COMP3222/6246 Week 1 (Not marked)
Setting Up an Environment

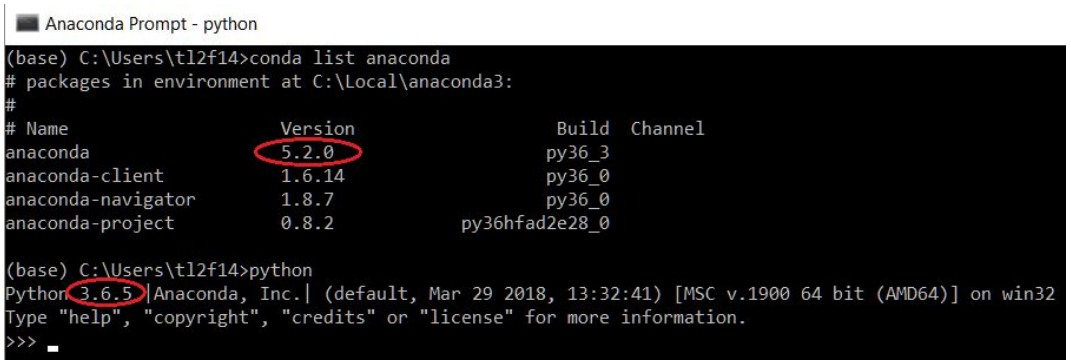
This document is a rough guide to help you setting up your programming environment before starting the laboratory session on the 2nd week. Though exercises are not marked, they are provided to help you familiarise with the Anaconda environment. Therefore, we strongly encourage you to try the exercises.

Install Anaconda

A number of programming languages including Python have softwares or *distributions* that help managing and maintaining the compilers and other necessary libraries. We will use Anaconda Python which is one of the popular distributions that allow us to easily develop machine learning projects. You can skip this step if you are working on University's PC and there is Anaconda 3 pre-installed.

To install the latest Anaconda with Python 3, download the Anaconda 3 installer from https://www.anaconda.com/download/ and install accordingly. Refer to Anaconda documentation for installation guide https://docs.anaconda.com/anaconda/install/.

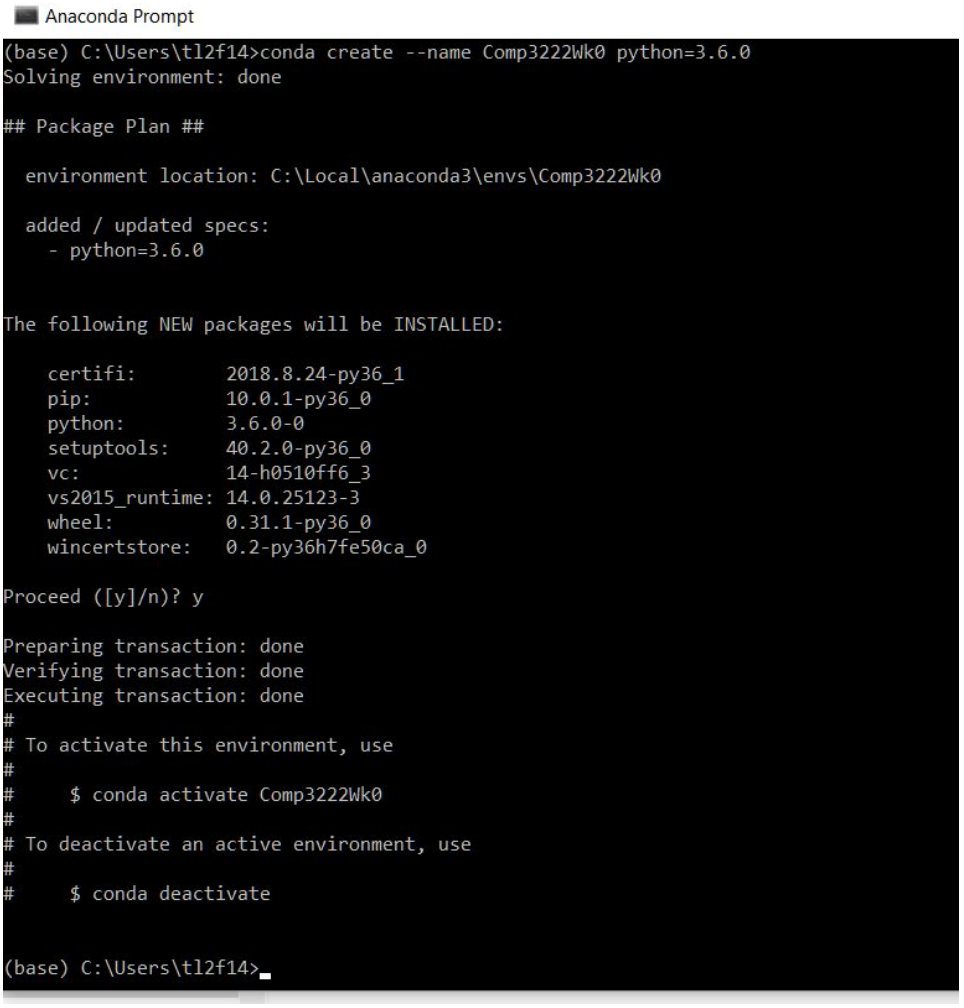
Next, verify your installation and your version of Anaconda and Python via Anaconda Prompt by following instructions on https://docs.anaconda.com/anaconda/install/verify-install. Make sure that your Anaconda's and Python's versions are at least 5.2 and 3.6 respectively.



