

AggieSTAAR

Python

Bootcamp

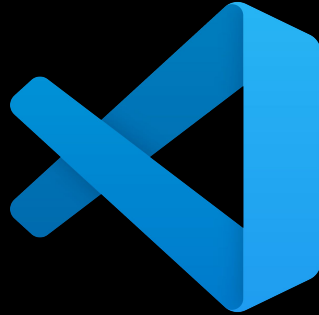
Tutorial 0:
Setting up Python



TEXAS A&M
UNIVERSITY®



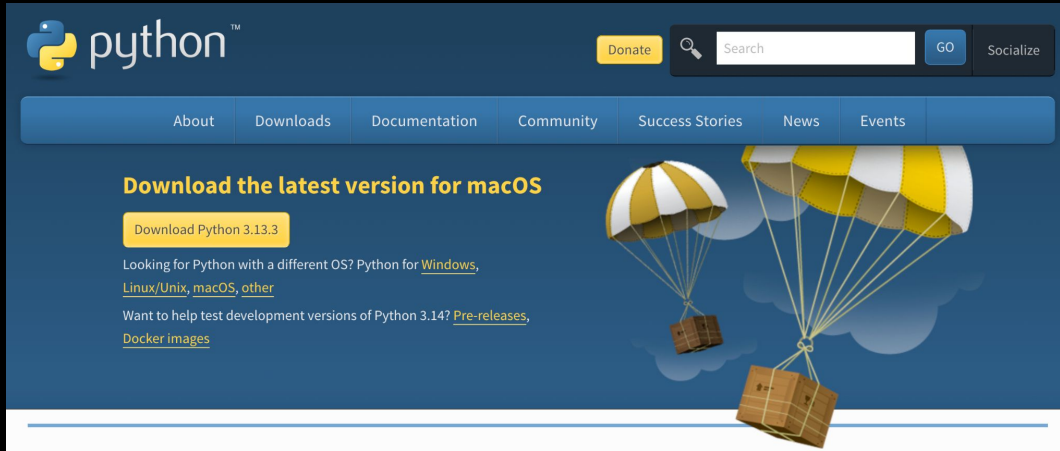
Python is your best friend!



It is the most frequently used coding language in astronomy today. Fortunately, it's (relatively) straightforward to learn.

We will learn how to use Python through notebooks, which are user-friendly platforms that allow you to interact with code.

(1): Installing Python



1. Download Python from python.org/downloads
 - a. Make sure you get the correct version for your operating system (MacOS, Windows, Linux).
2. Download your notebook program of choice.
 - a. I use [VSCode](#), although [Jupyter](#) is more widely used.
 - b. I will work through this tutorial with [Jupyter](#).

(2): Installing Jupyter

1. MacOS users: open your `terminal`, and type in `pip install jupyterlab`.
 - a. `pip` needs to be installed (should come automatically with `Python`). Let me know if this is giving you trouble.
2. Windows users: download and install `Anaconda`.
 - a. This method will also work on MacOS, if `pip` is giving you trouble.
 - b. `pip` may work on the `Windows command line`, but I have not tested it.



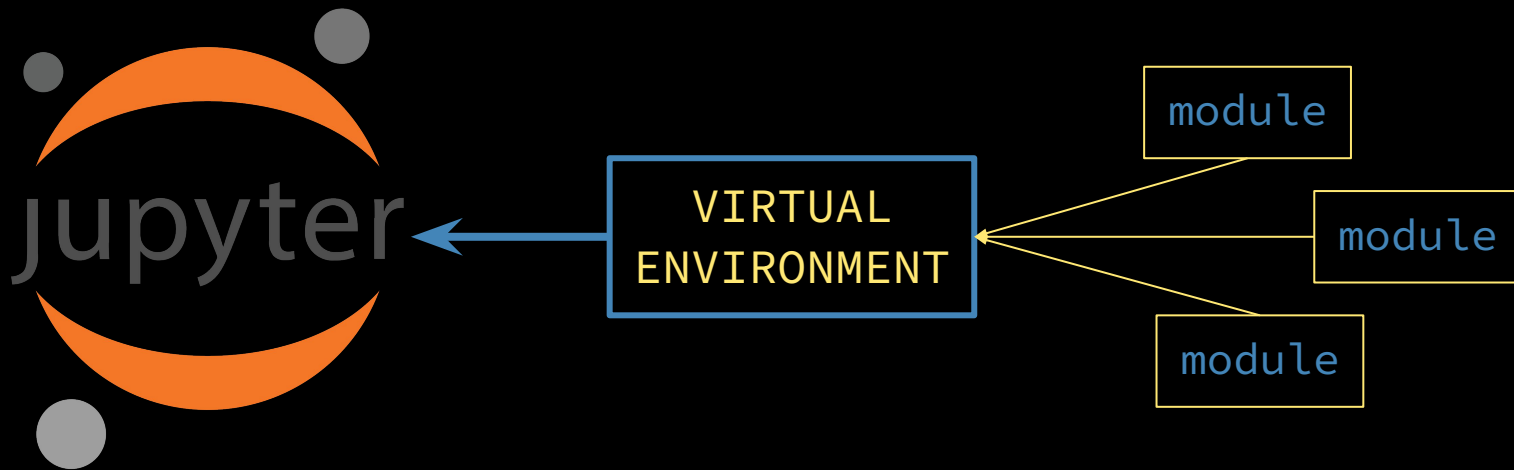
(3): Setting up a virtual environment

Python is a very powerful tool, but the version you install from online is pretty basic.

