

# Tutorial on Installing Anaconda, Jupyter Notebook, Python and the Markdown Language

This tutorial is an introduction to installing Anaconda with Jupyter Notebook, running a Python code, and using the Markdown language, for the purpose of this course. The above software is open-source.

## 1 Environment for running Python and Jupyter Notebook

We will work with **Anaconda 2023.09** (64-bit) and **Python 3.12** (note that Python 3.11 will work as well). Anaconda installation includes Jupyter Notebook local environment as well as other scientific libraries (NumPy, SciPy, Matplotlib, ...) used in this course. You can install Anaconda with Jupyter Notebook on your own computer.

Alternatively, Jupyter Notebook is available online on the Canadian universities' computation service <https://ucalgary.syzygy.ca/>. You can also access Google Colab by going to <https://colab.research.google.com/> and use it online as well.

Python can also be installed alone (see <http://www.python.org/>), but you will need to install multiple packages for data processing required in this course. It is best to install the whole environment called Anaconda.

Anaconda is a free open-source distribution of the Python and R programming languages for data processing, analytics, and overall scientific computing. It enables the whole package management and deployment, including image processing and machine learning tools required for many applications.

The Anaconda distribution is available on <https://www.anaconda.com/download/> for either Windows, Mac or Linux. We will work with Anaconda 2023.09 (64-bit) and Python 3.12. To install Anaconda, follow the wizard guide.

To run Jupyter Notebook, find “Anaconda3 (64-bit)/Jupyter Notebook” entry in the main menu of your operating system. It will open a console window and later your browser will be activated with a local webpage with Jupyter. Now you can run and modify the Notebook.

To close the Notebook without any losing your data, follow the menu: **File** → **Close and Halt**. On the Jupyter tab, click **Quit**, and the Jupyter console window will close.

## 2 Jupyter Notebook

Jupyter Notebook (formerly known as IPython Notebook) is an interactive web-based programming platform<sup>12</sup>. Everything that is available in the Jupyter console can be used and saved in files called Notebooks. Jupyter Notebook runs on your web browser. When active, the application should look similar to the one shown in Figure 1.

Using the Jupyter dashboard, you can control the environment. It shows which files are in the current folder and which notebooks are running. You can perform basic file operations (renaming, deleting, etc.) The tab *Running* offers a detailed information regarding the notebooks in execution. The button *Logout* and the tab *Clusters* allow for running the Jupyter on a server machine (the server operation is not covered by this document). Clicking the *New* button, you can create new files or folders, as seen in Figure 2.

<sup>1</sup><https://jupyter.org/>

<sup>2</sup>[https://en.wikipedia.org/wiki/Project\\_Jupyter](https://en.wikipedia.org/wiki/Project_Jupyter)

