CENGN

Jupyter Notebook Guide

Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to easily edit and run code. Jupyter Notebook provides an easy-to-use, interactive data science environment compatible with Python and many other languages.

The Jupyter name is derived from Julia and Python and R, the core languages supported by Jupyter.

Modes

Jupyter Notebook provides a modal environment with two modes: **edit** and **command.**

Edit Mode

- A cell that is in edit mode appears with a green outline.

- To enter edit mode, click on a BLUE cell or press **Enter**.
- To exit edit mode, press **Esc.**

As the name implies, when in this mode, you can edit the cell content and run the code in the cell.

Note that running the cell will take you back to command mode.

Useful Keyboard Shortcuts

Keys	Description
Tab	Tab completion
Shift+Tab	Tooltips, documentation
Esc	Exit to command mode
Ctrl+Enter	Run a cell

CENGN

Command Mode

- A cell that is in command mode appears with a blue outline.

- To enter command mode, press **Esc**.

Command mode gives you access to a lot more functionality including changing cell type, creating, moving, copy-pasting, splitting, deleting cells, etc.

Useful Keyboard Shortcuts

Note: Those commands are not case-sensitive.

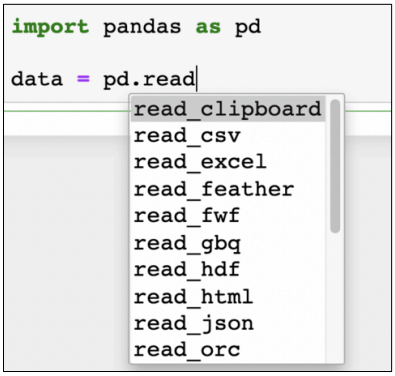
Keys	Description
H	Help, keyboard shortcuts
A	Create a cell above
B	Create a cell below
DD	Delete a cell
M	Change cell to markdown
Y	Change a markdown cell to a code cell
C	Copy
V	Create a new cell below and paste the copied content
X	Cut
Up/Down arrows	Navigate between cells

General Notes

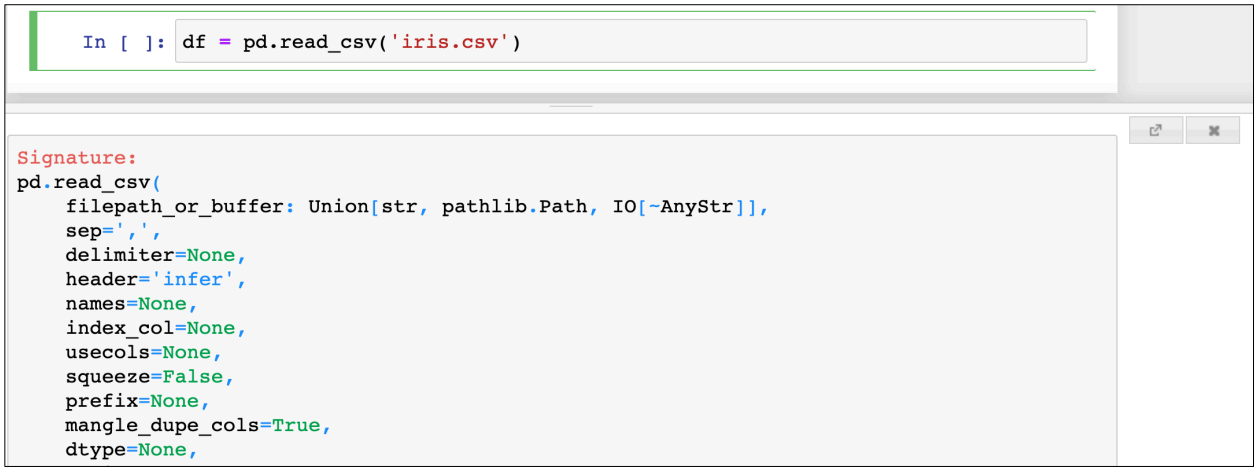
- Before you begin with each lab, it is recommended to create a checkpoint of the notebook, thus you can revert back to that checkpoint should you accidently delete part of the code.
- To create a checkpoint, click on File/Save and Checkpoint. Then to revert back, click File/Revert to Checkpoint.

- You can use the **IPython help** commands:
  - **?** after method, function, etc. to pull the documentation.
  - **??** after method, function, etc. to pull the source code.
- When doing the labs, you should **run all cells in order**. Skipping steps might lead to errors.
- Use your mouse to copy and paste (right-click menu), rather than Ctrl-C or Ctrl-V.
- The last statement in the cell will be evaluated and displayed on screen.

- Code completion offers choices for methods and variables that you may like to use. To invoke code completion, press **Tab** after establishing an assignment, that is, after assigning a value to a variable. For example: If you type ***data=pd.read*** and press **Tab**, you will get all possible pandas methods that you start with read.



- To better understand the code, it is very helpful to pull out the documentation of methods and functions known as tooltips. You can do that using the **Shift+Tab** and **Shift+4Tabs** shortcuts.



