Jupyter Notebook Introduction

Introduction

This file is a quick introduction to using Jupyter notebook. It is based on guide developed by Truyen Tran (Deakin University). It is recommended that students unfamiliar with Jupyter Notebook work through this document.

Python Overview

Jupyter Notebook is an interface to Python, surrounded by a rich environment for documenting, visualizing, and writing. It is based on the IPython shell, but provides a cell-based web page environment with great interactivity, where calculations can be organized and documented in a structured way. In this note we will demonstrate how to load (read) a notebook, execute, and create one.

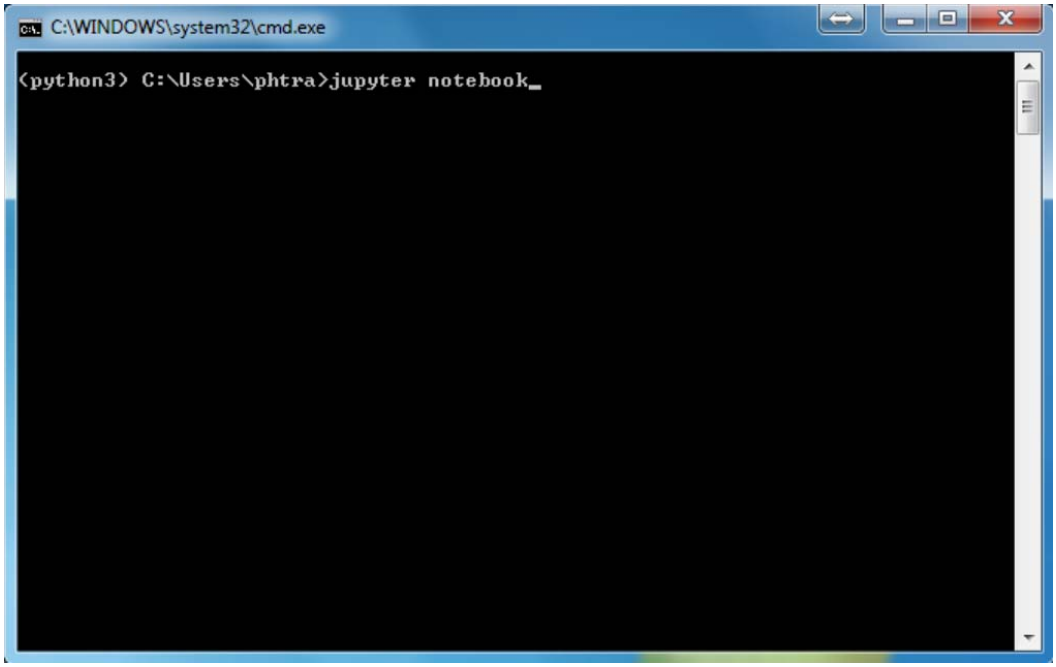
Since you have installed Anaconda Scientific Python Distribution in the previous tutorial (1-instructions.pdf) you already have all the necessary packages and modules to start Jupyter Notebook Server and use Jupyter Notebooks.

Jupyter Notebook is formerly known as IPython Notebook. Before IPython 3.0 (released on 28 Feb 2015) IPython Notebook was a part of IPython project. Since many of its components are language-agnostic and can be used with other languages, after this release they have been moved to Jupyter Notebook for better development. Jupyter Notebooks can be used along with languages such as Julia, 'Python', R, Ruby, 'Matlab', 'Octave', and many other.

