

# Introduction à Python

Licence 3 Informatique

Maxime Bros  
Xlim, Université de Limoges

Novembre 2019

## 1 Présentation

Python est un langage interprété orienté objet ; un langage interprété ne nécessite pas de compilation car l'exécution d'une commande se fait directement dans ce que l'on appelle un *interpréteur*.

Python est dit de haut niveau car il vous évite des manipulations de bas niveau, comme par exemple les types, les pointeurs, la gestion mémoire, etc ; en d'autres termes il s'agit d'un langage très simple comparé par exemple au langage C.

L'appel à l'interpréteur Python se fait simplement en tapant `python3` dans un terminal. Si vous tapez alors `a=2` (pas besoin de ; en Python !), la variable `a` vaut alors 2, pour l'afficher il suffit de taper `print(a)`.

## 2 Exécution d'un code contenu dans un fichier

L'interpréteur Python est très pratique pour tester la validité d'une commande, néanmoins pour réaliser des programmes plus conséquents, ceux-ci doivent être écrits dans des fichiers.

Par convention, supposons que nous disposons d'un fichier `mon_programme.py`, en se plaçant dans le dossier contenant ce fichier, il suffit de taper la commande `python3 mon_programme` pour que celui-ci s'exécute.

## 3 Affichages de variables et de chaîne de caractères

- `print(a)` : affiche le contenu de la variable `a`.

- `print("la valeur de a est"+str(a))` : affiche la chaîne de caractères "la valeur de a est" suivie de la valeur de la variable *a* (qui a été convertie en chaîne de caractères avec la commande *str*).
- `print("a = %s et b = %s"%(a,b))` : cette commande est une alternative à la commande précédente, elle permet d'incruster la valeur d'une ou plusieurs variables dans une chaîne de caractère.

## 4 Les conditions et les boucles

Python semble "facile" puisqu'il n'exige aucun point-virgule à la fin des instructions ou aucun type pour les variables ; en effet rien ne nous empêche d'écrire `a=2`, *a* est alors un entier, puis quelques lignes plus loin `a="bonjour"`, *a* devient alors une chaîne de caractère.

On parle de *typage dynamique*, à ne pas confondre avec l'*inférence de types* de certains langages, comme par exemple Ocaml.

Néanmoins, Python est très exigeant en ce qui concerne les tabulations. Une absence de tabulation dans un bloc conditionnel ou dans une boucle, et c'est le drame!

Voici les principales boucles ou commandes conditionnelles :

```
for i in range(1,10) :
    print(i)
# instruction en dehors de la boucle
```

Vous remarquez que `range(1,10)` renvoie un tableau des entiers de 1 à 9 inclus.

```
if a < b :
    print("a<b")
else :
    print("a>=b")
# instruction en dehors de la condition
```

```
while a < b :
    # bloc d'instructions
# instruction en dehors de la boucle
```

## 5 Écrire dans un fichier

L'écriture dans un fichier (qui sera créé) se fait par la commande suivante :

```
s="salut\n" # une chaine de caracteres
f=open("nom_fichier","w")
f.write(s) #
f.close() # il ne faut pas oublier cette instruction
```

Naturellement plusieurs appels à la commande `write()` permettront d'écrire plusieurs fois dans le même fichier.

Après cette instruction, le fichier contient une ligne avec "salut", le prochain ajout se fera sur la ligne suivante grâce au saut de ligne contenu dans la chaîne.

## 6 Tirage au hasard

Pour tirer aléatoirement<sup>1</sup> des valeurs entières, il suffit d'importer le module `random` avec la commande `import random`; par la suite, chaque appel à la fonction `random.randint(a,b)` renverra de manière uniforme un entier compris entre *a* et *b* inclus.

---

1. Plus précisément, il s'agit de valeurs tirées *pseudo-aléatoirement*, car pour avoir un vrai aléa, il faut un processus physique, comme par exemple le classique pile ou face avec une vraie pièce (avec deux cotés différents de préférence).