

01.introduction_to_jupyter

November 11, 2022

0.0.1 Jupyter installation instructions

In this course, we will use Python in Jupyter notebooks (a shorthand for **Julia**, **Python** and **R**). To get and use Jupyter notebooks, we recommend Anaconda distribution, which is available for the most common operating systems. To install Anaconda, check out [this](#) link.

After installing, you can either open the Jupyter via the app, or execute the `jupyter notebook` command in the terminal to start a Jupyter server.

As an alternative to Jupyter, you can use Python IDE [PyCharm](#) or [Visual Studio Code](#).

Side note: *There are many [other](#) available kernels.*

Why Jupyter Notebooks?... Jupyter is a user-friendly, browser-based interactive environment. Notebooks are usefull since they allow not only code execution but also they allow adding text boxes, formulas in $\text{\LaTeX}$, pictures, clickable links and many more. Notebooks can be saved to executable formats, such as `.py` and `.ipynb`, or to non-executable ones, such as `.pdf`, `.html`, `.tex`, etc.

0.0.2 The Jupyter notebook interface

Name of the current notebook (`01.introduction_to_jupyter`) is displayed in the top left corner of the page. You can edit it by either clicking on in the Jupyter files page by selecting the file and selecting the “Rename” option. Right next to the file name you may see the time of the last checkpoints (the last save) – make sure that it’s recent, so all of your changes are in place! You can save the notebook by either pressing `ctrl + s` (Windows), `command + s` (OS) or by clicking on the disk icon right below the “File” menu. Often the notebook is saved automatically with some periodicity.

The “File” menu contains several options, including download option for the notebook, reversion to the previous checkpoint and a few other options. The “Edit” menu contains a few usefull options including “Undo delete cells” and “Find and Replace” option that allows finding a pattern in a cell and replacing it with a custom element. Other useful options include merging, moving (can be done with the arrow up and down signs) and splitting cells. New cells can be added by pressing the plus sign `+` in toolbar, between the File and Edit menus. To delete a cell, press the scissors sign. Finally, the “Kernel” menu allows controlling the Python kernel by interrupting the process(es) if needed, restarting or shutting down the kernel, and even changing the kernel to another one (if available).

Jupyter notebook has a wide range of shortcuts and snippets that allow writing and manipulating the code much faster than without them. To learn more about the shortcuts, please see [this](#) link.

The major types of cells in Jupyter notebooks are the code cells (for writing code in Python) and markdown cells (for writing text). When a new cell is created, it is a code cell (indicated as “Code” in the toolbar). To convert it into the markdown cell, select “Markdown” from the expandable list of options.

0.0.3 Markdown Cells

Markdown cells allow writing text and formatting it in many different ways.

```
[1]: # This is the cell for writing code
    5 + 5
```

[1]: 10

To run a code cell, or render a markdown cell, either press a **shift + enter** combination, or click the “Run” button in the toolbar. To introduce changes into the rendered cell, double click on the respective cell.

Headings are indicated with hash symbols, for example: # Large heading ## Smaller heading

When writing the text, you may want to highlight it in **bold** or *italics* format, you may also want to include some commands in Python, for example `5+5` or `import numpy as np`, or even formulas such as $c^2 = \sqrt{a^2 + b^2}$. Longer expansions can be written in the new line with the following syntax:

$$c^2 = \sqrt{a^2 + b^2}.$$

Other than that, for explaining the steps of your algorithm, you may want to use enumerated lists, such as 1. Step 1 is this 2. Step 2 is that

- a) the first substep
- b) the second substep.

Alternatively, you may use bullet points: * Step 1 is this * Step 2 is that

Jupyter notebook allows inserting images as well:

To know more about the markdown syntax and tricks, check out [this](#) link.

The cell can be in one of the two interaction modes:

- **Command Mode** (cell highlighted blue) - allows selection of multiple cells by pressing **shift** and hitting up/down arrows,
- **Edit Mode** (cell highlighted green) - editable cell

0.0.4 Magic Commands

Magic commands are useful since they provide a wide range of functionality that might be out of the Python scope. To check out the full list of magic functions, visit [this](#) link, or use the following magic command:

```
[2]: %lsmagic
```

[2]: Available line magics:

```
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark  
%cat %cd %clear %colors %conda %config %connect_info %cp %debug %dhist  
%dirs %doctest_mode %ed %edit %env %gui %hist %history %killbgscripts  
%ldir %less %lf %lk %ll %load %load_ext %loadpy %logoff %logon  
%logstart %logstate %logstop %ls %lsmagic %lx %macro %magic %man  
%matplotlib %mkdir %more %mv %notebook %page %pastebin %pdb %pdef %pdoc  
%pfile %pinfo %pinfo2 %pip %popd %pprint %precision %prun %psearch  
%psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx  
%reload_ext %rep %rerun %reset %reset_selective %rm %rmdir %run %save  
%sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext  
%who %who_ls %whos %xdel %xmode
```

Available cell magics:

```
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript  
%%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2  
%%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit  
%%writefile
```

Automagic is ON, % prefix IS NOT needed for line magics.

With some magic functions one can convert the cell functionality. For example, bash magic function:

```
[3]: %%bash  
ls
```

```
01.introduction_to_jupyter.ipynb  
02.quick_python_overview.ipynb  
03.useful_python_packages_and_examples.ipynb  
04.python_hands_on_genetics.ipynb  
iris_dataset.csv
```

Congrats! Now you are familiar with the basics of Jupyter, you may continue with the exercises!