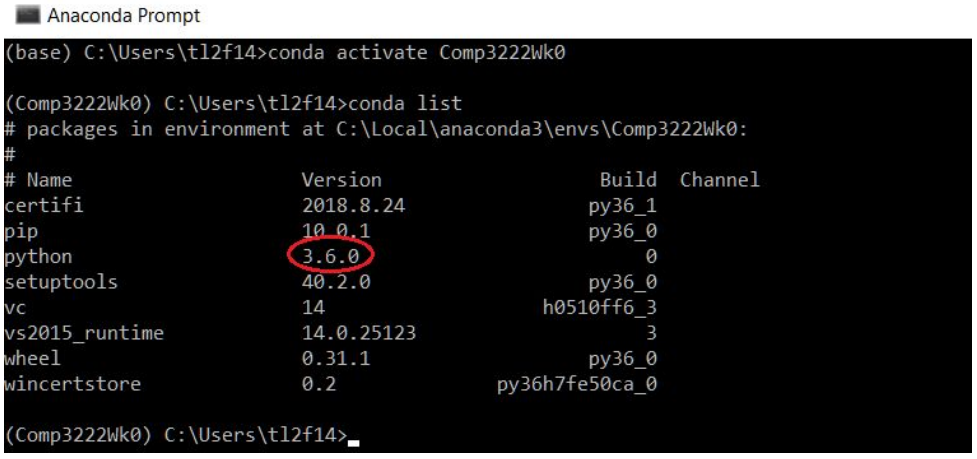
Working with Anaconda's environments

One of the useful features in Python is an ability to set up a separate working *environment* for each software project. Each environment is completely separate from each other. They will individually contain their files, packages, and their dependencies. It is a good practise to employ different environments for different projects. (Do a Google search on 'virtualenv' if you would like to learn more.)

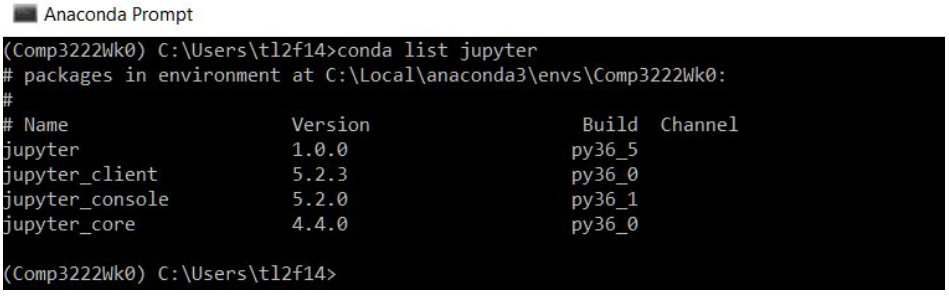
Now, let us create an environment with *Python 3.6 or better* to store all of your work. (Again, it is a good practise to employ different environments for different work and not directly use the default environment 'base' to store everything.)