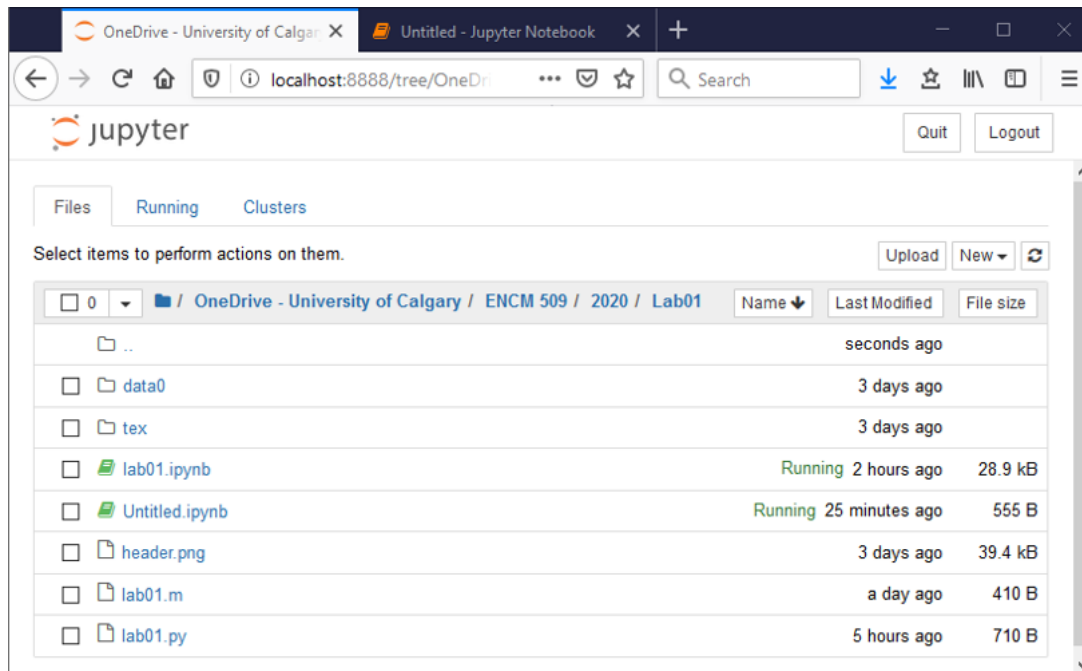


Figure 1: Jupyter Dashboard.

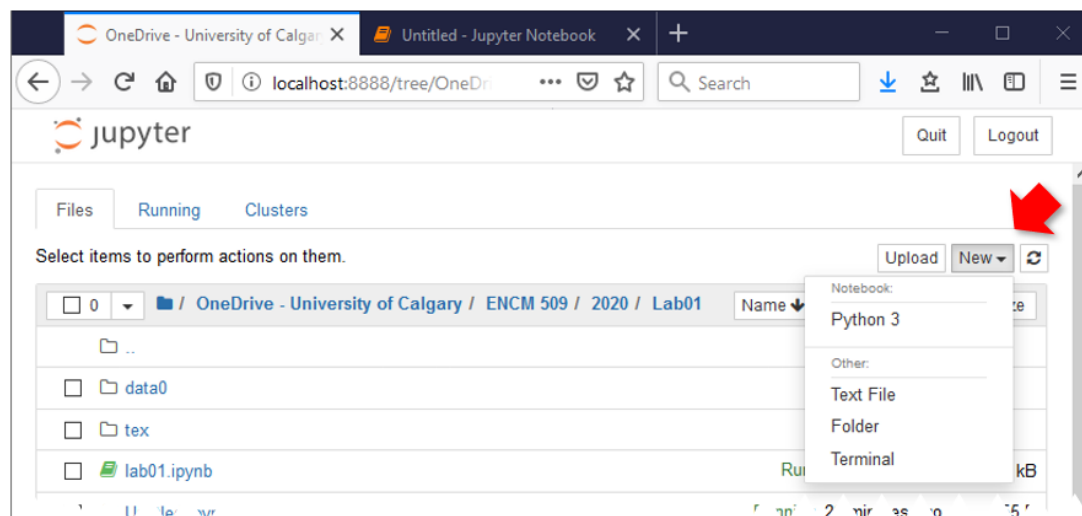


Figure 2: Button *New* activated in the Jupyter Notebook dashboard.

By choosing to create a new *Python 3* notebook, a new screen or tab will open in the browser. This new window is similar to the one shown in the Figure 3, where some elements are highlighted in red and described below.

**Notebook's name:** you can define the name of your final notebook file. By default, this name is set to *Untitled*, which means the file saved in the directory is *Untitled.ipynb*. The extension of notebook files is *.ipynb*. The file name can be changed by clicking on the current name, and choosing the new one, as shown in the Figure 4.

**Menu bar and Toolbar:** the menu bar allows you to access several operations such as inserting, running, exclusion of cells; copying, saving, creating, exporting and converting notebooks. Some of these basic functions can be found in the menu bar.

