Python Installation Instructions
INFO 4390/5390 / CS 5382 Spring 2024

Welcome

This is a refresher on how to open, edit, and save Jupyter Notebook (.ipynb) files on your computer.

If you need help getting set up, please make sure to visit student (office) hours so that you can get individualized help from a TA. You may also want to post on Ed Discussion (linked from Canvas). Include screenshots or error messages if possible.

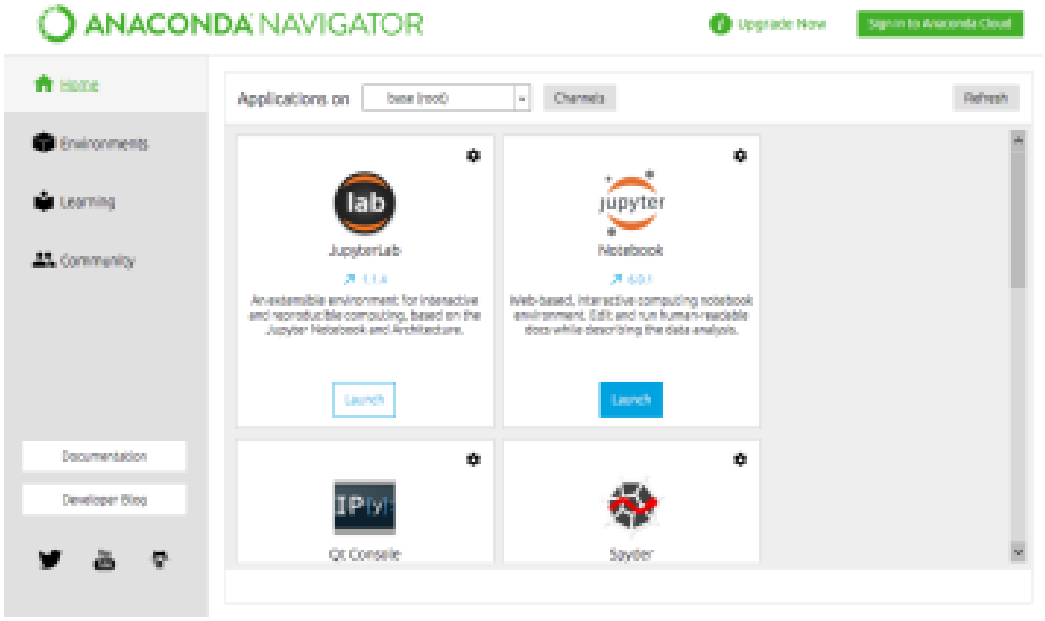
Installing Python and setting up Jupyter Notebooks

In this course, you can interact with Python through Jupyter Notebooks or Google Colab notebooks. Jupyter Notebooks are files with the extension .ipynb; they're documents where you can interactively execute code and also write text around the code, making it perfect for working on homework or presenting project reports. You can download Google Colab notebooks as .ipynb's as well.

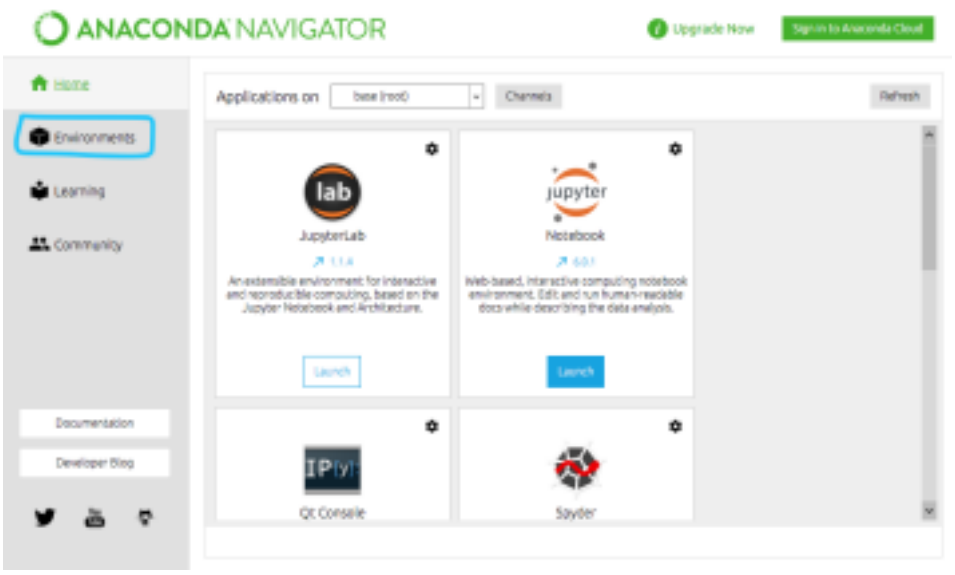
Even if you already have Python 3 installed on your system, please go through the steps below. We want to make sure everyone in the class has a consistent Python installation to avoid situations that are difficult to troubleshoot.

1. Anaconda is a Python installer and package manager that will help us get Python and important packages installed in a (hopefully) painless way. Start by installing Anaconda Distribution with the installer for your operating system. If you already have Anaconda installed, please be sure to update to the latest version.