Activate the new environment and verify that the specified version of Python has been installed. Then, try deactivating the new environment, use the same command to check all installed packages, and observe the difference between the new environment and the base environment.



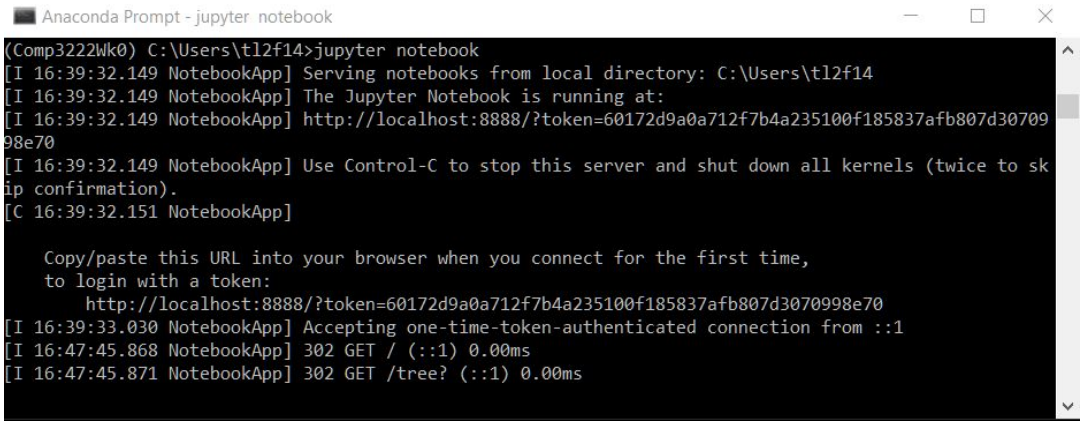
Before proceeding to the next section, in the new environment, install a package ‘jupyter’ via command ‘conda install jupyter’. Don’t forget to confirm that you have installed it properly.



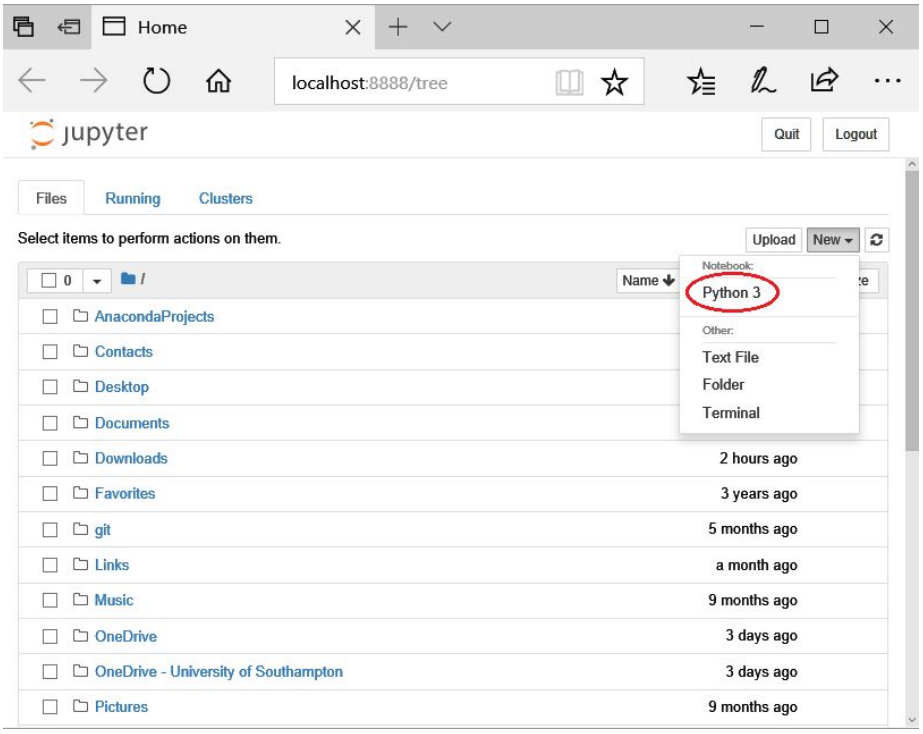
Jupyter Notebook

Jupyter Notebook is an open-source web application that provides an interactive coding environment and is geared toward data science project. The Jupyter Notebook can be used as a quick testbed, and it will be mostly used throughout our lab sessions. For further details on the Jupyter Notebook, refer to their online documentation on https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

