
Fiche TP N°3

Introduction à Python

1. Installer Python Anaconda : le meilleur outil de Machine Learning

Anaconda contient tous les outils et bibliothèques dont vous avez besoin pour faire du Machine Learning : Numpy, Matplotlib, Sklearn, etc. Commencez par télécharger puis installer Anaconda depuis le site officiel :

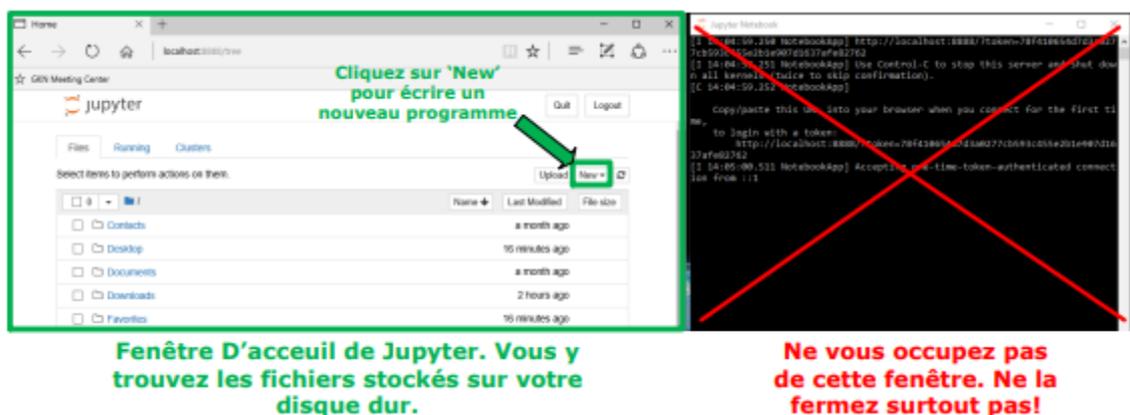
<https://www.anaconda.com/distribution/#download-section>

Note : Téléchargez toujours la version la plus récente de Python.

2. Utilisation de Jupyter Notebook pour vos projets

Jupyter Notebook est une application Web qui permet de créer et de partager des codes Python. C'est une application Web, mais il n'est pas nécessaire d'avoir une connexion Internet pour vous servir de Jupyter. Aussi, vos données/codes ne sont à aucun moment stockés sur Internet.

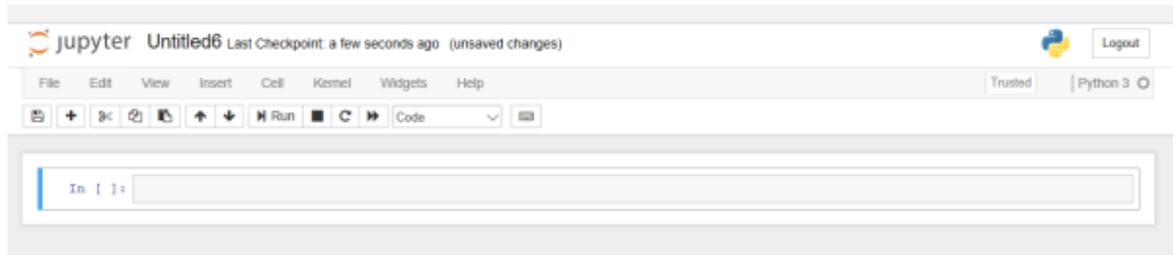
Lorsque vous démarrez Jupyter, il est possible que 2 fenêtres s'ouvrent, auquel cas ne vous occupez pas de la fenêtre noire (la console) et surtout ne la fermez pas (ceci déconnecterait Jupyter de votre disque dur).



La fenêtre principale de Jupyter n'est ni plus ni moins qu'un explorateur de fichier relié à votre disque dur, vous pouvez ouvrir tous vos fichiers, dataset, codes, etc. depuis cette fenêtre.

Cliquez sur le bouton 'New' situé en haut à droite de cette fenêtre pour commencer à écrire un nouveau programme (puis cliquez sur Python 3).

La fenêtre suivante s'affiche !



3. Les bases de Python

- 1) **Les commentaires** : Dans Python, il suffit de précéder votre code du symbole « # » pour le transformer en commentaire.

```
|# ceci est un commentaire
```

- 2) **Les variables** : Pour définir une variable dans Jupyter, il suffit de lui donner un nom et de lui assigner une valeur. Par exemple, vous pouvez choisir de créer la variable « vitesse » et de lui assigner la valeur 90. Note : les accents et chiffres sont à bannir pour le nom d'une variable ! Vous pouvez effectuer des opérations mathématiques entre les variables numériques. Une fois le code écrit, appuyez sur CTRL + Entrée pour exécuter votre code. Le résultat est affiché et une nouvelle cellule s'ouvre en bas pour continuer à écrire du code.

```
In [1]: vitesse = 90 # vitesse en km/h
        distance = 180 # distance en km
        temps = distance / vitesse #temps du trajet en heures
        temps # afficher la valeur temps

Out[1]: 2.0

In [ ]:
```

- 3) **Les fonctions** : Python contient déjà de nombreuses bibliothèques remplies de fonctions très utiles et il n'est pas nécessaire de coder ses propres fonctions pour faire du Machine Learning. Cependant, je vais tout de même vous montrer la structure d'une fonction dans Python, pour votre connaissance personnelle. Rappelez-vous qu'une fonction transforme le plus souvent une entrée en sortie :

Pour créer une fonction dans Python, il faut commencer par écrire « **def** » en début de ligne, puis donner un **nom** à la fonction, écrire les **inputs** entre **parenthèse**, et finir la ligne par un « : ».

