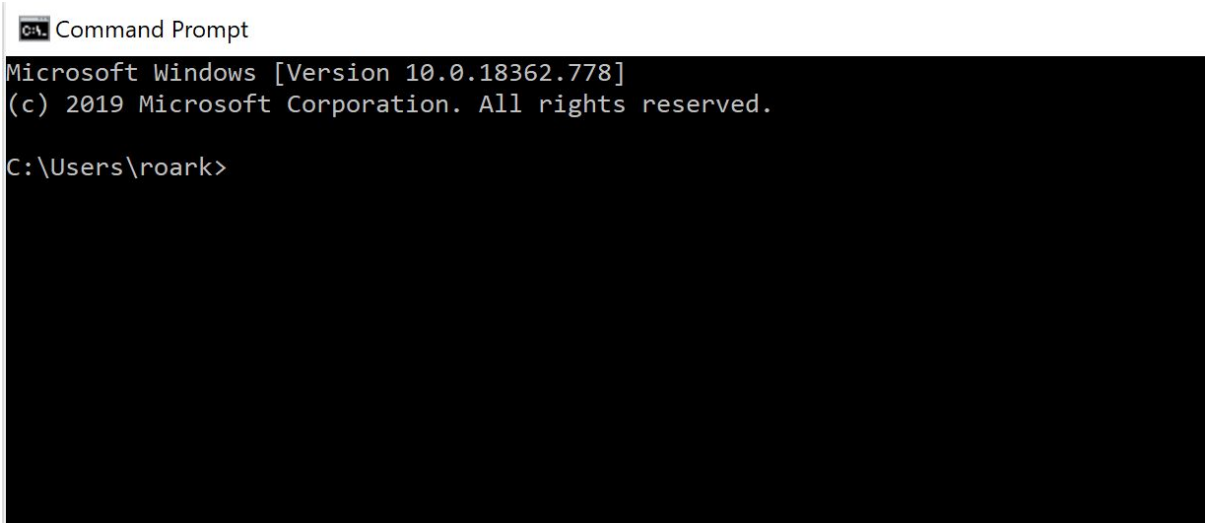
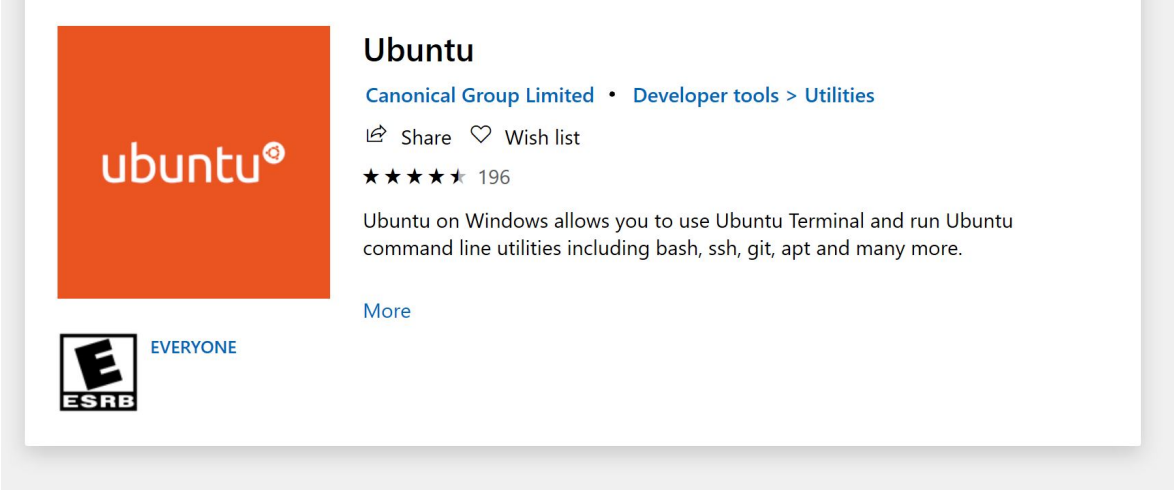
Windows Operating System

