

# TP 1 - Introduction à Python

Olivier Goudet

February 6, 2025

## Exercice 1 - Utilisation des listes en Python

### Questions

1. Construire une liste *liste1* qui contient les nombres de 1 à 99 inclus (avec la fonction *range*).
2. Ajouter à l'aide d'une boucle les éléments de 100 à 199, à une liste vide *liste2*.
3. En une opération, ajouter tous les éléments de la liste *liste2* à la liste *liste1*.
4. Avec la fonction *range()*, construire une liste3 composée de tous les nombres pairs de 0 à 100 (inclus).
5. À l'aide d'une boucle supprimez tous les multiples de 10 contenus dans la liste *liste3*.
6. Affichez la longueur de la liste *liste3*.
7. Affichez tous les multiples de 8 de *liste3*.
8. Affichez les 20 premiers éléments de la liste *liste3* (sans faire de boucle).
9. Affichez les éléments suivants les 20 premiers éléments de *liste3*.
10. Affichez les 10 derniers éléments de *liste3*.

Python propose la compréhension de listes permettant de construire une liste en une seule ligne avec une formule du type : *newList = [element for element in otherlist if condition]*. La nouvelle liste sera composée des éléments contenus dans une autre liste ou séquence (ici *otherlist*) et vérifiant la condition imposée. Par exemple on, on peut créer la liste des entier compris entre 5 et 8 de la façon suivante : *liste = [i for i in range(10) if (i > 5 and i < 8)]*.

11. Créer une *liste4* composée uniquement des nombres impairs jusqu'à 50 en une ligne.
12. On dispose de la liste de mots suivantes :  
*listWords = ["hello", "the", "world", "apple", "tree", "lemon"]*.  
En une seule ligne de code, créez une nouvelle liste contenant les mots de *listWords* de moins de 5 lettres.
13. Construire une *liste5* contenant les 20 premiers termes de la suite de Fibonacci. La suite de Fibonacci  $(F_n)_{n \in \mathbb{N}}$  est définie par  $F_0 = 0$ ,  $F_1 = 1$ , et la relation de récurrence  $F_n = F_{n-1} + F_{n-2}$  pour  $n \geq 2$ .
14. Créer une *liste6* avec les valeurs [1,2,3] puis créer une copie de *liste6* dans une *liste6bis* (attention si on change une valeur de *liste6bis*, la *liste6* ne doit pas être modifiée). cf. fonction *copy()*.

## Exercice 2 - Utilisation des tableaux NumPy

1. Importer la librairie NumPy.
2. Créer un numpy array de dimension 1 et de taille 10 contenant toujours la valeur 0.
3. Créer un numpy array de dimension 1 et de taille 10 contenant toujours la valeur 1.
4. Créer un numpy array de dimension 1 et de taille 10 contenant toujours la valeur 5.
5. Créer un numpy array tab1 de dimension 1 contenant tous les nombres de 1 à 10 avec un pas de 0.5.
6. Créer un numpy array tab2 de dimension 1 avec les entiers de 10 à 30.
7. Calculer la somme des entiers de tab2 (une seule ligne).
8. Calculer la moyenne des entiers de tab2 (une seule ligne).
9. Créer un vecteur tab3 de même dimension que tab2 contenant des valeurs aléatoires et effectuer le produit scalaire des vecteurs tab2 et tab3.
10. Sélectionner les éléments de tab2 dont la valeur est supérieure à 20.
11. Sélectionner les éléments en position 1, 5, 10 et 15 de tab2.
12. Générer une matrice identité de taille 9 (voir la fonction *eye* de la librairie numpy).
13. Créer un numpy array de dimension 2 et de taille  $10 \times 10$  contenant toujours la valeur 1.
14. Afficher les dimensions de la matrice précédente.
15. Créer un numpy array de dimension 1 avec les valeurs de 0 à 8 et le transformer en matrice  $3 \times 3$ .
16. Créer la matrice A suivante :  
0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  
0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.2  
0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3  
0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  
0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  
0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6  
0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7  
0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  
0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9  
0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.0
17. Extraire la matrice B suivante de la matrice A précédente :  
0.61 0.62 0.63 0.64  
0.71 0.72 0.73 0.74  
0.81 0.82 0.83 0.84  
0.91 0.92 0.93 0.94
18. Générer une matrice de 1 de la taille de la matrice B.
19. Extraire la matrice C suivante de la matrice A :  
0.21 0.23 0.25 0.27  
0.31 0.33 0.35 0.37
20. Calculer le produit matriciel entre la matrice C et la matrice B.
21. Récupérer la valeur 0.55 dans la matrice A.

22. Récupérer la 4ème colonne de la matrice A
23. Récupérer la 4ème ligne de la matrice A
24. Récupérer une sous-matrice de la matrice A en ne gardant que les lignes telles que la valeur de la colonne 4 est supérieure à 0.5
25. Calculer la moyenne globale de cette matrice A.
26. Calculer la moyenne de chaque ligne de la matrice A.
27. Calculer la moyenne de chaque colonne de la matrice A.
28. Calculer le produit scalaire de la première colonne de la matrice A avec la première ligne de la matrice A.
29. Remplacer le centre de la matrice par la valeur 0 comme l'exemple suivant :
 

0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.2
0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29	0.3
0.31	0.32	0.33	0	0	0	0	0.38	0.39	0.4
0.41	0.42	0.43	0	0	0	0	0.48	0.49	0.5
0.51	0.52	0.53	0	0	0	0	0.58	0.59	0.6
0.61	0.62	0.63	0	0	0	0	0.68	0.69	0.7
0.71	0.72	0.73	0.74	0.75	0.76	0.77	0.78	0.79	0.8
0.81	0.82	0.83	0.84	0.85	0.86	0.87	0.88	0.89	0.9
0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	1.0
30. Récupérer les index auxquels sont situées les valeurs 0.