Les lignes suivantes font partie de la fonction, vous pouvez y écrire des commentaires, créer de nouvelles variables, faire des opérations mathématiques etc.

La fonction s'arrête à la ligne « **return** » qui indique quelle sortie la fonction doit produire.

Une fois la fonction créée, il est possible de l'utiliser ! Exemple :

```
In [2]: def distance_freinage(vitesse):  
        # cette fonction calcul la distance d'arret en metres  
        # en fonction de la vitesse a laquelle la voiture roule  
        arret = vitesse * 3 / 10 #arret est une variable numérique  
        return arret
```

```
In [3]: distance_freinage(45)
```

```
Out[3]: 13.5
```

```
In [ ]:
```

4) Les boucles :

a) **La boucle for** : Contrairement à la majorité des autres langages, en Python, une boucle **for** va forcément **itérer via une collection** (liste, dictionnaire, string, etc.).

```
In [1]: myList = [7,2,4,10]  
  
        for elt in myList:  
            print(elt)
```

```
7  
2  
4  
10
```

Parcours d'une liste via une boucle

b) **La boucle while** : Voici comment se présente la syntaxe :

```
In [2]: nombreArbres = 0  
  
        while nombreArbres < 10:  
            nombreArbres += 1  
            print("J'ai planté", nombreArbres, "arbres")
```

c) Les structures conditionnelles :

```
Entrée [3]: nombre=-1
if nombre==0:
    print("le nombre est nul !")
elif nombre>0:
    print("le nombre est positif!")
else :
    print("le nombre est négatif")

le nombre est négatif
```

Voyons maintenant les principales bibliothèques qui contiendront les fonctions à connaître pour faire du Machine Learning.

5) Les 4 bibliothèques à maîtriser pour le Machine Learning :

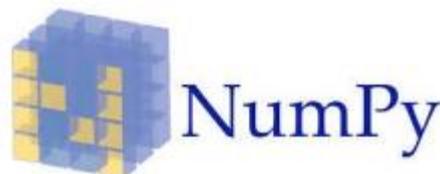
Pour importer une bibliothèque dans Python, il suffit d'écrire le nom de la bibliothèque précédé de « **import** » en tête de votre programme. Il est également possible d'importer certaines fonctions de la bibliothèque en écrivant **from** « **bibliothèque** » **import** « **fonction** ».

Exemple :

```
In [4]: import numpy as np
        from sklearn.datasets import make_regression

In [ ]:
```

A. Numpy est la bibliothèque qui permet de créer et manipuler des **matrices** simplement et avec efficacité. En Machine Learning, on insère le plus souvent notre **dataset** dans des **matrices**. Ainsi le calcul matriciel représente l'essentiel du Machine Learning. Il est important de le comprendre, mais les fonctions présentes dans **Numpy** font les calculs matriciels à notre place.



B. Matplotlib est la bibliothèque qui permet de **visualiser** nos datasets, nos fonctions, nos résultats sous forme de graphes, courbes et nuages de points. Nous nous servons principalement de `matplotlib.pyplot`.

```
import matplotlib.pyplot as plt
```

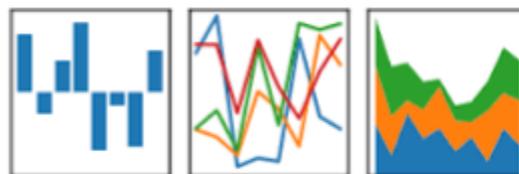


C. **Scikitlearn** est la librairie qui contient toutes les fonctions de l'état de l'art du **Machine Learning**. On y trouve les **algorithmes** les plus importants ainsi que diverses fonctions de **pre-processing**. Vous allez écrire de vrais programmes de Machine Learning en utilisant le module **Sklearn**.



D. **Pandas** est une excellente librairie pour importer vos **tableaux** Excel (et autres formats) dans Python dans le but de tirer des **statistiques** et de charger votre dataset dans Sklearn.

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Questions :

1. Écrire une fonction qui renvoie la plus grande de deux valeurs numériques passées en entrée.
2. À l'aide d'une boucle, écrire un code qui affiche les carrés des nombres entiers impairs de 1 à 17.
3. Créer un tableau d'entiers à trois lignes et trois colonnes pour sauvegarder les carrés calculés dans (2). Puis, faire l'extraction des colonnes d'indice impaire de tableau.
4. Multiplier une des colonnes par 2 et afficher le résultat.