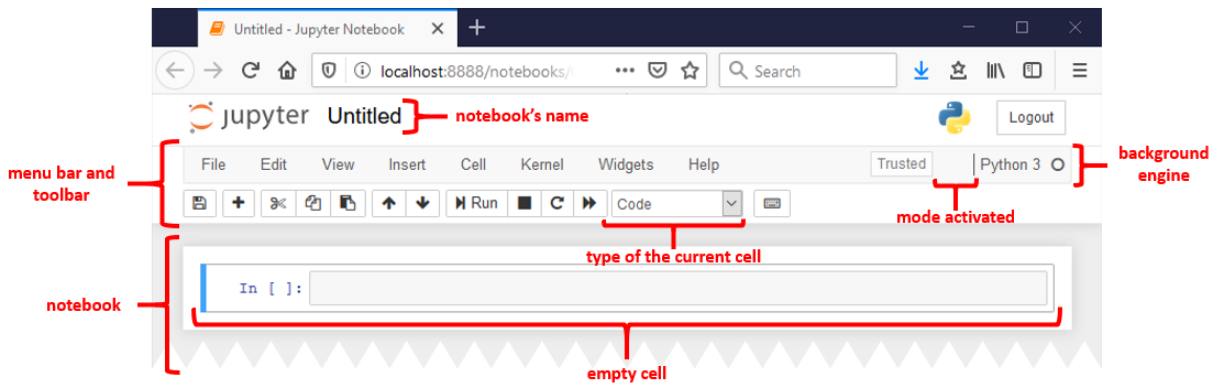


Figure 3: Jupyter Notebook.

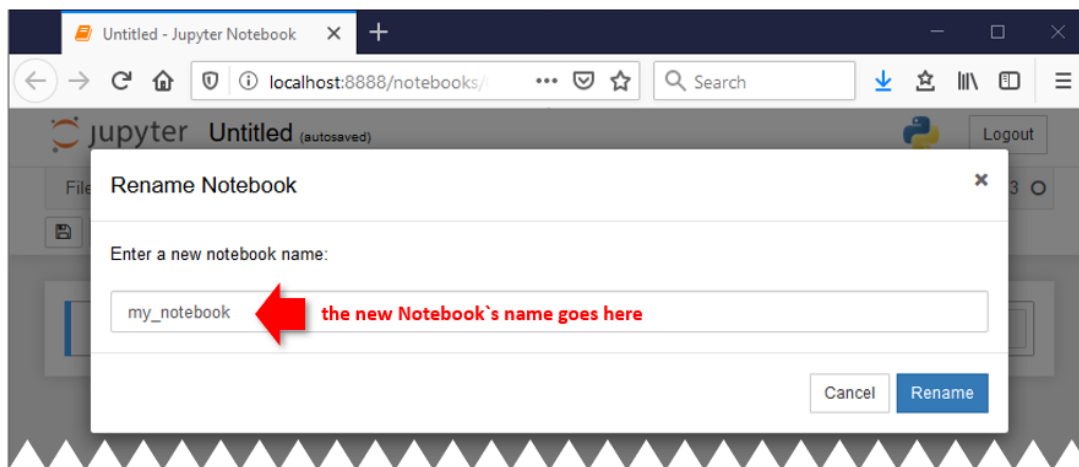


Figure 4: Renaming a notebook.

**Mode activated:** the Jupyter notebook has two modes: *editing* and *command*. When the *editing* mode is activated, you shall see a pencil icon on the space pointed in the Figure 3. In this mode, you can type code or text. If there is no pencil icon in the menu bar, you are in *command* mode, that allows you to interact with the notebook using the keyboard shortcuts (see Section 2.1 for more details on available commands).

**Background Engine:** Since Jupyter supports several programming languages, it shows which engine is being used for this particular notebook.

