

Initial setup: these instructions are for getting a version of the linux operating system working on your Windows machine. This allows you to use the same linux commands (e.g. for navigation and making directories) as you would on a remote linux machine. These also show you how to install the Anaconda distribution of python in that linux environment, and make plots appear.

Thanks to Amanda Manaster, Alexander Islas and other students from the 2020 Effective computing class for putting this together.

NOTE: You can also have a separate installation of Anaconda python that works directly in your Windows operating system. Just get the most recent graphical installer for Windows from here: <https://www.anaconda.com/> and follow their directions.

1) Activate WSL

- a) Navigate to: Control Panel>Programs>Programs and Features>Turn Windows features on and off
- b) Check: Windows Subsystem for Linux
- c) Click OK and let system reboot

2) Getting Linux

- a) Go to Microsoft Store and search for Ubuntu
- b) Download Ubuntu for Windows
- c) Open Ubuntu and create a user account and password
 - i) Note: from here, using bash in the windows Command Prompt gets you to the same Linux style prompt.
- d) In Ubuntu, enter: `sudo apt-get update && sudo apt-get upgrade`

3) Install Python in Ubuntu:

- a) Find the latest version of Anaconda for Linux at: <https://repo.anaconda.com/archive/> in any browser.
 - i) Find the version with "Linux-x86_64.sh" at the end.
 - ii) Find the version number as "Anaconda3-{version_number}-Linux"
 - iii) Use that version number in step 3.b and 3.c without the curly brackets.
- b) Enter: `wget https://repo.anaconda.com/archive/Anaconda3-{version_number}-Linux-x86_64.sh`
- c) Enter: `bash Anaconda3-{version_number}-Linux-x86_64.sh`, and wait.
- d) When prompted, select 'yes' to add to PATH, selection defaults to 'no'
- e) Enter: `conda list`, if this shows an error about 'command not found', conda was not added to path, continue, else skip to step 4.
- f) Enter: `nano ~/.bashrc`
- g) At the bottom of the file add: `export PATH=~/.anaconda3/bin:$PATH`, save and close file. Using Ctrl + o and then ctrl + x
- h) Enter: `source ~/.bashrc`
- i) Enter: `conda list`, now you should end up with a list of available modules

4) Update Python:

- a) Enter: `conda update --all`

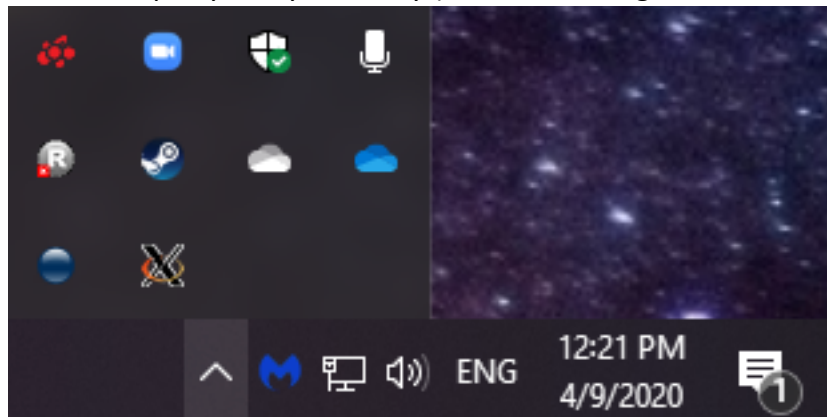
- b) Pip install
 - i) Try: `pip install netCDF4 seawater gsw`
 - ii) For the modules that give an error (often gsw), continue to 4.c
- c) Conda install
 - i) `conda install [package_name]`, without brackets.
 - ii) For the modules that give an error, continue to 4.d
- d) Conda install through channels
 - i) `conda install -c conda-forge [package_name]`, without brackets.

5) Verify that Python works in Ubuntu:

- a) Enter: `ipython --pylab`
- b) Enter:
 - `import matplotlib.pyplot as plt`
 - `import numpy as np`
 - `x = np.linspace(0,10,1000)`
 - `y = np.sin(x**2)`
 - `plt.plot(x,y)`
- c) If the plot doesn't show, continue to step 6 or within the same ipython enter:
 - `plt.savefig("[filepath]/[filename].png")`

6) To display graphics from Ubuntu:

- a) Download and install Xming X Server for Windows
 - i) <https://sourceforge.net/projects/xming/>
 - ii) Keep the default settings
- b) Launch Xming
 - i) It should show up in your system tray (the X icon being visible means it's running)



- c) Tell Ubuntu where to display the graphics
 - i) `export DISPLAY=localhost:0.0`
- d) Add the previous command to `~/.bashrc` to make it permanent
 - i) `nano ~/.bashrc`
 - ii) Add `export DISPLAY=localhost:0.0` to the end of the file and save it (Ctrl+O to write to the file, hit 'enter', then Ctrl-X to exit)
- e) Run `source ~/.bashrc`
- f) Try Step 5 again
- g) If step 5 still doesn't show the plot, check your backend in ipython

- i) Enter: `import matplotlib` and then `matplotlib.get_backend()`
 - ii) If the backend is `agg`, go to the next step.
 - h) Install libraries to get the correct backend (for Ubuntu, we want our backend to be `TkAgg`).
 - i) `sudo apt-get install tcl-dev tk-dev python-tk python3-tk`
- 7) Try Step 5 again, but enter `ipython --pylab=tk` to set the backend.**