**Preparing Your Windows Computer to Run Python Code Using Anaconda and Jupyter:**

**Introduction to Anaconda for Windows**

This is the second publication of the series “Preparing Your Windows Computer to Run Python Code Using Anaconda and Jupyter”. The previous publication in the series explained what Anaconda is and why you would want to use it. The present publication covers how to install Anaconda for Windows and how to create your first conda software environment. We will use the conda environment we create here as we continue on to the next publications in the series.

**Installing Anaconda for Windows**

Instructions for downloading and installing Anaconda on Windows can be found on the Anaconda documentation page. Go to <https://docs.anaconda.com/free/anaconda/install/windows/> and follow all the installation instructions. When Anaconda has installed successfully, open Anaconda Navigator. INSERT PIC HERE

**Introduction to Anaconda for Windows**

The main feature of the Anaconda installation for Windows is the Anaconda Navigator software. It is a graphical user interface (GUI) organized into a series of panes that are accessible on the left side of the screen. The Home pane contains various apps you can install and launch. The Environments pane is where you can download python packages and create, modify, delete, or import software environments. The Learning page contains mostly links to documentation that might be useful as you start programming with python, as well as a few links to Anaconda tutorials and training. The Community pane contains links to online discussion forums like Stack Overflow where you can ask questions and search answers to previously asked technical questions.

The Anaconda installation for Windows comes with a few other features besides Anaconda Navigator. On your computer’s start menu, under Anaconda3 you will see these features: Anaconda Powershell Prompt, Anaconda Prompt, and Jupyter Notebook. These features are alternatives to using the Navigator GUI. This series of publications will cover Anaconda Navigator and some, but not all, of the additional Anaconda features.

For this publication, we will use the Home and Environments panes in Anaconda Navigator to edit our base (root) environment and then to create a new environment. We will also create a new environment using the Anaconda Prompt.

Note: it’s a best practice to create a new environment for any package installations and NOT install anything to the base environment. However, there is at least one exception to this rule, which we will cover here. We want to enable jupyter software to automatically access all of the conda environments we create so that the environments will be available when we open a jupyter notebook. The easiest way to enable this is to install one particular package to the base environment, but otherwise it’s best to avoid installing packages to your base environment.

*Edit your base (root) environment (to allow jupyter to automatically access all your conda environments)*

Step 1: If it didn’t open automatically, open anaconda navigator from your computer’s start menu. In Navigator on the left pane, click Environments.

You should see one environment called base (root). The green play button next to it indicates that it is the active environment. We want to install nb\_conda\_kernels to this base environment.

Step 2: First, in the drop down menu select All. Now, in the Search Packages bar type nb\_conda\_kernels. Then, select the check box next to nb\_conda\_kernels and click Apply at the bottom right.

You’ll see the Install packages window pop up. Anaconda then executes its “solver” which searches for any package conflicts. Eventually, you will get a message about all the new packages that will be installed. This list of packages includes nb\_conda\_kernels as well as any additional packages that nb\_conda\_kernels requires (these are called dependencies and are marked with an asterisk in anaconda navigator).

Step 3: Click apply on the Install packages pop up window, which will start the installation. When the installation finishes the pop up window will disappear and the play button next to the base environment should again be green. Click the x in the Search Packages bar next to nb\_conda\_kernels to clear it from the search.

*Create a new conda environment using Anaconda Navigato*

Now we will do a similar process to create your first environment separate from the base environment. As a reminder, it’s a best practice to leave the base environment alone and install any new packages you need into new environments.

Step 1: In the Environments pane of Anaconda Navigator, click the create button. In the Create new environment pop up type myfirstenv in the name box, accept the selected default location and packages, and click create.

Your environment should be created and is now active if it has a green play button next to the environment name. Currently there is only the python core package along with its dependencies installed in your new environment. Note, only one environment can be active at a time and an environment must be active to install new packages into it.

Step 2: Add 3 additional packages to your new conda environment. First, in the drop down menu make sure All is chosen. In the Search Packages box type numpy, then click the check box next to the package called numpy. Now we will repeat this two more times with the packages matplotlib and ipykernel. In the Search Packages box type matplotlib, then click the check box next to the package called matplotlib. In the Search Packages box type ipykernel, then click the check box next to the package called ipykernel. Then click Apply at the bottom right.

You’ll again see a pop up while Anaconda searches for and resolves any package conflicts.

Step 3: When the full list of packages to be installed appears, click Apply on the pop up window, which will start the installation.

The pop up window will disappear and you will see a progress bar indicating that Anaconda is installing packages. When the installation finishes, the progress bar will disappear and the play button next to myfirstenv should again be green.

Optional step: If you click the x in the search bar next to ipykernel and select “Installed” from the drop down menu, you will be able to see the full list of packages installed to your myfirstenv environment. Most of the packages are actually dependencies, but scrolling through the list you will be able to see numpy, matplotlib, and ipykernel, as well as the python version we accepted as a default when we first created the environment.

*Create a new conda environment using Anaconda Prompt*

Now we will create an identical environment to myfirstenv using the Anaconda Prompt instead of Navigator to illustrate the differences in the process of environment creation. If you are familiar with typing at a command line you may find this method much easier than using Navigator.

Step 1: On your computer’s start menu, under Anaconda3 click Anaconda Prompt to access the Anaconda command line prompt.

You should see (base) followed by a directory location such as C:\Users\username followed by the prompt symbol >. The word in the parentheses (base) indicates which conda environment is active. The directory is the current directory location. The prompt indicates where to start typing

Step 2: type the following at the prompt and hit enter

conda create –name env2 numpy matplotlib ipykernel

This command will create an environment identical to the one we previously created except this time we name it env2. You will see messages appear indicating that Anaconda Is search for and solving any conflicts between packages. Eventually, the long list of packages to be installed will appear on the screen.

Step 3: Hit Enter or type y to proceed with the installation

You will see progress bars appear and disappear as packages are installed and then some messages about the status of preparing, verifying, and executing the “transaction” or the creation of the environment.

Step 4: Install an additional package in env2. Notice the base environment is still active. If you want to install additional packages to env2, you must first activate it because packages are installed to whichever environment is active. Type at the prompt

conda activate env2

You should see (base) change to (env2), meaning the env2 environment is now activate. Install an additional package by typing at the prompt

conda install xarray

And hit Enter or y to proceed with the installation.

*How to see all available conda environments*

In Navigator, all available environments are listed on the Environments pane

or

At the Anaconda Prompt type

conda env list

You should see the base environment, as well as the two environments you have created.

*How to see all packages installed in an environment*

In Navigator on the Environments pane, select one of the environments by clicking it and a green play button will appear indicating the environment is now active. If you select Installed from the drop down menu and make sure the Search Packages bar is empty, a full list of all the packages installed in the selected environment will appear.

Or

At the Anaconda Prompt, make sure the environment you are interested in is activated and then list its packages by typing two separate commands

conda activate myfirstenv

conda list

We’ll cover additional conda tips like how to import and export an environment in a later publication in this series. For now, continue on to the next publication in the series to learn how to open a jupyter notebook and access your new conda environments from inside the notebook.

IN ANOTHER PUB:

Also show the way to install the environment as a jupyter kernel with python -m ipykernel install