Users\quent\Documents\JPL\Tutorial\pyDMS_master> python setup.py install
```

You are now ready to use the downscaling notebooks to upsample ECOSTRESS LST from 70m to Sentinel 2 resolution. Follow the instructions in the notebook, read the comments for explanations and use the document listing some frequents errors if you encounter an issue.