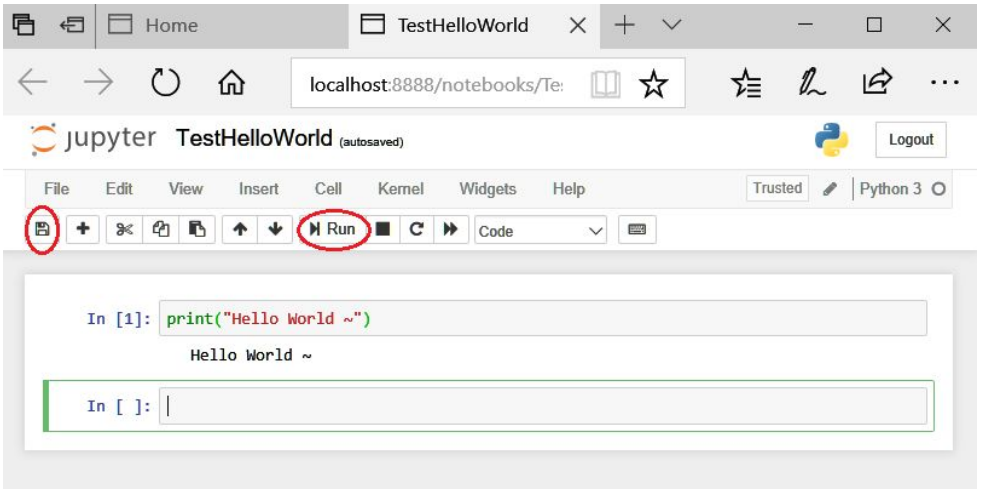
Now, start a Jupyter server by running a command 'jupyter notebook' in your Anaconda Prompt. The Jupyter server will then listen on port 8888 for any connection request. If you are using Windows, a new window showing Notebook Dashboard will be started. If not, you can access it by using a URL shown on the Anaconda Prompt.



Then, create a new notebook with Python 3 on the Notebook Dashboard. Observe in the Dashboard that a new notebook with name 'Untitled.ipynb' is created, and a new window/tab accessing the new notebook is also opened. This is called an editor where you can do coding interactively. For further details, have a look on the online documentation https://jupyter-notebook.readthedocs.io/en/stable/ui_components.html.



Familiarise yourself with the editor by changing the name of the current notebook, executing and saving Python code. Changing the name can be done easily by clicking the name at the top of the editor and modifying it accordingly. (Observe in the Notebook Dashboard that the filename is also changed automatically.) Executing a code and saving your work is easily done by clicking 'Run' button and saving button after you have typed in the code. Refer to other usages and examples of Jupyer Notebook on https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html.



Exercises (Not marked)

1. Familiarise with Anaconda's commands (especially commands for managing Conda environments) in the cheat sheet https://conda.io/docs/_downloads/conda-cheatsheet.pdf.
2. Create a new conda environment with packages Python 3.6 (or better), jupyter, matplotlib, numpy, pandas, scipy, and scikit-learn. These packages will be used throughout our lab sessions. Refer to the online documentation https://conda.io/docs/commands/env/conda-env-export.html as necessary.
3. Export the previously created environment by using command 'conda env export' into a file with .yaml extension. This yaml file lists all of the installed packages and their versions. It is also useful

for recreating a new environment with the same packages. Refer to the online documentation https://conda.io/docs/commands/env/conda-env-export.html as necessary.

4. Create another conda environment with a different name from the yaml file by using command ‘conda env create’. Verify that both environments have the same packages with the same version. Refer to the online documentation https://conda.io/docs/commands/env/conda-env-create.html as necessary.