

# Cours TAL - Laboratoire 1a - Mise en place de l'environnement

## Bienvenue au premier labo du cours TAL !

Pour travailler avec ce *notebook*, il faut d'abord télécharger l'archive ZIP du Laboratoire 1 qui inclut ce *notebook* en PDF et les quatre fichiers `TAL_Lab01a.ipynb`, `TAL_Lab01b.ipynb`, `TAL_Lab01c.ipynb` et `TAL_Lab01d.ipynb`. Vous pouvez commencer par lire le fichier PDF avant de mettre en place l'option qui vous conviendra parmi les suivantes, ce qui vous permettra d'ouvrir les fichiers `.ipynb`.

## Résumé

Le but de cette première partie (a) du Laboratoire 1 est de vous permettre d'installer et de tester l'environnement de programmation qui vous servira tout au long du [Cours TAL](https://cyberlearn.hes-so.ch/course/view.php?id=14966) (<https://cyberlearn.hes-so.ch/course/view.php?id=14966>). Le langage de programmation sera [Python 3](https://www.python.org/) (<https://www.python.org/>) avec les *notebooks* [Jupyter](https://jupyter.org/) (<https://jupyter.org/>) qui vous permettront d'écrire des instructions, conserver leurs résultats, et ajouter vos commentaires.

Dans les parties (b), (c) et (d) de ce Laboratoire 1 (*notebooks* appelés `TAL_Lab01b`, `TAL_Lab01c` et `TAL_Lab01d`), vous utiliserez la boîte à outils [NLTK](http://www.nltk.org/) (<http://www.nltk.org/>) pour travailler sur des fichiers locaux ou en ligne, et spécifiquement les segmenter en phrases et en mots.

Pour soumettre vos résultats, veuillez exécuter toutes les cellules de chaque *notebook*, puis regroupez les quatre fichiers `.ipynb` dans un fichier archive `zip`, à soumettre sur [Cyberlearn](https://cyberlearn.hes-so.ch/course/view.php?id=14966) (<https://cyberlearn.hes-so.ch/course/view.php?id=14966>).

## Options d'utilisation de Python et Jupyter

Tout d'abord, décidez si vous avez besoin d'un gestionnaire d'environnements comme Conda -- utile par exemple si vous devez gérer plusieurs versions incompatibles de Python ou de ses *packages* sur une même machine. Si oui, veuillez suivre les instructions de l'option 1 ci-dessous, puis continuez avec l'option 2. Sinon, passez directement à l'option 2. Si vous ne souhaitez pas du tout travailler en local, suivez l'option 3. En résumé :

- travailler sur votre ordinateur ou sur un service en ligne ?
  - ordinateur (recommandé) : *passer à la question suivante*
  - en ligne: *option 3*
- environnement unique ou multiple ?
  - unique (plus simple) : *option 2 seule*
  - multiple (plus flexible) : *option 1 puis option 2*

## Option 1 : gestionnaire d'environnements Conda

Avant d'installer Python et Jupyter sur votre ordinateur, demandez-vous si vous souhaitez installer un gestionnaire d'environnements qui évitera les conflits entre différentes versions de Python et des ses *packages* dont vous pourriez avoir besoin. Vous pouvez installer le gestionnaire Anaconda / Conda / Miniconda avant Python, mais il est toujours possible de le faire après. Si vous choisissez Miniconda3, Python 3 sera inclus dans l'installation. (Anaconda est une distribution plus grande qui contient beaucoup de *packages* Python installés.

### Instructions pour installer Conda

- suivez les instructions à <https://conda.io/projects/conda/en/latest/user-guide/install/index.html> (<https://conda.io/projects/conda/en/latest/user-guide/install/index.html>) et considérez plutôt l'installation de Miniconda3 (attention à l'OS : Windows, Linux ou MacOS).
- les programmes d'installation (exécutable Windows, script `bash` pour Linux) seront à <https://conda.io/en/latest/miniconda.html> (<https://conda.io/en/latest/miniconda.html>)
  - exemple sous Linux : exécuter `wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh` puis `bash Miniconda3-latest-Linux-x86_64.sh`
- après l'installation, toutes les commandes Conda (identiques sur tous les OS) se font en ligne de commande (les plus importantes sont celles pour créer un nouvel environnement et le charger). Sous Windows, on utilise le `command` prompt ou le prompt d'Anaconda.
- commande utile : `conda info`
- introduction très rapide à 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>)
- liste de commandes Conda: <https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html> (<https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html>)
- pour créer un environnement pour le cours de TAL et l'activer :
  - `conda create --name CoursTAL python=3.7`
  - `source activate CoursTAL`
- essayer `conda list` pour voir ce qui est installé par défaut
- pour quitter l'environnement : `source deactivate`