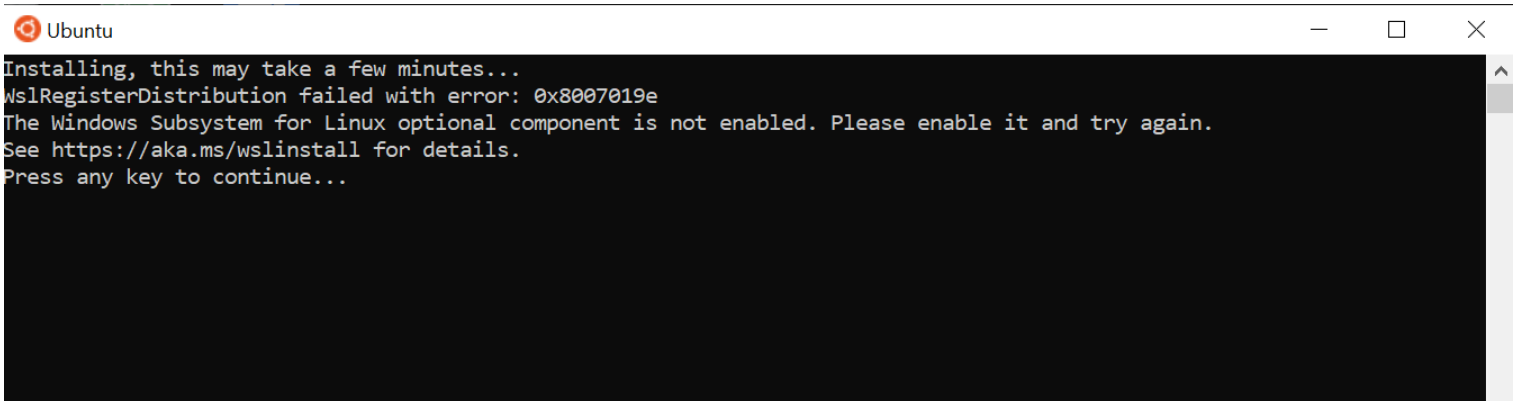
What a 'Windows operating system' actually means is if you type Terminal into your device's search bar, you get the following program with your username (instead of roark):



While I have a windows computer, UNIX based computers like Macs have a better time with python. Also, their terminals are easier to navigate. Therefore one of the best ways to learn terminal work and python is by downloading a 'vitrual' UNIX terminal. Ubuntu is an example of this and it is available on the Microsoft store as an app:



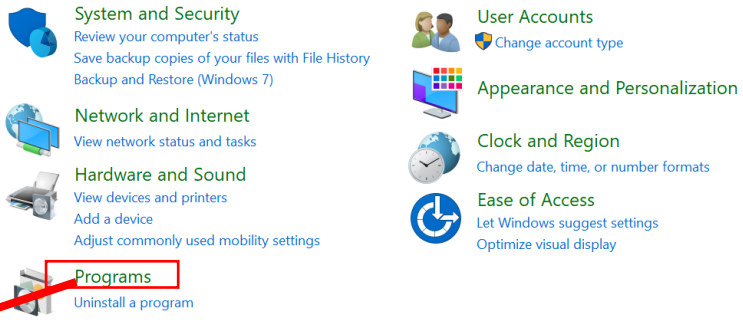
Once you've downloaded the app, try to run it. Unless you already have a Linux subsystem on your computer, you'll see this:

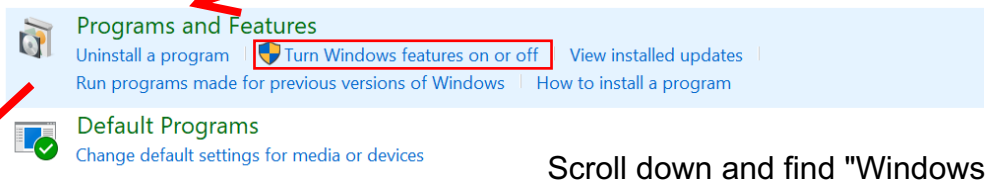


If you do not get this error, feel free to skip the next section on enabling a Linux subsystem on your computer.