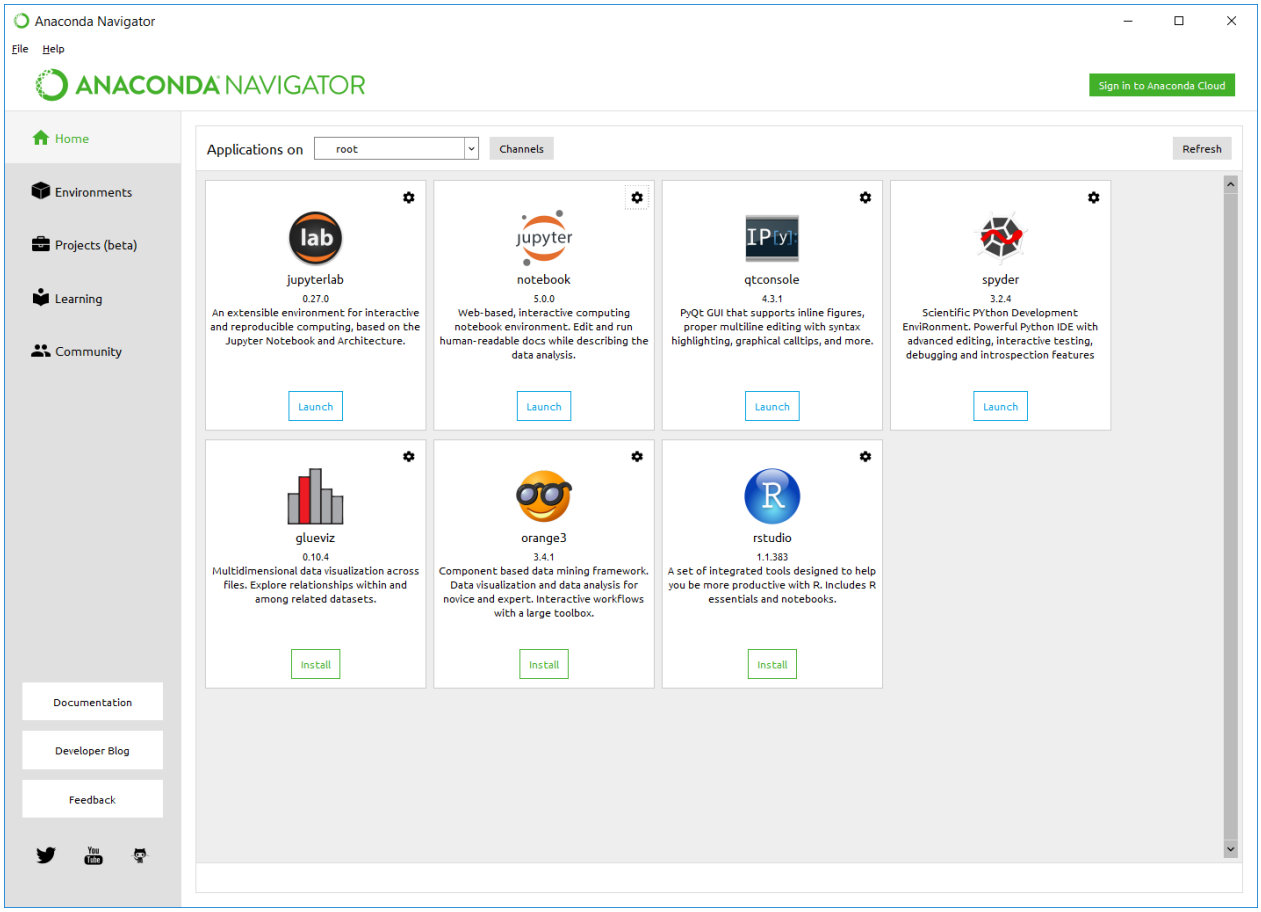
Note: From now on whenever we mention IPython notebook, in fact we mean a Jupyter notebook which runs a Python kernel. We may also use IPython notebooks and Jupyter notebooks, interchangeably.

Starting Jupyter Notebook server

To start using Jupyter Notebooks, first you need to start Jupyter Notebook server. Execute the following command jupyter notebook in terminal to start the server:



Alternatively, you can start the Anaconda application and select launch on Jupyter Notebook.