**Notebook:** is a collection of cells. Each cell, including the outputs, will be saved in the `.ipynb` file.

**Cell:** is the place to put your code and text. Each cell has a type corresponding to an engine.

**Type of Cell:** cells can allocate Python code or texts with formatting using the Markdown language, which is covered in Section 3. In order to choose which one to run (Python code or Markdown), you have to change the cell type. The default type is “Code”, thus, it is expected that you insert a Python code there. Otherwise, you will get an error when running it.

## 2.1 Keyboard navigation

The notebook can be also controlled by shortcuts. Table 1 lists some of those. A complete list can be accessed off-line using the menu **Help** → **Keyboard Shortcuts** of your notebook, or on-line in Jupyter Notebook documentation page<sup>3</sup>.

Table 1: Basic keyboard shortcuts.

| Command              | Expected result  |
|----------------------|--|
| [enter]              | enter in the editing mode of the selected cell.                              |
| [shift] + [enter]    | runs the cell (editing mode).  |
| [up]/[k], [down]/[j] | changes the cell selected (command mode).                                    |
| [s]                  | saves the notebook (command mode).   |
| [y], [m]             | changes the cell type: [y] for code and [m] for Markdown (command mode).     |
| [a], [b]             | creates a new cell, [a] above or [b] below the selected cell (command mode). |

## 3 Markdown Language

Markdown is a markup language created by John Gruber and Aaron Swartz, and released in 2004. It is similar to  $\text{\LaTeX}$  and HTML. Note that there are several versions of Markdown: GitHub Markdown and the Jupyter Notebook Markdown. They have many commands in common, however, here we focus only on the Jupyter Notebook version.

You can use the Markdown inside the cells (remember: you need to change the cell type first). In a Jupyter Notebook, every single output which is a result of running the Markdown cell, is statically embedded into the .ipynb file. However, the linked images (for example, in .jpg or .png format) need to be located in the same directory as the Notebook file.

Below, we provide few examples. For a more complete documentation, visit the page of Jupyter Notebook on Markdown cells<sup>4</sup> and the DataCamp<sup>5</sup>.

You will perform exercises with Markdown in Lab 1.



<sup>3</sup><https://jupyter-notebook.readthedocs.io>.

<sup>4</sup><https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html>

<sup>5</sup><https://www.datacamp.com/community/tutorials/markdown-in-jupyter-notebook>

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Python code:

```
def f(x):
    return x**2
```

C-code:

```
for (i=0; i<n; i++) {
    printf("hello %d\n", i);
    x += 4;
}
```

Python code:

```
```Python
def f(x):
    return x**2
```
```

C-code:

```
```C
for (i=0; i<n; i++) {
    printf("hello %d\n", i);
    x += 4;
}
```
```

$$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$$

$$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$$

| Header 1 | Header 2     | Header 3      |
|----------|--------------|---------------|
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |

| Header 1 | Header 2     | Header 3      |
|----------|--------------|---------------|
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |
| center   | left-aligned | right-aligned |

Unordered (bulleted):

- \* first item
- \* second item
- \* third item

Ordered:

1. first item
2. second item
3. third item

Nested lists:

- \* first item:
  - \* sub-first item
  - \* sub-first item again
- \* second item:
  1. sub-second item
  - \* sub-second item (without number!)
  - \* sub-second item (you just need to put the first number)

Unordered (bulleted):

- first item
- second item
- third item

Ordered:

1. first item
2. second item
3. third item

Nested lists:

- first item:
  - sub-first item
  - sub-first item again
- second item:
  1. sub-second item
  2. sub-second item (without number!)
  3. sub-second item (you just need to put the first number)

! [\[UofC Dino\]](#) (dino.png)



## Acknowledgment

The template for this Lab was developed by Dr. H. C. R. Oliveira (postdoc in the Biometric Technologies Laboratory in 2020-2022). We also acknowledge this course TAs, O. Shaposhnyk, I. Yankovyi and M. Zakir, for verifying this lab code.

---

*Dr. S. Yanushkevich  
Biometric Technologies Laboratory  
January 11, 2024.*