

CS917 - Programming Labs

Setting Up a Python Environment

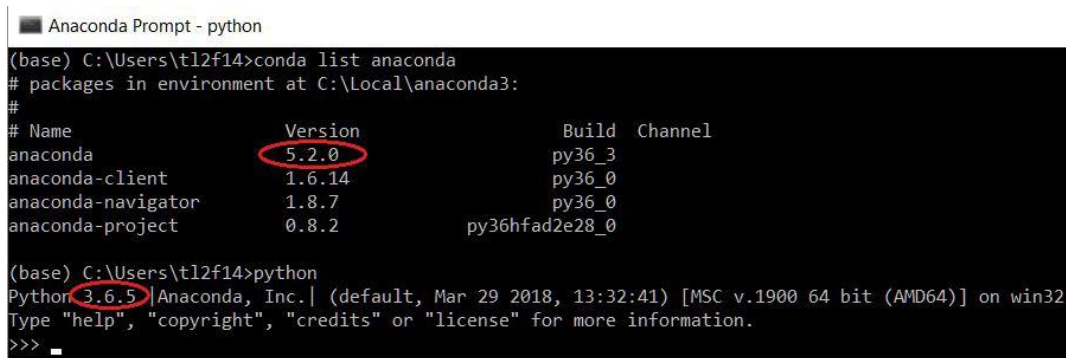
This document is a rough guide to help you setting up your programming environment before starting the laboratory session on the 2nd week. Please ignore this document if you already have Python installed on your computer/laptop, Otherwise please go through this document before the labs as it is to help you familiarise with the Anaconda/Python environment.

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.

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.



```
Anaconda Prompt - python
(base) C:\Users\tl2f14>conda list anaconda
# packages in environment at C:\Local\anaconda3:
#
# Name                  Version              Build    Channel
anaconda                5.2.0                py36_3
anaconda-client         1.6.14               py36_0
anaconda-navigator      1.8.7                py36_0
anaconda-project        0.8.2                py36hfad2e28_0

(base) C:\Users\tl2f14>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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.)

```
Anaconda Prompt
(base) C:\Users\tl2f14>conda create --name Comp3222Wk0 python=3.6.0
Solving environment: done

## Package Plan ##

  environment location: C:\Local\anaconda3\envs\Comp3222Wk0

added / updated specs:
- python=3.6.0

The following NEW packages will be INSTALLED:

certifi:          2018.8.24-py36_1
pip:              10.0.1-py36_0
python:           3.6.0-0
setuptools:       40.2.0-py36_0
vc:               14-h0510ff6_3
vs2015_runtime:  14.0.25123-3
wheel:            0.31.1-py36_0
wincertstore:     0.2-py36h7fe50ca_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate Comp3222Wk0
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\Users\tl2f14>
```

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.

```
Anaconda Prompt
(base) C:\Users\tl2f14>conda activate Comp3222Wk0

(Comp3222Wk0) C:\Users\tl2f14>conda list
# packages in environment at C:\Local\anaconda3\envs\Comp3222Wk0:
#
# Name          Version          Build      Channel
certifi         2018.8.24        py36_1
pip             10.0.1           py36_0
python          3.6.0            0
setuptools      40.2.0           py36_0
vc              14               h0510ff6_3
vs2015_runtime  14.0.25123       3
wheel           0.31.1           py36_0
wincertstore    0.2              py36h7fe50ca_0

(Comp3222Wk0) C:\Users\tl2f14>
```

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.

```
Anaconda Prompt
(Comp3222Wk0) C:\Users\tl2f14>conda list jupyter
# packages in environment at C:\Local\anaconda3\envs\Comp3222Wk0:
#
# Name                        Version      Build    Channel
jupyter                      1.0.0        py36_5
jupyter_client               5.2.3        py36_0
jupyter_console              5.2.0        py36_1
jupyter_core                  4.4.0        py36_0

(Comp3222Wk0) C:\Users\tl2f14>
```

Jupyter Notebook - optional

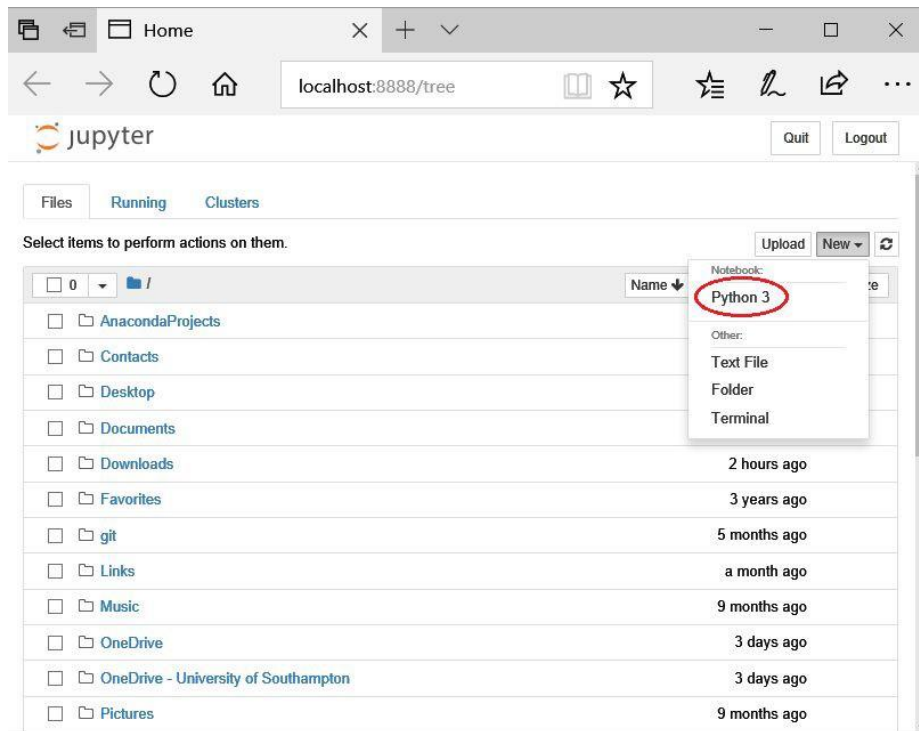
Jupyter Notebook is an open-source web application that provides an interactive coding environment and is geared toward data science project. The Jupyter Notebook can be used as a quick testbed, and it will be mostly used throughout our lab sessions. For further details on the Jupyter Notebook, refer 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.

```
Anaconda Prompt - jupyter. notebook
(Comp3222Wk0) C:\Users\tl2f14>jupyter notebook
[I 16:39:32.149 NotebookApp] Serving notebooks from local directory: C:\Users\tl2f14
[I 16:39:32.149 NotebookApp] The Jupyter Notebook is running at:
[I 16:39:32.149 NotebookApp] http://localhost:8888/?token=60172d9a0a712f7b4a235100f185837afb807d3070998e70
[I 16:39:32.149 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:39:32.151 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=60172d9a0a712f7b4a235100f185837afb807d3070998e70
[I 16:39:33.030 NotebookApp] Accepting one-time-token-authenticated connection from ::1
[I 16:47:45.868 NotebookApp] 302 GET / (::1) 0.00ms
[I 16:47:45.871 NotebookApp] 302 GET /tree? (::1) 0.00ms
```

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 Jupyter Notebook on https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html.