Either approach will open a new browser window (or a new tab in an existing window) with an index page where existing files and folders are shown. The information printed on the terminal screen says that:

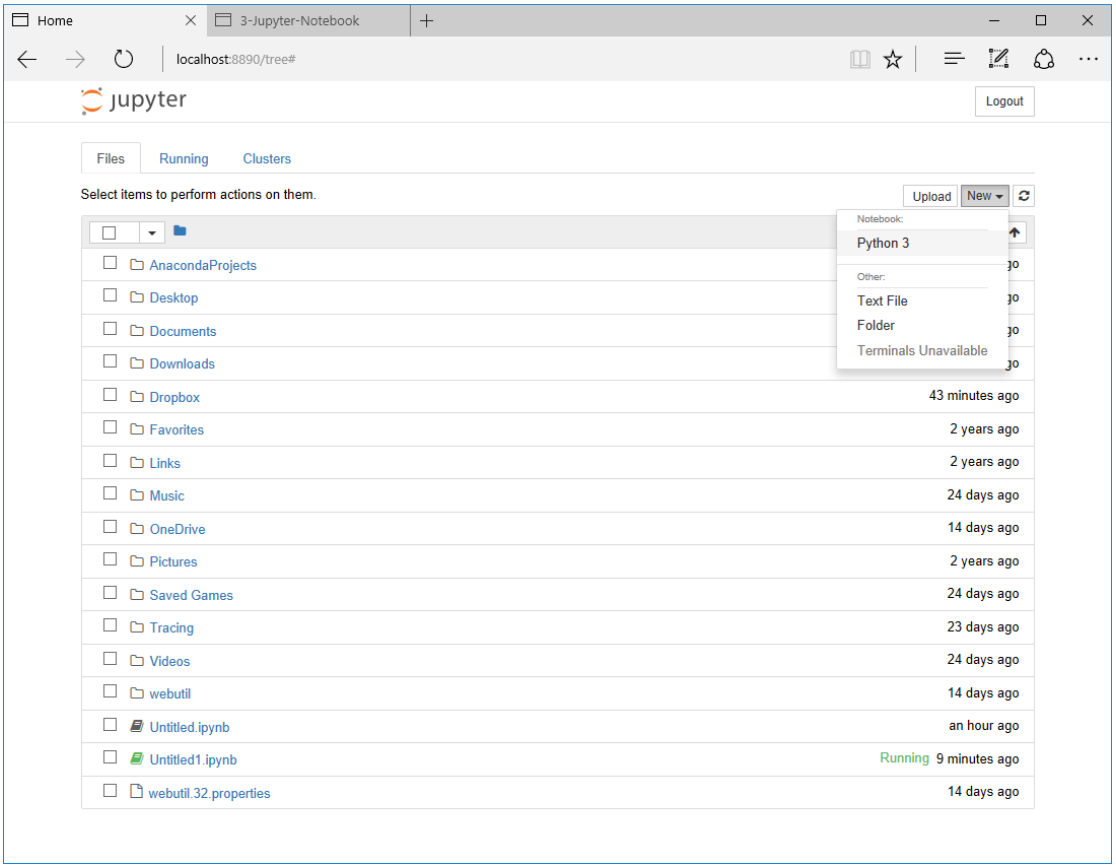
```
Jupyter Notebook is running at: http://localhost:8888/
```

Therefore if a new tab or page was not opened automatically, or you have closed it, type this address into your browser and hit Enter to navigate to the index page. Please note that if the default port (8888) is occupied by another service then Jupyter Notebook will start to serve another port and the url you see on the terminal screen will have a different port.