- If you would like to experiment with a code snippet, it is recommended that you press C then V while in command mode (blue cell), this will create a duplicate cell below and preserve the original code.

4. Accessing Your Jupyter Notebook Node

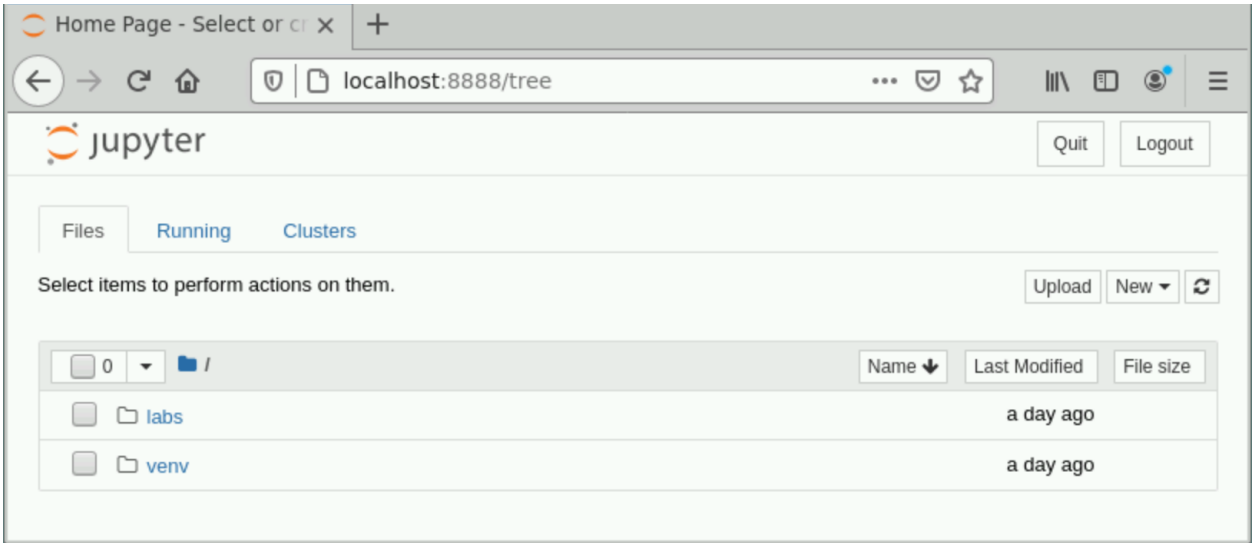
1. Go to https://mylab.training.cengn.ca/#/ and use the myLab credentials provided to you in the enrollment email to log in to your myLab environment. (The myLab service uses two-factor authentication for enhanced security. You will need to use Google authenticator.)

2. Once logged in, go to All Connections, then click on the "+" sign associated with your connection, then click on your ml-node to get access to your node.

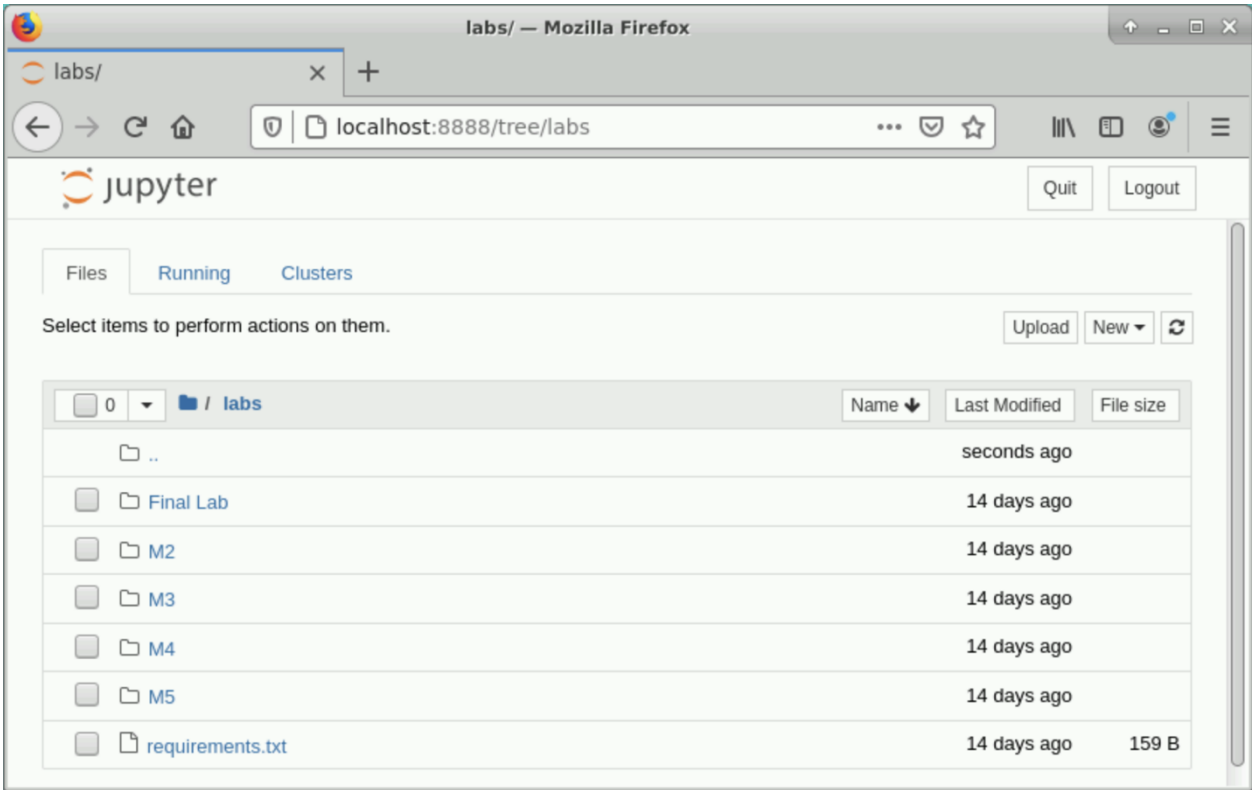
3. Once in your node, open a terminal and enter the following:

```
$ cd ~/ml
$ source venv/bin/activate
$ jupyter notebook #Jupyter notebook will open in the browser:
```