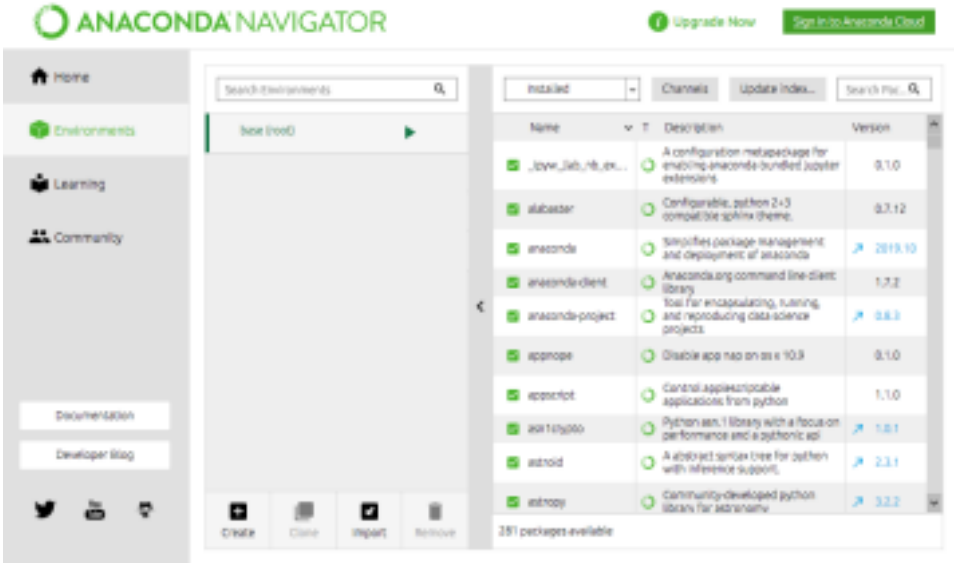
2. **Open the Anaconda Navigator app**. The main app window should look something like:



3. In the menu on the left, **select the “Environments” tab**:



You’ll be taken to a window that looks like this:

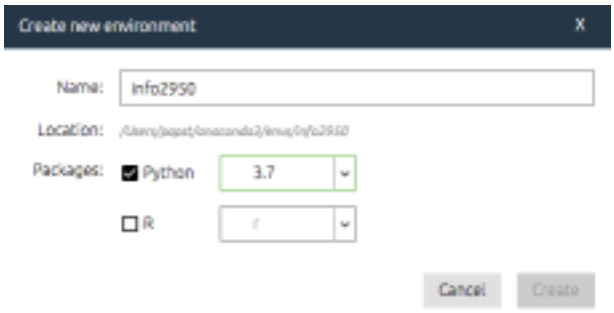


You will notice there is already one “base (root)” environment. We’re going to set up a new environment for our course-specific installation of Python to make sure that everyone’s setup is consistent, whether or not you’ve installed Python on your computer before.

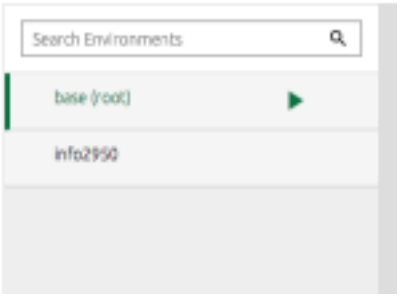
Basically, we’re going to make an info4390 “box” into which we will put all of our course software tools. When you want to access these tools, you have to tell your computer to look into the box for them (it will not know where they are by default). More on this later.

4. **Click the “Create” button** to make a new environment:

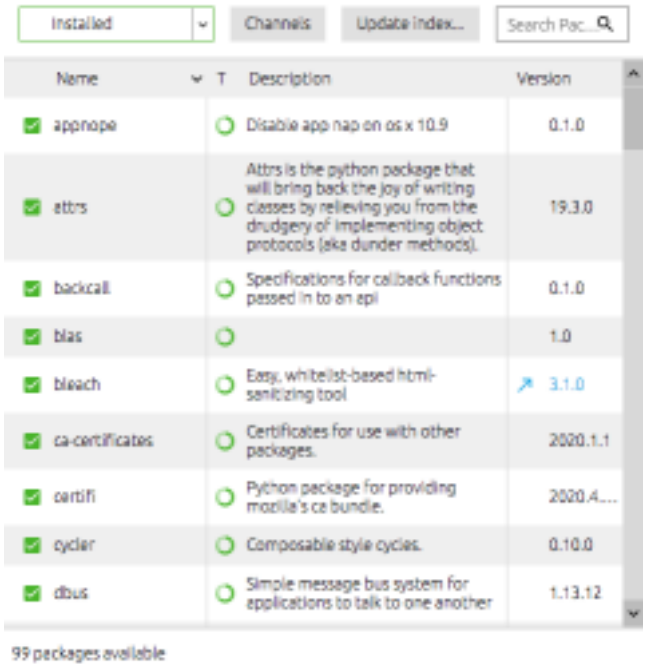
A dialogue box will open. Fill in **info4390** in the name field (the screenshot is older; you may see version 3.10 or 3.11):



Click create. You should now see a new environment called “info4390” in your environments list:



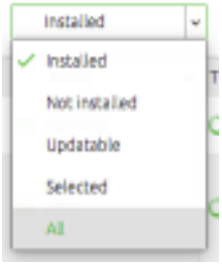
Click the “info4390” environment name. You should notice that window pane on the right refreshes:



This is the list of packages currently installed in your “info4390” environment. **Since we set up a *new* environment, it may not have any Python packages you may have installed for other courses.** Instead, it will have whichever default packages Anaconda thinks most users want in a new environment.