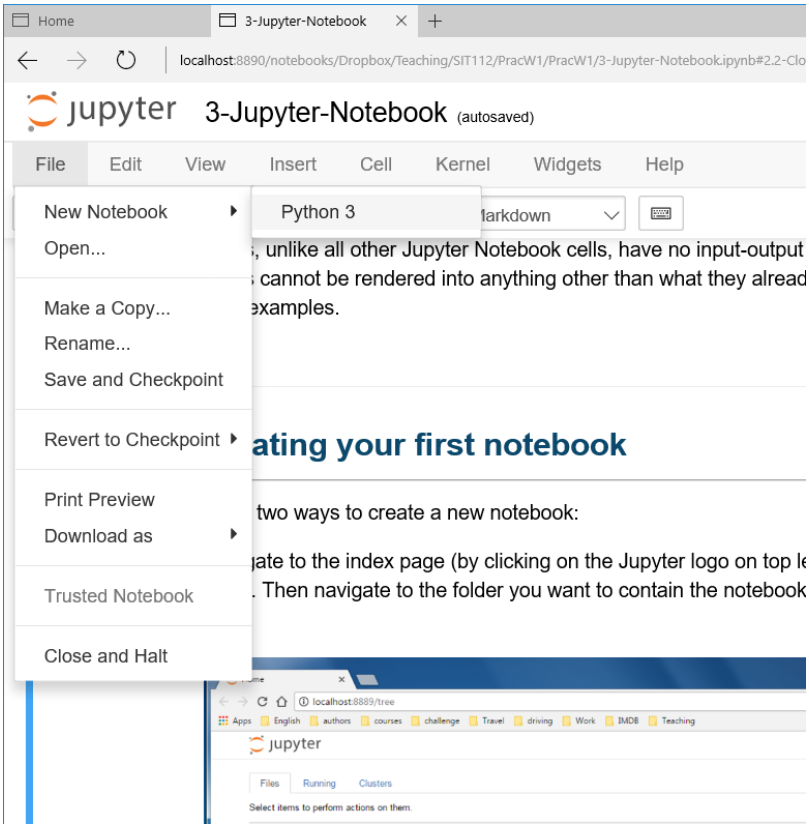
Creating your first notebook

There are two ways to create a new notebook:

1. Navigate to the index page (by clicking on the Jupyter logo on top left of the page or entering the URL). Then navigate to the folder you want to contain the notebook. `Click on New -> Python 3`

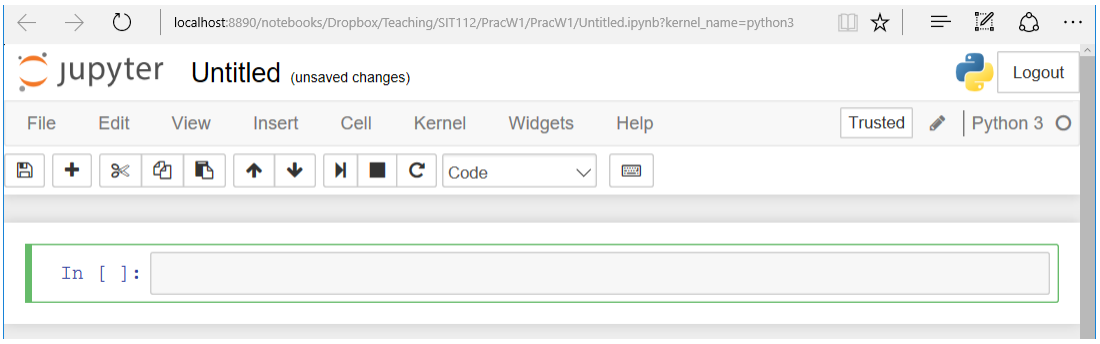


2. Or when you have a notebook open you can also create a new notebook from the menubar select File -> New Notebook -> Python 3.



The (*.ipynb) file will be saved in the directory you are currently using

Now you should have a notebook similar to the image below.