Jupyter Notebook will open in a browser as shown below:



'**labs'** folder contains all the notebooks and datasets that you will need for this course:

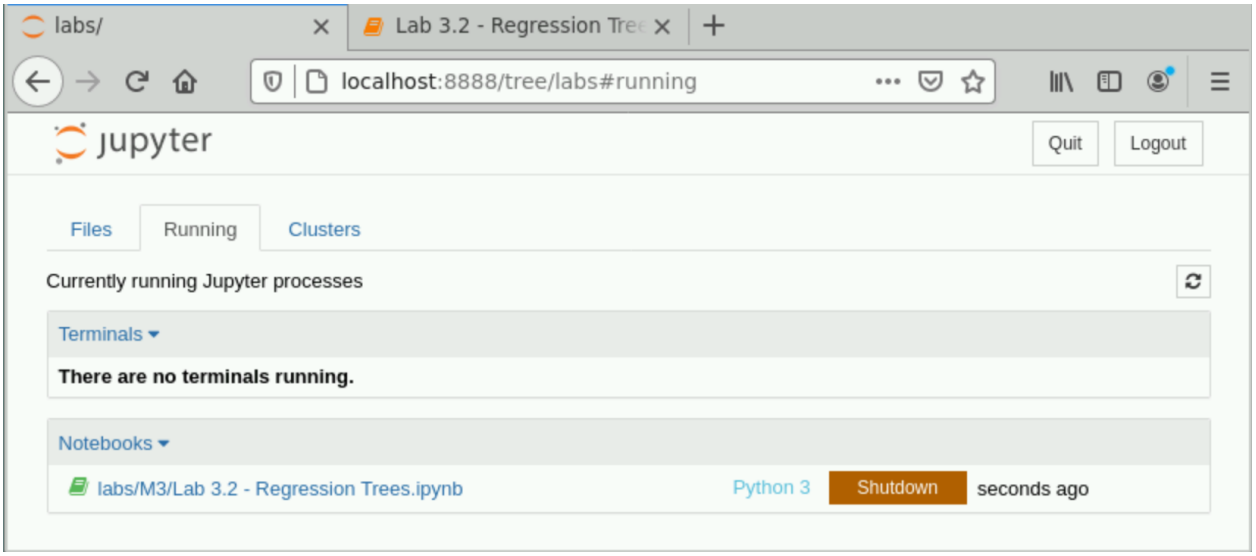


Folders are named by the course module numbers.

Below is a breakdown of the notebooks:

Module#	Notebooks
M2	Consists of one empty notebook and the relevant dataset: Use this notebook to practice any code from Module 2 - Python for Machine Learning.
M3	Consists of five notebooks and relevant datasets. Lab 1 - Linear Regression.ipynb Lab 2 - Regression Trees.ipynb Lab 3 - Logistic Regression.ipynb Lab 4 - Support Vector Machines.ipynb Lab 5 - Naive Bayes.ipynb
M4	Consists of five notebooks and relevant datasets. Lab 1 - K-means Clustering.ipynb Lab 2 - K-means Clustering_Image_Compression.ipynb Lab 3 - Hierarchical Clustering.ipynb Lab 4 - Mixture Models.ipynb Lab 5 - Apriori Algorithm.ipynb
M5	Consists of one notebook. Lab 1 - PCA and t-SNE
Final Lab	Consists of one notebook, which has the instructions for the lab, and a dataset.

Once you complete a lab, switch to the running tab and shutdown the notebook.



4.1. **Magic Functions**

Magic functions are enhancements provided by IPython intended to solve certain problems. They perform behind-the-scenes setup for IPython to work properly.

Magic Code	Description
%%time	Print the time taken to run a cell.
%matplotlib inline	Set the backend of Matplotlib to inline backend. Thus, Matplotlib plots will be displayed in the browser, rather than showing only the text output.
;	Hide the output of a cell when insert at the end of a cell code.