
TP N° 1 : Introduction à Python et prise en main

Objectifs du TP : lancer jupyter notebook, sauvegarder, relancer etc., utiliser les premières commandes Python standard, importer une librairie.

1 Utiliser un jupyter notebook

Selon l'environnement que vous utilisez, veuillez vous reporter à l'aide correspondant à votre système d'exploitation (en anglais : *operating system*, OS). Pour ceux qui utilisent les machines des salles TP, les indications concernant le chargement de jupyter notebook est détaillée dans le polycopié du cours, Section 2.3.1 disponible ici :

<http://josephsalmon.eu/enseignement/Montpellier/HLMA310/IntroPython.pdf>

- 1) Lancer un nouveau jupyter notebook.
- 2) Vérifier quelle version de Python vous avez (sur les machines de l'université cela devrait être 3.5.5), en utilisant :

```
from platform import python_version
print(python_version())
```

- 3) Ajouter un titre dans une cellule sous format markdown (en utilisant l'onglet "Cell" puis "Cell Type" et en choisissant "Markdown"). Ce titre sera formé de la manière suivante :

```
# TP1 - HMLA310: Prénom Nom
```

où vous remplacerez "Nom" et "Prénom" par votre prénom et votre nom de famille. Pour obtenir quelques rudiments de markdown, vous pouvez consulter le lien :

<http://www.boiteaoutils.info/2013/02/ecrire-tout-simplement-introduction/>

- 4) Sauvegarder le notebook sous le nom `tp1_hmla_prenom_nom.ipynb`, où vous remplacerez `prenom_nom` par votre nom de famille et votre prénom (sans majuscule). Faites-le à la main dans cette question.
- 5) Tester l'auto-complétion de la manière suivante : dans une cellule de type code commencer à taper `pyth` et appuyer sur la touche "tab" du clavier. Vous devriez alors avoir accès à un menu déroulant et pouvoir retrouver la fonction `python_version` sans avoir à taper le nom entier.
- 6) Taper la commande `pwd` (pour *print working directory*) dans une cellule code. À l'endroit obtenue dans votre arborescence, créer un fichier texte (par exemple avec le logiciel gedit) composé des lignes suivantes :

```
from platform import python_version
print(python_version())
```

Enregistrer ce fichier à cet endroit sous le nom de `test.py`. Dans votre jupyter notebook vérifier avec la commande `ls` que le fichier `test.py` est bien disponible. Lancer alors ce fichier en exécutant la commande :

```
run test.py
```

On veillera dans la suite à créer une cellule par question, et avant chaque question on ajoutera un titre markdown, en tapant le type de commande suivante (par exemple pour la Question 1)

```
## Question 1:
```

2 Chaînes de caractères et nombres

- 7) Ici, on veut créer la chaîne de caractères `tp1_hmla_nom_prenom.ipynb` automatiquement en partant seulement des éléments suivants :

```
prenom = "Joseph" # à remplacer par ce qu'il faut
nom = "Salmon" # à remplacer par ce qu'il faut
extention = ".ipynb"
tp = "TP1_HMLA310"
```

En particulier on pourra utiliser la concaténation (ou bien la fonction `join`) et l'attribut `.lower()` (pour enlever les majuscules automatiquement). Vous pourrez alors copier-coller la chaîne de caractères obtenue pour remplacer le nom du fichier.

- 8) À partir de la chaîne de caractères suivante :

```
import string
alphabet = string.ascii_lowercase
```

générer par des opérations de *slicing* les chaînes de caractère `'cfilorux'` et `'zxvtrpnljhfdb'`.

- 9) Utiliser la commande “Restart & run all” de l'onglet “Kernel”. Quelle est la différence avec “Restart & clear output”.
- 10) Afficher le nombre π à 22 décimales après la virgule. Afficher de même $e = \exp(1)$ cette fois à 6 chiffres après la virgule. On pourra partir de l'exemple suivant :

```
s = "Le nombre {} est égal à {}"
print(s.format("pi", math.pi))
```

et consulter de l'aide sur l'affichage avec `format` pour des flottants disponible à l'adresse suivante : <https://pyformat.info/#number>

3 Listes

- 11) En partant de la syntaxe suivante proposer une manière d'obtenir une liste `lst_of_even` des nombres pairs entre 2 et 21 :

```
list_of_ints = list(range(1,10))
```

- 12) Modifier la liste `lst_of_even` pour qu'elle soit triée (en anglais :*sorted*) dans l'ordre décroissant ; on se servira de l'argument `sort` pour cela (taper par exemple `lst_of_even.` et tab, et chercher l'argument `sort`)¹.
- 13) Créer par *slicing* une nouvelle liste appelée `list_mult_four` contenant les nombres qui sont des multiples de 4 dans la liste `lst_of_even`.
- 14) Ajouter à `lst_of_even` l'entier 24 à la fin de la liste avec `append`. Dans un second temps rajouter 0 au début de la liste avec l'argument `insert` de la liste, et enlever les deux derniers nombres de la listes (les nombres les plus à droite donc) avec `remove` ou avec `pop`.
- 15) Exécuter la commande suivante :

```
print(2 in lst_of_even)
print(4 in lst_of_even)
print(3 in lst_of_even)
```

Pouvez vous interpréter la valeur des booléens créés ?

- 16) Supprimer le deuxième et le troisième élément de la liste `lst_of_even` avec la fonction `del`.

1. pour créer une copie de la liste qui soit triée il faut par contre utiliser la fonction `sorted`

4 Dictionnaires

Commençons par créer un dictionnaire vide avec la commande suivante² :

```
dico_vide = dict() # ou dictionnaire_vide = {}
```

17) Ajouter l'élément suivant au dictionnaire `dico_vide` clef : "Dioxyde de carbone" et valeur "CO₂".

18) Charger un dictionnaire de gaz polluants avec la commande³ :

```
import pandas as pd
url="http://josephsalmon.eu/enseignement/datasets/polluants.csv"
dico_polluants = pd.read_csv(url, header=None).set_index(0).squeeze().to_dict()
print(dico_polluants)
```

19) Fusionner dans le dictionnaire `dico_polluants` les deux dictionnaires précédents en utilisant la commande `update`.

20) Enlever du dictionnaire `dico_polluants` l'entrée 'Particules ou poussières en suspension' avec la commande `del`.

5 Pour aller plus loin

Quelques éléments de détails peuvent être nécessaires pour comprendre le type "range". Une discussion riche d'enseignements est disponible ici :

<https://www.pythoncentral.io/pythons-range-function-explained/>

2. Noter que la commande similaire pour créer une liste vide est `list_vide=[]`

3. On verra extensivement la librairie Python d'ici quelques séances, pour le moment on ne demande pas de comprendre la commande elle-même