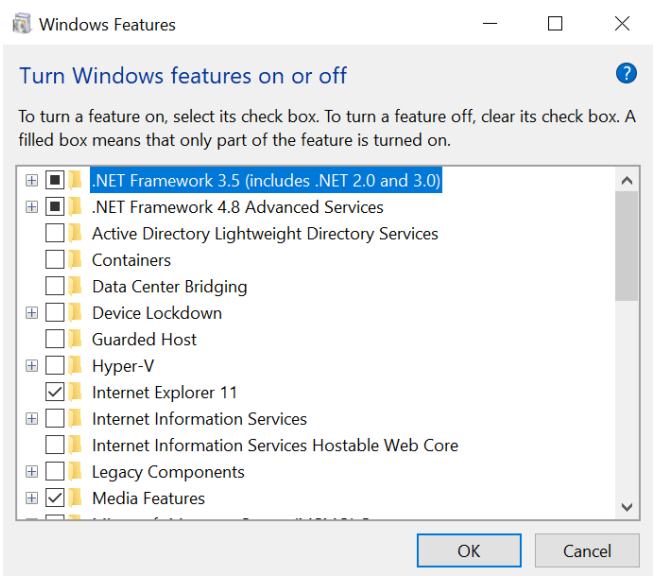
Enabling Linux Subsystem for Windows

Go to your control panel in the Start menu and click on the Programs subheading. Then within that menu, click on "Turn Windows Features On or Off" which will most likely have a shield next to it. You then might need to allow changes to your system's hard drive (a Pop-Up window will appear if that happens)

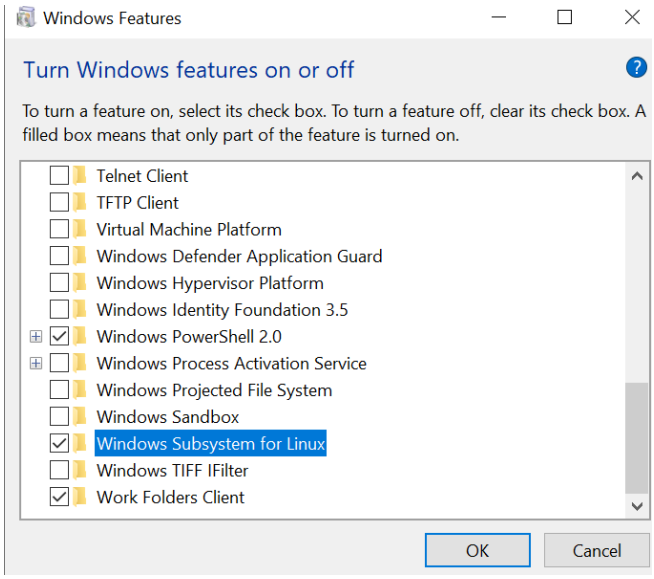




A new window should pop up that looks like the following:

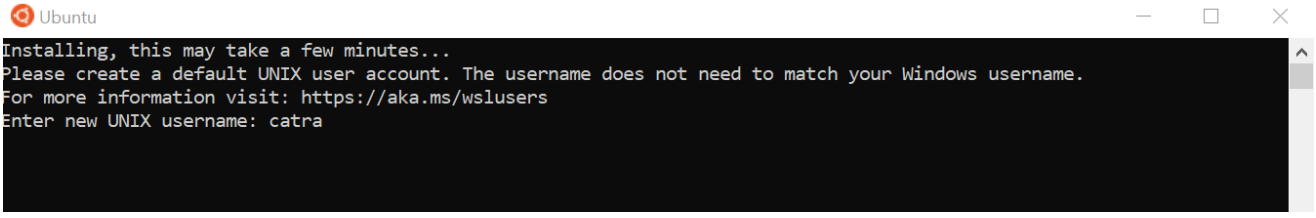


Scroll down and find "Windows Subsystem for Linux" and click on the check box. Then click Ok.



Then you will need to restart your computer. There should be a pop up window that suggests you restart your computer, and offers you an option to. Go ahead and restart your computer.

Assuming you have the subsystem for Linux enabled, then go ahead and open Ubuntu. You should get a prompt in the Linux terminal asking you for a username. Choose wisely! It can be your Windows username but regardless of what you use it can be difficult to change. Stick to just letters and numbers.



