

Setup your Python environment

You are free to setup your python environment however you like. The following is my recommended approach and what I will assume you have done for the purposes of this course.

1. [Install Conda](#)
 2. [Configure Conda](#)
 3. [Create a sandboxed Python environment for this course](#)
 4. [Run JupyterLab](#)
 5. **OPTIONAL:** [Install Visual Studio Code \(VSCode\)](#)
- [Python in the cloud](#)

1. Install Conda

Download and install the latest version of [Miniconda3](#).

*On macOS I suggest using the **pkg** installer as it is the simplest.*

2. Configure Conda

Conda groups collections of packages into [channels](#). Most of the packages that you will install for this course will come from either the defaults or [conda-forge](#) channels. Conda only searches channels that you tell it to. To setup conda so that it automatically searches the [conda-forge](#) channel before searching the defaults channel, open a command shell or terminal and run:

```
conda config --add channels conda-forge
```

To see a list of all of the channels that conda will search (top = highest priority, bottom = lowest priority):

```
conda config --show channels
```

Note that you can always specify the channel to search when installing packages on a per package basis. An example of this is shown in the next step.

3. Create a sandboxed Python environment for this course

First, create a new conda environemnt named "neu365". In a command shell or terminal run:

```
conda create --name neu365
```

Next, set neu365 as the active environment for all conda commands. Run:

```
conda activate neu365
```

Now install the latest version of the **python** package (from the defaults channel) into the active (neu365) environment.

```
conda install python
```

Finally, install the latest version of the **jupyterlab** package (from the conda-forge channel) into the active (neu365) environment. This will give you everything you need to work with [Jupyter notebooks](#) which we will use in this course.

```
conda install jupyterlab -c conda-forge
```

*Note that if you configured conda to automatically search the conda-forge channel as described above, then you could omit the **-c conda-forge** portion of the last command.*

⚠ Note that each time you open a new command shell or terminal, you will need to activate the conda environment that you want to use:

```
conda activate neu365
```

The currently active environment should be displayed in your command prompt, e.g. **(neu365)**.

You can see a list of all your conda environments with:

```
conda env list
```

You can see a list of all installed packages in your active environment with:

```
conda list
```

If you ever want to delete this environment and start over:

```
conda remove --name neu365 --all
```

For just about everything else you can do with conda there is a plethora of available online information.

4. Run JupyterLab

[JupyterLab](#) is a complete Python integrated development environment (IDE) that supports Jupyter notebooks (which we will use in this course). It runs in a tab in your browser. You should already have installed JupyterLab in the previous step.

To run JupyterLab first make sure your neu365 environment is active:

```
conda activate neu365
```

Then, simply run the command:

```
jupyter lab
```

This will start **JupyterLab** which will open up a tab in your browser with a complete user interface for writing and running Python code.

When you are done using JupyterLab, from within the JupyterLab browser interface select **File->Shutdown**.

5. OPTIONAL: Install Visual Studio Code (VSCode)

There are a variety of user interfaces for Python coding from the most basic text editor and cmd shell to more complex integrated development environments (IDEs) with more features such as JupyterLab. One alternative to JupyterLab that is compatible with Jupyter notebooks is VSCode.

1. Download [Visual Studio Code \(VSCode\)](#)
2. Open VSCode and install the **Python** and **Jupyter** extensions.
3. In VSCode, Select **View->Command Palette...** and search for and select **Python: Select Interpreter**, then select the Python version associated with your neu365 environment (e.g., **Python 3.11.0 ('neu365')**).
4. In VSCode you can also set the Python interpreter for each individual file. When you open a Python file the interpreter will be displayed in one of the corners. Make sure it is set to the Python version associated with neu365, otherwise click the displayed interpreter and select from the dropdown the one for neu365.

It does NOT matter which user interface you use. Both JupyterLab and VSCode will give you a pretty similar experience, as will several other IDEs such as [PyCharm](#) or [Spyder](#).

Python in the cloud

[Google Colab](#) is a purely cloud-based option that is worth taking a look at. However, for this course I want you to setup a local Python environment on your computer by following steps 1-4(5) above. This way you will be able to familiarize yourself with managing your own packages and virtual environments, both of which are important skills for working in a general Python ecosystem.