Recommendations for Installing Python

Download and Install Anaconda

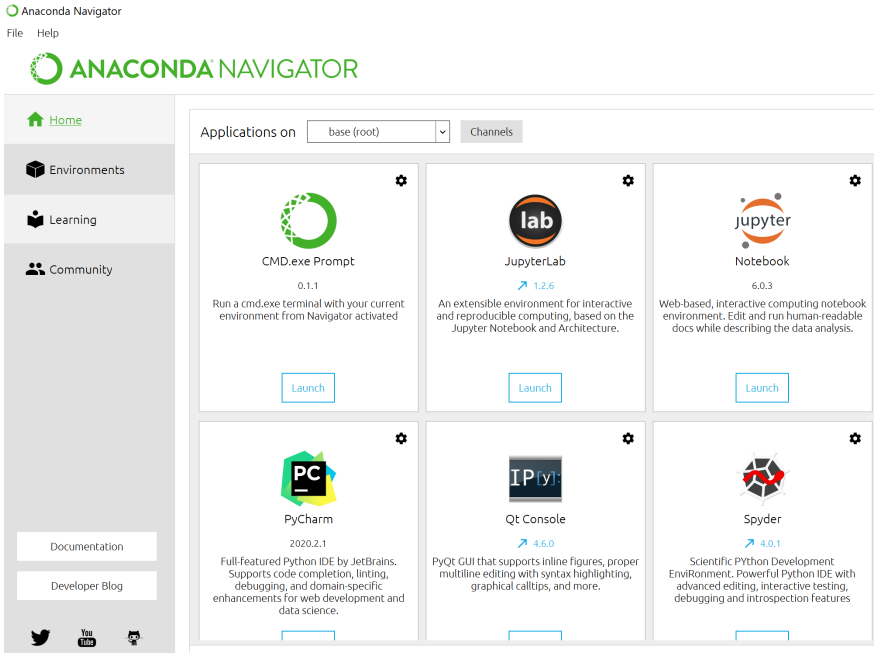
Python is included on many Unix installations. However, I recommend beginners install the Anaconda package manager. The Anaconda distribution includes standard Python, many scientific packages, and the Jupyter Notebook development environment that we use for examples in this book. Anaconda installs over 150 packages and is quite large (2 GB). If download speeds and space are a limitation for you, you can download Miniconda.

First, install the latest version of Anaconda (Python version 3.7, not version 2) appropriate to your operating system https://www.anaconda.com/download/ This step is operating system independent. Miniconda can be downloaded from https://docs.conda.io/en/latest/miniconda.html.

Anaconda comes with the Anaconda Navigator (https://docs.anaconda.com/anaconda/navigator/) which is a Desktop graphical user interface (GUI). If you are new to scientific programming, I recommend browsing their excellent user guide.

Launching Jupyter Notebooks

Both Windows and Unix users can open the **Anaconda Navigator** to launch Jupyter Notebooks.

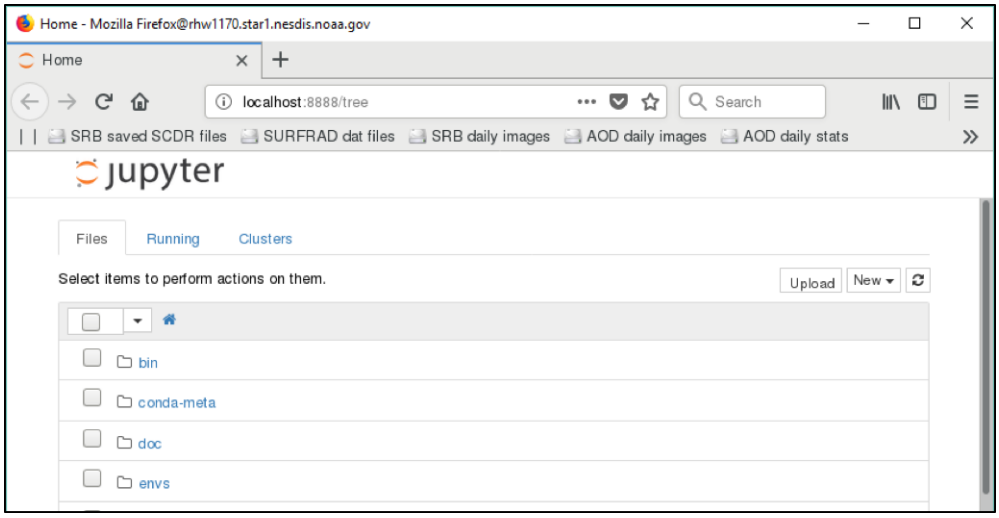


It can be faster, however, to open the terminal on Mac/Linux Machines/Anaconda Prompt in Windows and type:

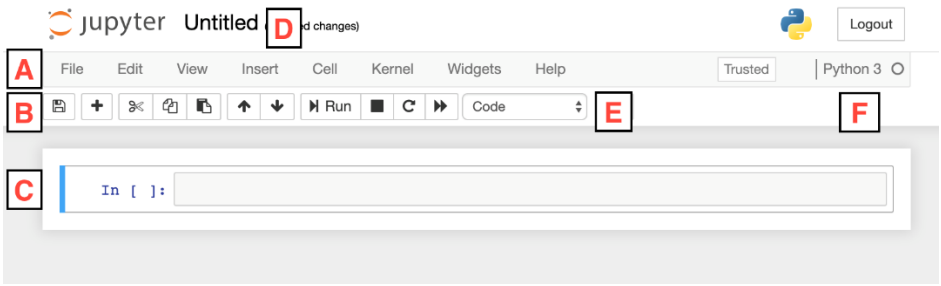
```
jupyter notebook
```

Windows users can navigate to a Jupyter Notebook icon from the start menu. If it succeeds, a

browser window will open and you will see the Jupyter notebooks main page. To create a new notebook, click on New Python 3. A new tab will open with an empty notebook.



In the annotated screen capture below, **A** shows the menu options. File contains tasks such as save/rename; edit allows you to add, copy, duplicate, and cut cells. Cell permits you to run code blocks within cells and set the cell type. The latter is useful for converting your notebooks to latex or html. The Kernel allows you to stop a command or the entire session if something goes wrong (e.g. you try to open too many files and your computer runs out of memory).



B shows many of the commonly accessed menu commands as icon shortcuts. The icons, from left to right, are save, new cell, cut a cell, copy a cell, duplicate cell, move cell up, move cell down, run active cell, stop the cell, restart the kernel, and restart and run all.

C is where we type code. Once you type your code into the cell, you can click the run button or hit Shift and Enter to run from the keyboard. The output will display below.

In **D**, you can rename the current notebook. By default, it will be called Untitled; get in the habit of giving them good names because it is easy to accumulate many over time.

E shows a dropdown menu for you to select cell type. The options include code (default), markdown, or raw. Markdown is a plain text formatting syntax, so it's valuable for documenting your code. Markdown's presence in Jupyter Notebooks is part of the reason they are called notebooks; you can mix code and detailed notes and documentation. Raw cell types are just plain text; all spaces and tabs will be preserved in this format.

Finally, **F** shows the status of the kernel. While most notebooks are written in Python, Jupyter notebooks also supports over 100 programming languages, although only include Python by

default. So, the top right corner indicates what language the kernel is in and the small dot to the right indicates the status. If code is running, the dot will be filled in and you must wait for the cell to finish running before another is processed.

Rebekah Esmaili
Sept 18, 2020