Next you'll have to type a password. Remember it! Write it down somewhere, like the notes on your phone. Make it simple. The weird part of this terminal interface is that when you type a password the letters will not show up and the cursor (flashing white underscore) will not move. So be careful that you don't type an extra letter or symbol. Hit enter when you finish the password, and then retype the exact same password. After this, you should get the following:

```
Enter new UNIX username: catra
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 4.4.0-18362-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat May 16 08:54:59 EDT 2020

  System load:    0.52      Processes:              7
  Usage of /home: unknown   Users logged in:        0
  Memory usage:   26%       IPv4 address for wifi0: 192.168.1.19
  Swap usage:     0%

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

Then you should get a basic terminal entry with a flashing cursor where you can type commands. First type "cd /" without the quotations. The "cd" stands for 'change directory'. This will take you to the base directory for the subsystem. To get to your normal PC's files, type (without the quotations) "cd mnt/c/Users/my_name/" where my_name is your Windows username (you can find this in your file explorer by going to Local Disk (C:) and then Users).

After this, type ls or la (la is a shorthand command for ls -a). ls stands for list and the first letter is a lower case L. The -a is an optional parameter for the command that means 'all' as in list all the files and links and folders in your current location.

```
catra@catra-pc:~$ cd /
catra@catra-pc:/$ cd mnt/c/Users/catra/
catra@catra-pc:/mnt/c/Users/catra$ la
```

Setting up Python!

In order to set up Ubuntu for Python coding, we need to update the software.

Type "sudo apt update" without quotations. The terminal will ask for your password that you set when you opened Ubuntu for the first time. If you type the password correctly, Ubuntu will begin downloading the most recent versions of the software it contains.

Then, once there command prompt appears again, run "sudo apt upgrade" which will take all the updates and actually apply them to the software. First it will ask if you want to continue with the upgrade. Type "y" and hit enter (this is shorthand for yes). The upgrade will take some time and will look like the image on the left while the files are edited.

```
(Reading database ... 31836 files and directories currently installed.)
Preparing to unpack .../libapt-pkg6.0_2.0.2ubuntu0.1_amd64.deb ...
Unpacking libapt-pkg6.0:amd64 (2.0.2ubuntu0.1) over (2.0.2) ...
Setting up libapt-pkg6.0:amd64 (2.0.2ubuntu0.1) ...
(Reading database ... 31836 files and directories currently installed.)
Preparing to unpack .../apt_2.0.2ubuntu0.1_amd64.deb ...
Unpacking apt (2.0.2ubuntu0.1) over (2.0.2) ...
Setting up apt (2.0.2ubuntu0.1) ...
(Reading database ... 31836 files and directories currently installed.)
Preparing to unpack .../00-apt-utils_2.0.2ubuntu0.1_amd64.deb ...
Unpacking apt-utils (2.0.2ubuntu0.1) over (2.0.2) ...
Preparing to unpack .../01-python3.8_3.8.2-1ubuntu1.1_amd64.deb ...
Unpacking python3.8 (3.8.2-1ubuntu1.1) over (3.8.2-1ubuntu1) ...
Preparing to unpack .../02-libpython3.8_3.8.2-1ubuntu1.1_amd64.deb ...
Unpacking libpython3.8:amd64 (3.8.2-1ubuntu1.1) over (3.8.2-1ubuntu1) ...
Preparing to unpack .../03-libpython3.8-stdlib_3.8.2-1ubuntu1.1_amd64.deb ...
Unpacking libpython3.8-stdlib:amd64 (3.8.2-1ubuntu1.1) over (3.8.2-1ubuntu1) ...
Preparing to unpack .../04-python3.8-minimal_3.8.2-1ubuntu1.1_amd64.deb ...
Unpacking python3.8-minimal (3.8.2-1ubuntu1.1) over (3.8.2-1ubuntu1) ...
Preparing to unpack .../05-libpython3.8-minimal_3.8.2-1ubuntu1.1_amd64.deb ...
Unpacking libpython3.8-minimal:amd64 (3.8.2-1ubuntu1.1) over (3.8.2-1ubuntu1) ...
Preparing to unpack .../06-distro-info-data_0.43ubuntu1.1_all.deb ...
Unpacking distro-info-data (0.43ubuntu1.1) over (0.43ubuntu1) ...
Preparing to unpack .../07-libjson-c4_0.13.1+dfsg-7ubuntu0.2_amd64.deb ...
Unpacking libjson-c4:amd64 (0.13.1+dfsg-7ubuntu0.2) over (0.13.1+dfsg-7) ...
Preparing to unpack .../08-libnetplan0_0.99-0ubuntu3~20.04.1_amd64.deb ...
Unpacking libnetplan0:amd64 (0.99-0ubuntu3~20.04.1) over (0.99-0ubuntu1) ...
Progress: [ 23%] [#####################.....................................................................]
```

