

# TP 4 : Les listes en Python

PCSI, lycée Bertran de Born

## 1 Manipulations élémentaires de listes

### 1.1 Qu'est-ce qu'une liste ?

En Python, une liste est un objet dans lequel sont répertoriés des éléments, pas nécessairement du même type (`int`, `float`, `bool`, `string`, listes, etc.) L'idée est de pouvoir manipuler cette liste en ajoutant ou en enlevant des éléments, de pouvoir faire une recherche dans la liste, de compter le nombre d'éléments de la liste, etc. En pratique, une liste est délimitée par des crochets. Par exemple, on peut créer une liste vide avec la commande `list=[]`

### 1.2 Un exemple pour se faire la main

Des parents cherchent à faire une liste de prénoms pour la naissance d'un bébé à venir... sous Python...

1. La maman prend l'initiative en créant une liste avec 3 prénoms qu'elle aime bien grâce à la commande suivante :

```
liste_prenoms = ["Alexandre", "Amandine", "Anna"]
```

Exécutez cette commande dans la console Python.

2. Le papa se demande quel prénom a été mis en premier dans la liste et essaye les deux commandes suivantes :

```
liste_prenoms(1)  
liste_prenoms[1]
```

A-t-il réussi ? Que proposez-vous comme commande ?

3. Le papa veut maintenant connaître le dernier prénom. Quelle commande peut-il rentrer s'il sait qu'il y a exactement 3 prénoms dans cette liste ?
4. Le papa aime bien le prénom Anna, mais préfèrerait l'écrire Hannah. Proposez une commande permettant de faire la modification dans la liste.
5. La fonction `insert` permet d'insérer un élément dans la liste. Comme il adore le prénom Yvette, il décide d'exécuter la commande suivante

```
liste_prenoms.insert(3, "Yvette")
```

Exécutez cette commande, puis afficher la liste. Proposer ensuite une commande permettant d'insérer dans la liste le prénom Alexis de manière à ce que les prénoms restent dans l'ordre alphabétique.

6. Exéutez la commande suivante :

```
liste_prenoms[1:3]
```

De manière générale, décrire la commande `list[p:n]`.

7. Que fait la commande `liste_prenoms[0:2]=["Paul", "Jacques"]` ?
8. Que fait la commande `len(liste_prenoms)` ? La maman ne sais pas combien il y a de prénoms dans la liste. Quelle commande peut-elle exécuter pour afficher le dernier prénom de la liste actuelle ?

9. Tester les commandes `liste_prenoms[-1]` et `liste_prenoms[-3]`
10. On peut ajouter des éléments en fin de liste autrement qu'avec la fonction `insert` qui oblige à préciser le rang. Pour cela on utilise la fonction `append` pour ajouter un élément ou `extend` pour concaténer une nouvelle liste à la première. Pour vous en convaincre, exécutez dans la console les commandes suivantes :

```
liste_prenoms.append("Eléonore")
liste_prenoms
liste_prenoms.extend(["Audrey", "Ophélie"])
liste_prenoms
```

11. On peut évidemment supprimer des éléments de la liste. Pour cela on peut utiliser les fonctions `del` ou `pop`. Pour vous en convaincre, exécutez dans la console les commandes suivantes :

```
del liste_prenoms[1:3]
liste_prenoms
liste_prenoms.pop(4)
liste_prenoms
```

12. Si l'on ne sait pas où se situe l'élément que l'on veut supprimer dans la liste, on peut utiliser la fonction `remove`. Exécuter les commandes suivantes :

```
liste_prenoms.remove("Paul")
liste_prenoms
```

13. Exécuter la commande `liste_prenoms.clear()` A quoi sert cette commande ?
14. Comme dit en préambule, il est possible de créer des listes de listes. Dans `temp.py`, écrire le programme suivant **sans l'exécuter**, puis prédire ce qui sera affiché dans la console à l'exécution.

```
liste_garcons=["Alexandre", "Alexis"]
liste_filles=["Delphine", "Zoe"]
liste_prenoms=[liste_garcons, liste_filles]
print(liste_prenoms)
print(liste_prenoms[1])
print(liste_prenoms[1][0])
print(liste_prenoms[1][0][2])
```

## 2 Exercices

**Exercice 1.** Faire un programme Python qui demande à l'utilisateur 5 prénoms, crée une liste avec ces 5 prénoms, puis affiche ensuite une phrase du type : “les prénoms de la liste commençant par la lettre A sont ... et ...”

**Exercice 2.**

1. Faire un programme Python qui demande à l'utilisateur autant d'entiers qu'il le souhaite, puis qui crée deux listes, l'une avec la liste des nombres proposés et l'autre avec le nombre d'occurrences. Par exemple, si l'utilisateur entre 4,4,8,4,9,7,7, la première liste devra être [4,8,9,7] et la deuxième [3,1,1,2]
2. Adaptez le programme pour que la première liste soit dans l'ordre croissant et que la liste des occurrences corresponde.