Modules are what make Python useful in astronomy. Think of modules as aftermarket modifications you install on a car: Toyota Corollas are quick, but a new body kit can make it quicker!

For Jupyter to recognize and use the modules you download, one should set up a virtual environment.

(3): Setting up a virtual environment



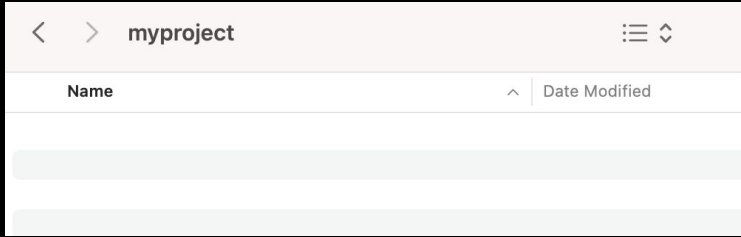
Think of `virtual environments` as profiles on a website.

You can load `Jupyter` without a `virtual environment`, and use it as a ‘guest user’, but your functionality will be limited.

Or, you can load into your profile, with your custom `modules` loaded, and `Jupyter` will have a lot more functionality.

Similarly, you can change between `virtual environments`, if you need different `modules`.

(3): Setting up a virtual environment



`python -m venv howdy_env`



- Open up a `terminal` (PowerShell if on Windows).
- Navigate to a folder where you want to install your `virtual environment`.
- Type in `python -m venv "environment name"`.
- This creates a folder called "environment name".

(3): Setting up a virtual environment

```
[(base) frank@Franklins-MacBook-Air myproject % source howdy_env/bin/activate  
[(howdy_env) (base) frank@Franklins-MacBook-Air myproject % pip list  
Package      Version  
-----  
pip          22.0.4  
setuptools   58.1.0
```

Now you need to make Jupyter recognize your virtual environment.

- While in myproject, type in:
`source [environment_name]/bin/activate` to load the virtual environment.
- You should now see your environment name in parentheses on the left, in my case I have `(howdy_env)`.
- Do a quick `pip list` to see what modules are loaded in your virtual environment.

(3): Setting up a virtual environment

```
[(howdy_env) (base) frank@Franklins-MacBook-Air myproject % python -m ipykernel install --user --name=howdy_env  
Installed kernelspec howdy_env in /Users/frank/Library/Jupyter/kernels/howdy_env  
(howdy_env) (base) frank@Franklins-MacBook-Air myproject %
```

Now you need to make Jupyter recognize your virtual environment.

- Check you are in your environment – see if your environment name is in brackets on the left.
- Type: `pip install ipykernel` and wait for `ipykernel` to install.
- Type:
`python -m ipykernel install --user --name=env_name`

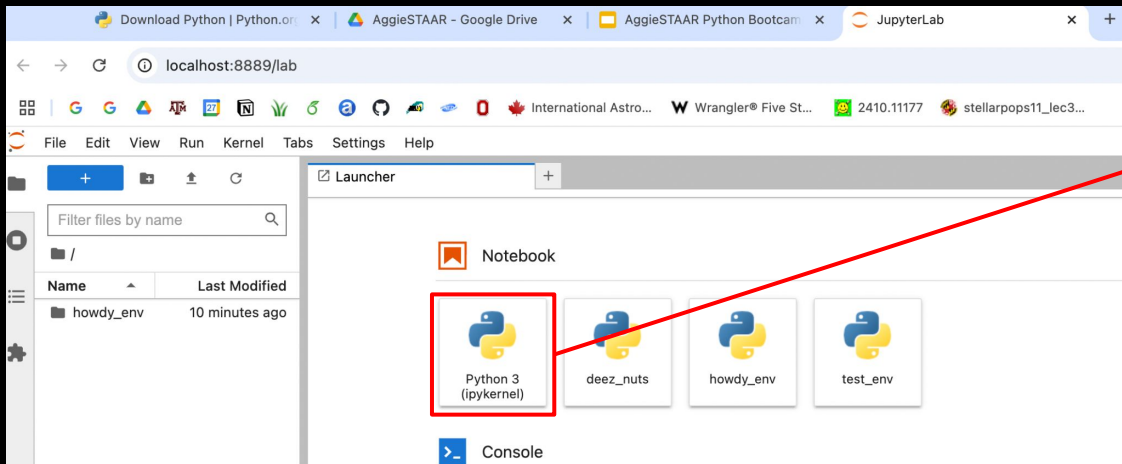
(3): Setting up a virtual environment

```
((howdy_env) (base) frank@Franklins-MacBook-Air myproject % pip install numpy
Collecting numpy
  Using cached numpy-2.0.2-cp39-cp39-macosx_10_9_x86_64.whl (21.2 MB)
Installing collected packages: numpy
Successfully installed numpy-2.0.2
```