```
catra@catra-pc:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils distro-info-data glib-networking glib-networking-common glib-networking-services l
  libapt-pkg6.0 libjson-c4 libldap-2.4-2 libldap-common libnetplan0 libpulse0 libpulsedsp libpytho
  libpython3.8-minimal libpython3.8-stdlib netplan.io pulseaudio-utils python3-distupgrade python3
  python3-update-manager python3.8 python3.8-minimal sosreport strace ubuntu-release-upgrader-core
28 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 10.2 MB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

After typing y and hitting enter, the terminal should begin to update itself.

Once the upgrade completes, you should be able to type "python3" and get a Python shell! We will experiment with it later. For now, type "exit()" without the quotations into the python shell and you will return to the normal Terminal command prompt (note that your Python version might be different; it must start with a 3 though, i.e. Python 3.#.#).

```
catra@catra-pc:/mnt/c/Users/catra$ python3
Python 3.8.2 (default, Mar 13 2020, 10:14:16)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Once you've returned to the terminal's command prompt, we need to download the python Pip library. This will allow you to download a variety of useful libraries in a simple way. The command is "sudo apt install python3-pip" and will work similar to the "upgrade" command we just used!

```
catra@catra-pc:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
  gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-9-dev libgomp1 libisl22 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.8-dev
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-wheel python3.8-dev zlib1g-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf
  automake libtool flex bison gdb gcc-doc gcc-9-multilib glibc-doc bzr libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
  gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-9-dev libgomp1 libisl22 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.8-dev
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-pip python3-wheel python3.8-dev zlib1g-dev
0 upgraded, 50 newly installed, 0 to remove and 0 not upgraded.
Need to get 46.7 MB of archives.
After this operation, 200 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

To completely set up python, we are going to install some important and useful libraries. The first is NumPy, the second is MatPlotLib, and the third is AstroPy. More on the uses of these libraries later. For now let's just install them using Pip.

For each library we need to type "python3 -m pip install LIBRARY" without quotations. LIBRARY will need to be replaced by "numpy", "matplotlib", and "astropy" in succession.

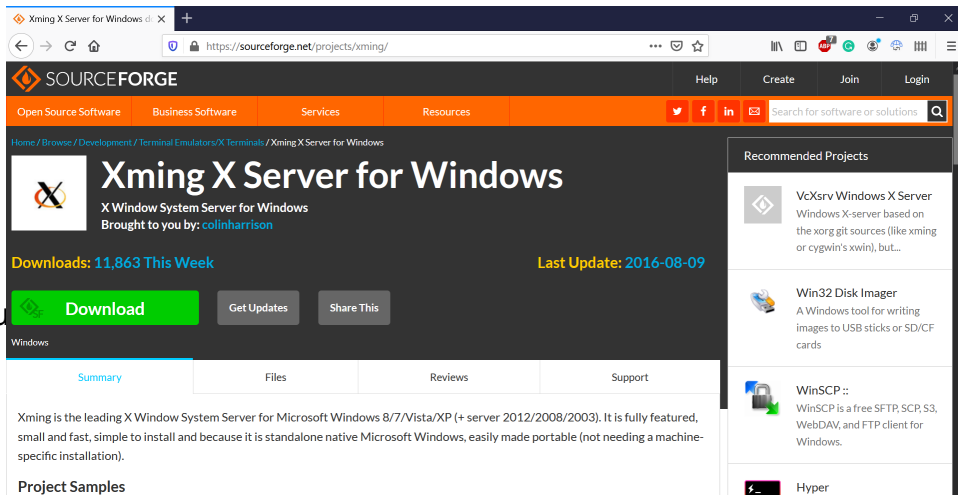
```
catra@catra-pc:~$ python3 -m pip install numpy
Collecting numpy
  Downloading numpy-1.18.4-cp38-cp38-manylinux1_x86_64.whl (20.7 MB)
     |████████████████████████████████| 20.7 MB 725 kB/s
Installing collected packages: numpy
  WARNING: The scripts f2py, f2py3 and f2py3.8 are installed in '/home/catra/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.18.4
catra@catra-pc:~$
catra@catra-pc:~$ python3 -m pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.2.1-cp38-cp38-manylinux1_x86_64.whl (12.4 MB)
     |████████████████████████████████| 12.4 MB 971 kB/s
Collecting python-dateutil>=2.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB 6.5 MB/s
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     |████████████████████████████████| 67 kB 4.9 MB/s
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Requirement already satisfied: numpy>=1.11 in ./.local/lib/python3.8/site-packages (from matplotlib) (1.18.4)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.2.0-cp38-cp38-manylinux1_x86_64.whl (92 kB)
     |████████████████████████████████| 92 kB 114 kB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.1->matplotlib) (1.14.
0)
Installing collected packages: python-dateutil, pyparsing, cycler, kiwisolver, matplotlib
```

```
catra@catra-pc:~$ python3 -m pip install astropy
Collecting astropy
  Downloading astropy-4.0.1.post1-cp38-cp38-manylinux1_x86_64.whl (6.5 MB)
     |                                | 6.4 MB 6.5 MB/s eta 0:00:01
```

