

Guide to setting up, using and editing Jupyter Notebooks

Setup of Jupyter:

On terminal PC's

- Start up the computer, use the Ubuntu OS (Windows is not supported)
- Log in using your Science login
- Open a console (CTRL + ALT + T)
- Type: `gedit ~/.profile` and press enter (if that doesn't work you can type and execute: `source ~/.profile`)
- In the open gedit window, add the following line to the bottom of your file (Note: your file may be empty): `export PATH=/opt/anaconda3/bin:$PATH`
- Relog on your computer (use a reboot if necessary)
- Reopen a console (CTRL + ALT + T)
- Type `"spyder"` to open the Spyder IDE or `"jupyter notebook"` to start a notebook
- Navigate to your assignment folder and open the .ipynb Jupyter Notebook

Note that these operations will add the Anaconda distribution to the search path. This means that some standard functionality may be overwritten. After the course is done, or when you run into trouble during other courses which may need Python, simply remove the line you added to the `.profile` file, and relog/reboot.

On your own computer

- Install Anaconda (take the Python 3.X distribution). It can be downloaded from: <https://www.anaconda.com/products/individual>
- Make sure to install the entire package!
 - For Linux: answer yes to *"Do you wish the installer to prepend the Anaconda install location to PATH in your /home/%username%/.bashrc"*
Only do this if you do not already have a Python installation, or if you are absolutely aware of the possible consequences and side-effects.
- Open a console
 - On Windows: If you set added Anaconda to PATH during installation, this works from CMD, otherwise you will have to use the Anaconda Prompt)
- Type `"jupyter notebook"` and press enter. A notebook will open in your browser
- Navigate to your assignment folder and open the .ipynb Jupyter Notebook
 - You can only move to subfolders from the path from which you typed `"jupyter notebook"`!

Using a variable inspector for Jupyter notebooks

It is often useful to be able to easily view properties of the variables in your workspace. Often this is done easily from an IDE such as Spyder (included in Anaconda), or PyCharm. You are welcome to use these to develop your methods, as long as you ensure that the notebooks still contain executable (and executed!) code, and all answers to the questions.

It might sometimes be easier to inspect your variables directly from within your notebook, this saves a lot of time spent copying and rerunning your code. You can achieve this by installing a Jupyter Notebook extension. You can do this by running the following commands from within your console:

- `conda install -c conda-forge jupyter_contrib_nbextensions`
- `jupyter contrib nbextension install --user`
- Then start your notebook using: `jupyter notebook`
- Then go to the nbextensions tab, which should show in your starting screen
- Enable the “Variable Inspector” Notebook extension.
- After clicking the extension, you should see a small information screen. Read the information thoroughly, and beware of some of the limitations on the possible datatypes. (You should not encounter any problems with most datatypes used during the course.)
- Using the instructions from the information screen you should be able to inspect your variables!

Using and editing Jupyter notebooks

Jupyter notebooks are a nice way of combining a report and the code used to produce results. A notebook has several types of cells:

- Markdown cells
- Code cells

In Markdown cells you can type and edit text using the Markdown makeup language (<https://guides.github.com/features/mastering-markdown/>). You can simply edit a Markdown cell by double clicking it, editing the text, and pressing shift+enter when you are done editing.

Code cells are able to execute code in console style. Our Jupyter Notebooks use iPython with Python 3.6, although other languages are possible for other notebooks. Cells are not executed automatically, nor are they executed in a top-down manner, you have to execute them manually, and the order matters. To execute a cell, press shift+enter, this will run the code. Until the notebook kernel is shut down, all variables will remain! Take care to not overwrite variables when you don't want to!

Some nice tricks:

- You can add a cell by using Insert->Insert Cell Above/Insert Cell Below
- You can convert between cell types using Cell->Cell Type
- You can clear all outputs by using Cell->All Output->Clear
- You can remove cells using Edit-> Delete Cells
- You can save your notebook by pressing CTRL+S
- You can run multiple cells using Cell->Run All/Run All Above/Run All Below