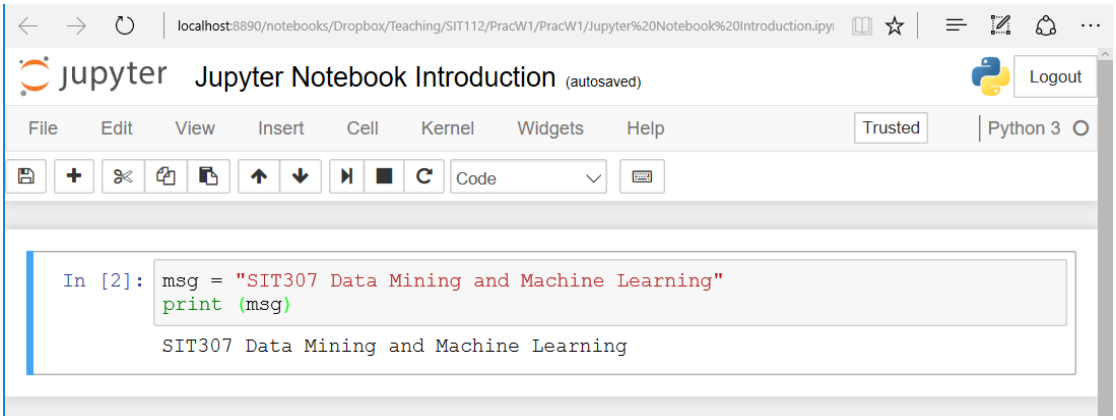


Click on the notebook name and rename it. Copy the code below and paste it into the first cell of Hello_World notebook.

```
msg = "SIT307 Data Mining and Machine Learning"
print (msg)
```

Now to run the code that is in the code cell simply hold down `Ctrl` and press `enter`.

Remember to execute a cell you have to select it and press **`Ctrl + Enter`** or **`run cell`** from the toolbar.



Opening a notebook

To open a notebook once created you can either navigate to where your Jupyter notebooks are stored (with *.ipynb extension) and open an existing notebook by clicking on its name. Or while starting the server, pass the path to the folder where you have stored your notebooks:

```
jupyter notebook C:\Dropbox\my_notebooks
```

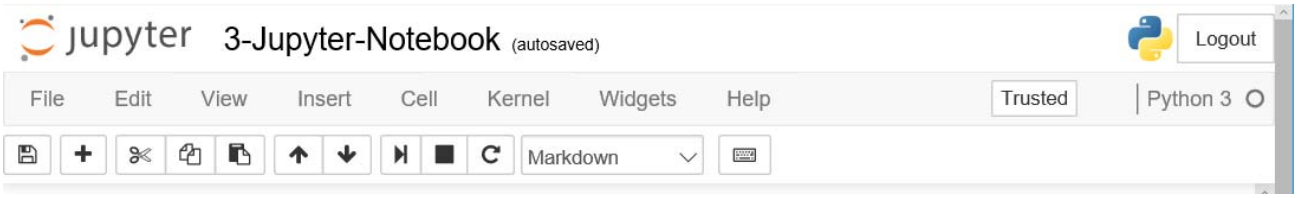
So Jupyter starts from the specified directory and then click on the note book you want to open.

Closing a notebook

Closing the tab or browser window will not close the notebook. If you check the index page after closing the tab, you will see that the page you have closed is still running. It means you can re-open the notebook and have access to the workspace (all variables and objects) as if you have not left the notebook at all. To close (shut down) a notebook, use File -> Close and Halt.

Notebook's Toolbar

Spend some time to make yourself familiar with the toolbar and menubar.



Click on `Help -> User Interface Tour` to learn about the interface

Click on `Help -> Keyboard Shortcuts`

Click on `Help -> Notebook Help` and look at the tutorials there. Make sure to read at least the first four notebooks

Click on `Help -> Markdown` to learn about how to use Markdown for text cells in notebooks

Notebook cells

Each notebook consists of different cells which act as containers for you code and text.

There are essentially three kinds of cells in your Jupyter notebook: Code Cells, Markdown Cells and Raw Cells.

Code cells

By default, Notebooks' cells are code cells and will execute Python. Jupyter Notebooks enerally also support JavaScript, Python, HTML, and Bash commands.

Markdown cells

In Jupyter Notebooks, Markdown Cells are the easiest way to write and format text.

Raw cells

Raw Cells, unlike all other Jupyter Notebook cells, have no input-output distinction. This means that Raw Cells cannot be rendered into anything other than what they already are. They are mainly used to create examples.

Further reading

Jupyter Notebook Tutorial: The Definitive Guide

IPython

Jupyter project

Jupyter Notebook Documentation