Python Setup

We will be using python (specifically python3) for many of the assignments and breakout sessions in this class. As such, it is incredibly important that you have python3 installed and are capable of installing packages and loading the Jupyter (formerly iPython) notebooks we will use. This document will walk you through the installation.

Note: I have developed this guide with MacOS and Linux in mind, as I do not have a Windows machine for active development. If you have a Windows machine, I recommend that you use [Windows Subsystem for Linux](), which lets you use a Linux development environment on your Windows machine. If you do develop on Windows, I expect that most things in this course will work fine, but my lesser experience means it will be harder for me to help you resolve issues.

Quick Start (for the experienced)

We will be using a virtual environment for this course and I have provided the packages you will be using in the `requirements.txt` file found in the same directory as this guide. Setup will proceed as follows:

(You can also use Anaconda to setup the environment. The following tutorial is just for virtual environment without Anaconda. I strongly suggest everyone use Anaconda to setup the environment.)

```
# Move to where you would like to store your virtualenv.
# I recommend the parent of the folder with this file.
cd $PARENT_FOLDER

# Create a virtual environment and activate it
# IMPORTANT: you should use a Python version from
# Python 3.8 - Python 3.11 (3.12 may not work)
python3 -m venv cs5404venv
source cs5404venv/bin/activate

# Install the packages
pip3 install -r requirements.txt

# Add the new virtual environment “kernel” to iPython/Jupyter
# (needed to use packages in this environment)
ipython kernel install --user --name=cs5404venv --display-name
"Python3 (CS5404)"
# Launch Jupyter
```

```
jupyter notebook
```

Install Python3

We will be using python (specifically python3) for all the assignments in the class.

IMPORTANT: you might want to prefer using a Python version from Python 3.8 – Python 3.11 (3.12 may not work). On my Mac, `brew install python@3.11` installs the version I wanted and replacing `python3` with `python3.11` in all commands below made sure I used the correct version.

Open a terminal window, `terminal` on MacOS and Linux and `Command Prompt` on Windows, and type `python3 --version`. If this succeeds without error, you can move on to the next step. If not, you may need to install python3. Installation varies based upon the operating system:

- On the Mac, typing python3 at the command line might prompt install via the command line utilities (via Xcode). Alternatively, the recommended install uses the [Homebrew package manager]() (with `brew install python`)
- Some versions of Linux already have python3 installed, but the most common way to install python3 on Debian systems is with `sudo apt install python3` (see [this guide]() for more details).

Create a Python3 Virtual Environment

Since we will require a number of packages to run code, it is most convenient to use a *python virtual environment* to install the packages, so that the installation does not happen operating-system-wide and conflict with other projects you may be working on elsewhere. The Virtual Environment installs packages *locally* so that they do not conflict. If you are using Windows, you might want to follow the instructions here instead: https://docs.python.org/3/library/venv.html

The following code will create a virtual environment in a folder called cs5404venv in the current directory and "activate it":

```
# Move to the desired folder (example: cd /Users/cezhou/class/L00/.. )
# I recommend the parent of the folder with this file.
cd $PARENT_FOLDER

# Create a virtual environment and activate it
```

```
python3 -m venv cs5404venv source
cs5404venv/bin/activate
```

Once activated, you might notice that your prompt changes at the command line to include `cs5404venv` before your cursor; this indicates success. If that succeeds, we can install the packages via pip, the python package manager:

```
# Install the packages
pip3 install -r L00_python_getting_started/requirements.txt
```

This will take a few minutes.