Now we can install modules to the virtual environment. We already have installed one - ipykernel!

Go ahead and type `pip install numpy` into your terminal. Make sure that your virtual environment is loaded!

(3): Setting up a virtual environment



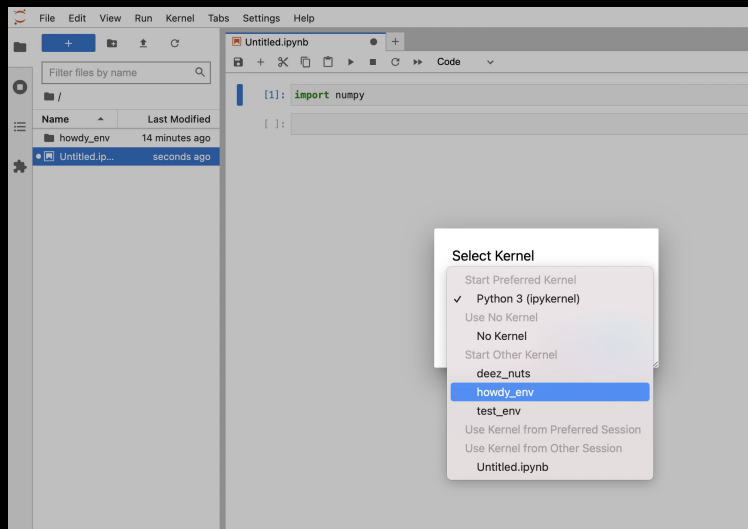
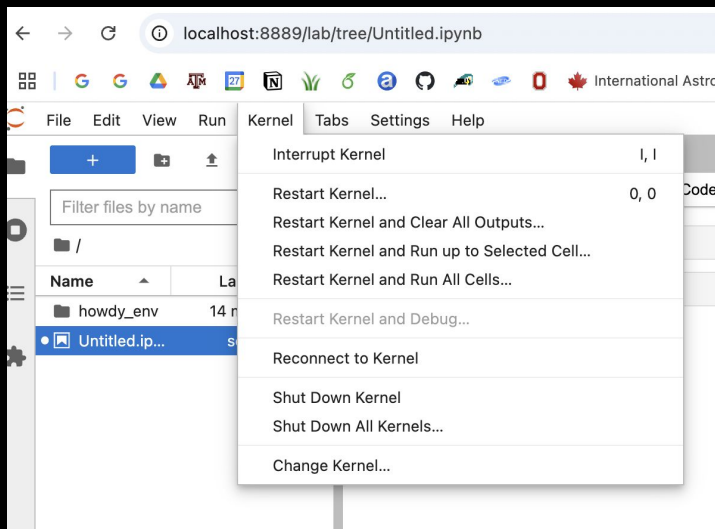
Start a new
notebook by
clicking here.

Now start up Jupyter:

- Use your current terminal (that is in myproject),
- OR, open a new terminal, and navigate to myproject.
- Type `jupyter lab` into your terminal.

Jupyter should open in a browser.

(3): Setting up a virtual environment



In Jupyter, navigate to “Change Kernel...” under the “Kernel” tab, and select the virtual environment that you created.

You will be able to select other virtual environments that you may have created.

(3): Setting up a virtual environment



Test that everything was installed OK: type `import numpy` into the first code block, then press `SHIFT + ENTER` to execute the code block.

Hopefully, this runs without issue!

If it did, congratulations! You've created and loaded into your first **virtual environment**. Now you should install other vital astronomy **modules**.

(4): Installing modules

Now, you should install the important astronomy Python modules:

- numpy (already installed)
- matplotlib (for plotting)
- astropy (a comprehensive collection astronomy Python packages)

Install these modules with `pip install [module name]`. Make sure you are loaded into your virtual environment when you do so!!

You should always test these modules (try importing them into Python) to make sure that they are correctly installed. If you get a “**module not found**” error, ensure that you’ve loaded the correct virtual environment (kernel) in Jupyter.

If you are in the correct virtual environment and you’re still getting the error, try reloading Jupyter. If the issue persists, check the module documentation, and see if there are extra installation steps you are missing.