### Instructions pour utiliser Jupyter avec Conda

Jupyter est un serveur qui supporte des pages dynamiques avec des textes, du code, et des résultats (comme celle-ci). On peut demander à Jupyter d'utiliser un certain environnement Conda (comme *kernel*). Installez d'abord Jupyter (voir Option 2), de préférence dans le même environnement ( `CoursTAL` ). Une fois installé, exécutez ces commandes :

- `source activate CoursTAL`
- `python -m ipykernel install --user --name CoursTAL --display-name "CoursTAL"` Ainsi, lorsque vous créez un *notebook* Jupyter, vous pourrez choisir l'environnement "CoursTAL" pour le kernel.

## Option 2: installation locale de Python (sans Conda)

- Installez un interpréteur [Python 3](https://www.python.org/downloads/) (<https://www.python.org/downloads/>) (Python 3.8 ou 3.7, et non pas 2.7).
- Installez ensuite le [package Jupyter](https://jupyter.org/install) (<https://jupyter.org/install>). Une bonne idée est d'utiliser [Package Installer for Python \(pip\)](https://packaging.python.org/tutorials/installing-packages/) (<https://packaging.python.org/tutorials/installing-packages/>) pour ajouter de nouveaux *packages* (en général avec la commande `pip install xxxxx`.
  - Ces opérations se font en ligne de commande (sous Windows: `cmd` ou Power Shell).
- Créez un dossier où vos stockerez vos *notebooks* et ajoutez-y `TAL_Labola.ipynb` et les 2 autres.
- Depuis le dossier précédent (`cd xxxx`), lancez le serveur Jupyter avec la `jupyter notebook --no-browser`
- Ouvrez l'URL <http://localhost:8888> (<http://localhost:8888>) pour voir la liste des *notebooks* connus de Jupyter, puis exécutez celui appelé `TAL_Labola.ipynb` -- celui-ci même !

## Option 3 : travailler sur un service en ligne (Colab)

Vous pouvez utiliser l'environnement en ligne gratuit proposé par Google, appelé [Colab](https://colab.research.google.com) (<https://colab.research.google.com>). Vous aurez toutefois besoin d'un compte Google (par exemple Gmail) et vos fichiers seront stockés sur Google Drive. Colab est offert par Google pour expérimenter avec le *machine learning* (sur CPU, GPU ou TPU), mais pas pour des calculs très intenses.

Une fois votre session ouverte sur Colab, vous devrez importer les *notebooks* Jupyter vers cet Colab. Les *packages* Python que nous utiliserons sont déjà installés.

## Quelques tests simples

Pour tester votre installation, veuillez écrire un simple calcul dans la cellule suivante : combien de séquences de trois lettres peut-on former avec les 26 lettres (minuscules) de l'alphabet latin ? (Sans diacritiques.)

```
In [1]: # Ecrivez votre code à la ligne qui suit, puis exécutez la cellule pour afficher le résultat
```

Autre test : en Python, veuillez créer une liste contenant les entiers de 1 à 10, puis veuillez afficher la liste de leurs valeurs au carré. Vous pouvez vous inspirer de ce [Tutoriel Python de l'université d'Edimbourg](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](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python_intro.ipynb)).

```
In [2]: # Ecrivez votre code à la ligne qui suit, puis exécutez la cellule pour afficher le résultat
```

## Comment se familiariser avec Python?

Comme indiqué ci-dessus, la première option est [le tutoriel de l'UEdin \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python\\_intro.ipynb\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/python_intro.ipynb) qui est proposé dans le [cours de TAL de l'UEdin \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/). Ce cours offre aussi une bonne [introduction au package numpy \(http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/numpy\\_intro.ipynb\)](http://www.inf.ed.ac.uk/teaching/courses/fnlp/Tutorials/numpy_intro.ipynb). Le premier *notebook* introduit aussi [Matplotlib \(https://matplotlib.org/\)](https://matplotlib.org/), qui permet de faire des graphiques dans Jupyter.

Une introduction systématique au langage Python est proposée par J.R. Johansson dans ses [Scientific Python Lectures \(http://github.com/jrjohansson/scientific-python-lectures\)](http://github.com/jrjohansson/scientific-python-lectures), qui sont accompagnées de *notebooks* Jupyter. La [documentation Python \(https://www.python.org/doc/\)](https://www.python.org/doc/) en ligne est également utile.

## Fin de la partie (a) du Laboratoire 1

Veuillez passer maintenant au *notebook* `TAL_Lab01b.ipynb`.