Now that we have those libraries, our python setup is complete for most physics applications! There still is one implementation problem: MatPlotLib is a plotting library. It creates graphs, figures, and diagrams. For it to show its work, we need a plotting software installed on Windows. That way we will be able to see the cool plots we produce with python and the MatPlotLib Library!

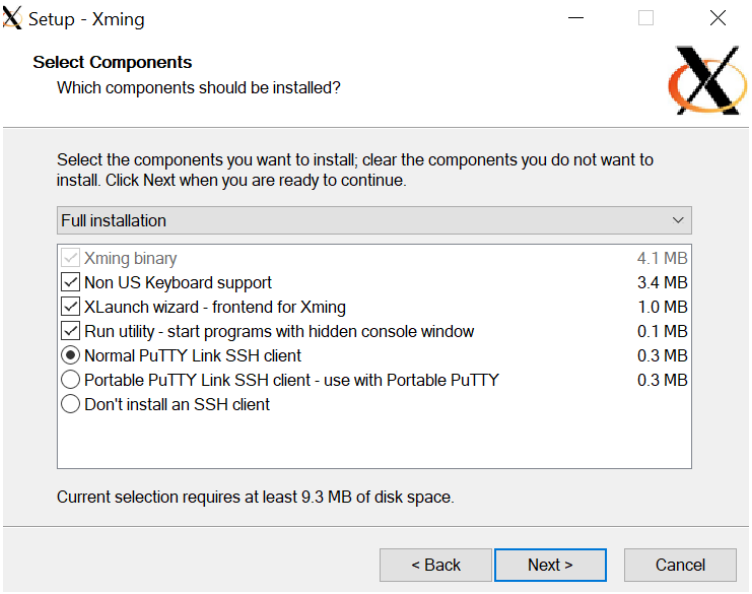
Installing Xming Plotting Software

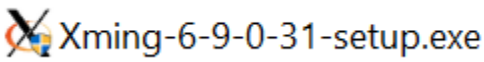
The plotting program I recommend is Xming. You can download it at https://sourceforge.net/projects/xming/ which looks like the figure to the right.



Just press the green download button and save the file. It should be a .exe file type, and once you've downloaded it you need to run it to set up the server.

Follow the setup Wizard and select the full installation, with the following components





The final step is to register the Xming server in the Ubuntu Terminal. This is done by typing the command "export DISPLAY=localhost:0.0" into the Ubuntu terminal. Then close Ubuntu and restart your computer. You're all set now!

```
catra@catra-pc:~$ export DISPLAY=localhost:0.0
```

NOTE: Xming is a "behind-the-scenes" program and you won't see an application for it unless you plot something in Python. Instead, you can check if it is running/active by checking the bottom left hand corner of your screen. If the small Xming logo is there, you are all set! If not, just start the app from your start menu.