



HAUTE ÉCOLE  
D'INGÉNIERIE ET DE GESTION  
DU CANTON DE VAUD  
[www.heig-vd.ch](http://www.heig-vd.ch)

# TAL Practical Work 1a : Setting Up the Environment

## Summary

The goal of this first session of the first lab is to get you introduced to an environment that will enable you to run experiments in text analysis for the [TAL course](http://iict-space.heig-vd.ch/apu/cours-tal/) (<http://iict-space.heig-vd.ch/apu/cours-tal/>). You will use the [Python 3](https://www.python.org/) (<https://www.python.org/>) programming language, with [Jupyter](https://jupyter.org/) (<https://jupyter.org/>) notebooks to write instructions and log your results.

In the second and third sessions (separate notebooks 1b and 1c), you will use the [NLTK](http://www.nltk.org/) (<http://www.nltk.org/>) toolkit to get texts from the web or local files, and segment (split) them into sentences and words.

To submit your practical work to the teacher, please execute all cells of your Jupyter notebook, then print it as a PDF document, and [send it by email](mailto:andrei.popescu-belis@heig-vd.ch) (<mailto:andrei.popescu-belis@heig-vd.ch>).

## Welcome to the first lab of the TAL course !

To see this text in your browser, you have two options. In both cases you need to download Jupyter notebook files from the [TAL website](http://iict-space.heig-vd.ch/apu/cours-tal/): (<http://iict-space.heig-vd.ch/apu/cours-tal/>) this file is `TAL_Labola.ipynb`.

### First option

You can use a free online environment provided by Google, named [Colab](https://colab.research.google.com) (<https://colab.research.google.com>). You will need a Google account (the same as Gmail if you are using it), and your files will be stored on Google Drive. Once you open a session in Colab, you will need to import the Jupyter notebooks to Colab. This environment is offered by Google for machine learning experiments, with CPU/GPU/TPU, but not for time-consuming jobs. All Python packages that we will use are already installed.

### Second option

With a local installation you will be able to work offline, and gain experience with Python and Jupyter.

- You need first to install [Python 3](https://www.python.org/downloads/) (<https://www.python.org/downloads/>) (Python 3.6, not 2.7).
- Install then the [Jupyter package](https://jupyter.org/install) (<https://jupyter.org/install>).
- A good idea is to use the [Package Installer for Python \(pip\)](https://packaging.python.org/tutorials/installing-packages/) (<https://packaging.python.org/tutorials/installing-packages/>) to add new packages.
- All these operations are done in the command line (on Windows: `cmd` or Power Shell).
- The `TAL_Labola.ipynb` file must be saved into your repository of Jupyter notebooks.
- You start Jupyter with `jupyter notebook --no-browser`, open the Jupyter interface at <http://localhost:8888> (<http://localhost:8888>) and launch this notebook.

## Using Conda environments (optional)

Before installing Python and Jupyter locally, you may consider using Conda to manage different versions of Python and of other modules on your computer. You may wish to install Conda before Python, but you can also do it after it. In this case, Conda will install its own Python libraries and versions, depending on the environment.

Conda offers an elegant solution to manage your environments, especially when you need to manage several versions of Python and/or some of its components, for instance for several courses. Conda is a minimal environment manager, while Anaconda is a larger software distribution that includes also Python and R software.

## Instructions to use the Conda package & environment manager

- To install Conda, follow the instructions at <https://conda.io/projects/conda/en/latest/user-guide/install/index.html> (<https://conda.io/projects/conda/en/latest/user-guide/install/index.html>) and consider installing Miniconda. Follow the instructions for your OS: Windows, Linux or MacOS.
- You will be prompted to get a suitable installer from <https://conda.io/en/latest/miniconda.html> (<https://conda.io/en/latest/miniconda.html>) (an executable for Windows, or a `bash` script for Linux).
- In Linux, run `wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh` and then `bash Miniconda3-latest-Linux-x86_64.sh`

After you install it, note that all Conda commands (most importantly: creating and loading an environment) will be performed from the command line. In Windows, it could be the command prompt or an Anaconda Prompt. The Conda commands are the same on all OSs. Try `conda info` for a start.

- Use the following instructions to get familiar with Conda: <https://conda.io/projects/conda/en/latest/user-guide/getting-started.html> (<https://conda.io/projects/conda/en/latest/user-guide/getting-started.html>)
- Create an environment for the TAL course and activate it, for instance like this:
  - `conda create --name CoursTAL python=3.6`
  - `source activate CoursTAL`
  - you can try `conda list` to see what is installed in this environment
  - to exit from (deactivate) the environment, type `source deactivate`
- Useful cheatsheet of conda commands: <https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html> (<https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html>)

## Instructions to use Jupyter with Conda

Jupyter supports the use of Conda environments. It can run a notebook in a kernel associated to a specific environment. First of all, Jupyter must be installed (not necessarily in the same environment, but doing it in the same one may make you life easier).

Once installed, run these commands:

- `source activate CoursTAL`
- `python -m ipykernel install --user --name CoursTAL --display-name "CoursTAL"`

Once you have done this, you can just choose this kernel (environment) when you open a new Jupyter notebook. It is also possible to switch kernels from within Jupyter.

## Running some simple tests

To test your settings, please compute in the next cell how many 3-letter strings exist, given the 26 letters of the English alphabet.

```
In [1]: # Write your computation below this line, then execute the cell to obtaine and display the result.
```

One more test: please create a list with the first ten numbers (1 to 10) and then print the list of their squares. If needed, use inspiration from this [UEdin tutorial \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python\\_intro.ipynb\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python_intro.ipynb) on Python.

```
In [2]: # Write your code here, to display (or return) the list of the first ten squares from 1.
```

## Getting more familiar with Python

If you need to get familiar with Python, one option is to go through [this notebook \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python\\_intro.ipynb\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python_intro.ipynb) from the [University of Edinburgh NLP course \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/). (They also offer a useful [introduction to the numpy package \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/numpy\\_intro.ipynb\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/numpy_intro.ipynb).) The notebook above also gives a very short introduction to [Matplotlib \(https://matplotlib.org/\)](https://matplotlib.org/), which will allow you to plot graphics in Jupyter. A more thorough introduction is provided by J.R. Johansson in his [Scientific Python Lectures \(http://github.com/jrjohansson/scientific-python-lectures\)](http://github.com/jrjohansson/scientific-python-lectures) accompanied by Jupyter notebooks for practicing. And of course the definitive reference is the [Python documentation \(https://www.python.org/doc/\)](https://www.python.org/doc/) itself.

## End of Lab 1a

This lab is not part of the homework, so there is no need to print and submit it. Please move now to the notebook for Lab 1b.