

Module : Programmation en Python

Compte rendu TP2

Réalisé par :

- **Aymane Wardi**
- **Hajar Rachid**

Objectif du TP :

L'objectif de ce TP est de pratiquer la programmation en Python à travers la création de fonctions simples qui manipulent des listes. Ce travail permet de renforcer la compréhension des boucles, des conditions, des types de données (listes, tuples, chaînes de caractères) et des fonctions prédéfinies du langage Python.

Exercice 1 :

```
def calcule_moyenne(liste:list[int]):\n\n    if(len(liste)==0) :\n\n        print('0.0')\n\n    else:\n\n        somme=0\n\n        for i in range(len(liste)):\n\n            somme=somme+liste[i]\n\n        moyenne=somme/len(liste)\n\n        print("la moyenne est:",moyenne)\n\n        print(liste)\n\nliste=[1,2,3,4]\n\ncalcule_moyenne(liste)
```

Exercice 2 :

```
def trouver_indices(l:list[int], valeur):\n    indices = []\n    for i in range(len(l)):\n        if l[i] == valeur:\n            indices.append(i)\n    return indices\nprint(trouver_indices([5,18,5,69,5,5],5))
```

Exercice 3 :

```
def fusionner_et_trier(liste1:list[int], liste2:list[int]):
    liste3=[]
    liste3=liste1+liste2
    print(liste3)
    liste3.sort()
    print(liste3)
liste1=[0,9,8,7]
liste2=[5,6,7,1]
fusionner_et_trier(liste1,liste2)
```

Exercice 4 :

```
def filtrer_par_longueur(liste: list[str]):
    liste1 = []
    n = int(input("Donner un nombre: "))
    for i in range(len(liste)):
        if len(liste[i]) >= n:
            liste1.append(liste[i])
    return liste1
```

Exercice 5 :

```
def regrouper_pairs_impairs(liste: list[int]):
    liste1 = [] # nombres pairs
    liste2 = [] # nombres impairs

    for x in liste:
        if x % 2 == 0:
            liste1.append(x)
        else:
            liste2.append(x)

    return (liste1, liste2)
```

Exercice 6 :

```
def compter_occurrences(liste:list[str]=[], cible:str=""):

    compteur = 0

    for mot in liste:

        if mot == cible:

            compteur += 1

    print(compteur)
```

Exercice 7 :

```
def trouver_doublons(liste: list[int]):

    doublons = []

    for element in liste:

        if liste.count(element) > 1 and element not in doublons:

            doublons.append(element)

    return doublons
```

Conclusion:

À la fin de ce TP, nous avons appris à manipuler les listes en Python à travers plusieurs exercices pratiques. Ces exercices nous ont permis de comprendre comment créer et utiliser des fonctions, parcourir des listes avec des boucles, filtrer des données à l'aide de conditions, et exploiter certaines fonctions intégrées comme `sum()`, `len()`, `sorted()` ou encore `count()`.

Ce travail nous a aidées à renforcer notre logique de programmation et à développer une meilleure compréhension du fonctionnement du langage Python. Nous avons également amélioré notre capacité à résoudre des problèmes de manière structurée et à écrire un code clair et efficace.