5. For our course, we need to **install a few packages** that don’t come in a new Anaconda environment by default, so let’s install them now.

   In the top left of the packages pane, there is a dropdown menu that reads “Installed”. From this menu, choose “All”:



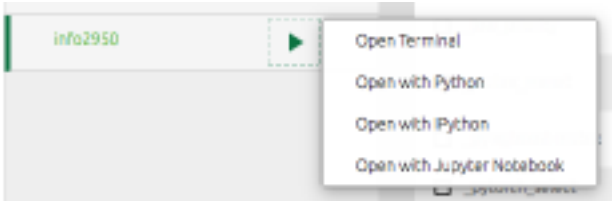
   Using the search bar in the top right of the packages pane, search for the jupyter package. You may have to click the “Update index...” button to refresh the package list. You will see a number of search results, but look for the one whose package name is just **jupyter**; its description should read “Jupyter metapackage. install all the jupyter components in one go”. Click the checkbox next to the package name and then click the “Apply” button in the bottom right of the packages pane (only appears when at least one package is selected). Anaconda will determine if there are any other packages that need to be installed in order for jupyter to work, and will ask if you want to install them. Click "Apply". This will install the jupyter package—it may take a few moments.

   Repeat the above process to also install **pandas** and **seaborn**.

   In the future, if you install a package while a notebook is running, you will need to restart the notebook before it can access the new package contents.

6. Now we just need to know how to **launch a Jupyter Notebook file**, because unfortunately it’s a little more complicated than just double clicking the file...

To open Jupyter Notebooks within the context of our course-specific install, we need to launch Jupyter from within our “info4390” environment. Select the “info4390“ environment from the environment list. You should see a triangular button appear by the environment name. Click this button and you will see a menu pop up:



Select “Open with Jupyter Notebook”. You’ll see a Terminal window pop up, followed by a new browser tab that takes you to a list of files on your system. By default, this will be in your computer’s “home” directory. You can now navigate to wherever the Jupyter notebook you downloaded lives on your computer, and then double click it (in the Jupyter browser) to open.

Resolving Package Install Errors

Sometimes, you may try to import a package but get an error that the module is not installed. When you search for the package in Anaconda Navigator (clicked on the info4390 environment), the package might not even show up when you search for it using the upper-right-hand-corner search bar! What should you do then?

Using the duckdb package as an example, we can do the following:

1. Open your terminal (on Mac, on Windows) and activate your virtual environment by typing:

   ```
   conda activate info4390
   ```

   After you hit enter, the next line that shows will start with ‘(info4390) username@...’ rather than ‘(base) username@...’

2. Now, install duckdb in the info4390 virtual environment by typing the following command into the terminal:

   ```
   conda install -c conda-forge python-duckdb
   ```

3. Now, duckdb should be installed. You should be able to successfully `import duckdb` in your IDE (e.g., in VS Code). You may need to close and reopen Anaconda Navigator to see it appear in the Anaconda Navigator interface.

Submitting Jupyter Notebooks on Gradescope

You will be submitting Jupyter Notebooks as .ipynb files on Gradescope. You can download them under the “File” dropdown in either VS Code or Google Colab.

If the above doesn’t work, here are additional instructions courtesy TA Arunabh Sarkar

I have seen a lot of students in my OH still run into issues with the Anaconda Environment that should

work for **both** Windows and Mac users.

This is how to install Jupyter Lab on your machine:

1. Open your terminal and type "pip --version" and ensure that you have pip installed on your machine. If this step is not completed please redirect to this page to install pip. Once you get a message back with something along the lines of "pip 21.3.1 from ..." then you are good to continue on.

2. Now in the command line, you will type "pip install jupyterlab" and let this run entirely. It should take a couple of seconds or minutes based on your machine. Make sure there is **NO** space between the words "jupyter" and "lab" when installing.

3. Once this is done, simply type in "jupyter lab" into the terminal and hit enter. It should either redirect you to a Jupyter Lab page within your browser or ask you to select your browser of choice (I recommend Chrome). Now you should be able to access your homework like normal from the file that they are stored in on your machine.

**As a note for future reference

a. Every time you want to close or reopen Jupyter Lab, all you have to do is open your terminal and type in "jupyter lab" and it should open automatically in your browser. For closing Jupyter Lab, close it like any other application by x'ing out of the terminal and the browser tab.

b. Every time you want to install new packages that do not exist in your Jupyter Lab environment, use "!pip install ___" and type in the pip install command for the package. To find the correct command, you can simply look up "pip install [package name]" and click the first link.