Configure and run the Jupyter Notebook

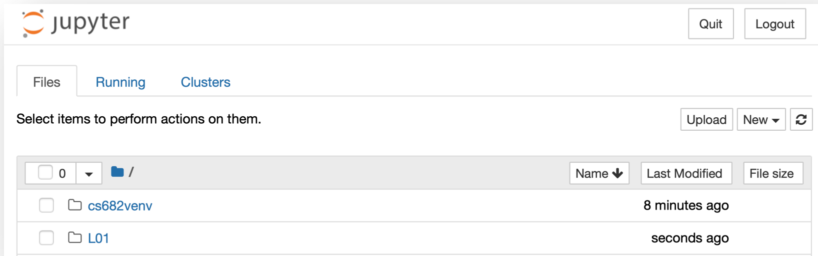
We will be using a Jupyter Notebook (formerly iPython) for running code in this class. It provides an excellent way to write snippets of code, create figures, and include helpful information and LaTeX all in a single document. Before running, Jupyter needs to be "told" about your virtual environment and the packages you have installed (the "kernel" which will be running the code in the background). To do so, run the following command:

```
# Add the new virtual environment "kernel" to iPython/Jupyter
# (needed to use packages in this environment)
ipython kernel install --user --name=cs5404venv --display-name
"Python3 (CS5404)"
```

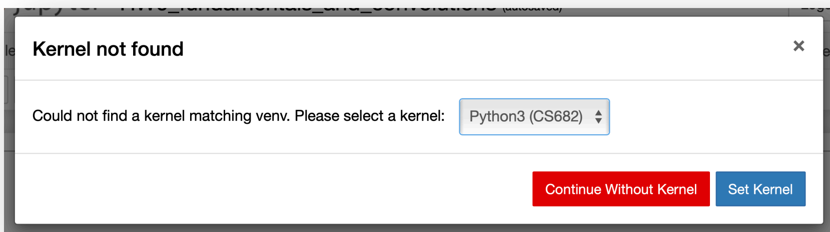
Once this is done, you should be able to run the Jupyter environment:

```
# Launch Jupyter
jupyter notebook
```

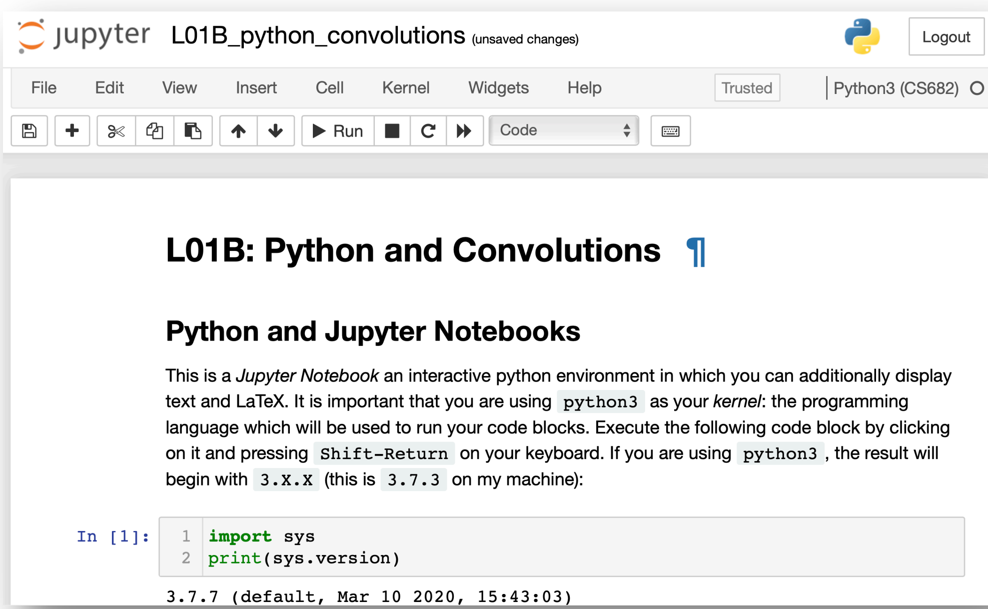
You should be greeted with a page that looks something like this (except with `cs5404` instead):



Click on this folder and then on the `basics_python_numpy.py` notebook contained inside. Upon opening one of the notebooks I have provided for the course, you *may* get an error message prompting you to select a new kernel. Select the one you just created as follows:



You should see a screen that looks like this:



Once you have opened the notebook, you can follow the instructions there to move forward.