

# Correction - Batterie Listes

## 1. Somme des éléments d'une liste :

Écrivez une fonction

`somme_liste` qui prend une liste en entrée et renvoie la somme de tous ses éléments en utilisant une boucle.

2.

## Éléments communs à deux listes :

Écrivez une fonction

`elements_communs` qui prend deux listes en entrée et renvoie une liste contenant les éléments communs aux deux listes, sans utiliser de fonction de conversion de type ou de compréhensions de liste.

3.

## Éléments pairs d'une liste :

Écrivez une fonction

`elements_pairs` qui prend une liste en entrée et renvoie une nouvelle liste contenant uniquement les éléments pairs de la liste, sans utiliser de compréhensions de liste.

4.

## Vérification si une liste est triée :

Écrivez une fonction

`est_triee` qui prend une liste en entrée et renvoie True si la liste est triée par ordre croissant, sinon False, en utilisant une boucle.

5.

## Moyenne des éléments d'une liste :

Écrivez une fonction

`moyenne_liste` qui prend une liste en entrée et renvoie la moyenne de ses éléments en utilisant une boucle.

6.

**Suppression des doublons d'une liste :**

Écrivez une fonction

`sans_doublons` qui prend une liste en entrée et renvoie une nouvelle liste sans doublons, en conservant l'ordre initial des éléments, sans utiliser de dictionnaires.

7.

**Mots dont la longueur est supérieure à un nombre donné :**

Écrivez une fonction

`mots_longueur_superieure` qui prend une liste de mots et un entier représentant la longueur minimale, et renvoie une liste contenant uniquement les mots dont la longueur est supérieure à l'entier donné.

8.

**Vérification d'éléments communs entre deux listes :**

Écrivez une fonction

`elements_communs_present` qui prend deux listes en entrée et renvoie True si les deux listes ont au moins un élément en commun, sinon False, sans utiliser de fonctions de conversion de type ou de compréhensions de liste.

Voici les corrections pour chaque exercice :

**1. Somme des éléments d'une liste :**

```
def somme_liste(liste):  
    somme = 0  
    for element in liste:  
        somme += element  
    return somme
```

**2. Éléments communs à deux listes :**

```
def elements_communs(liste1, liste2):
    communs = []
    for element in liste1:
        if element in liste2 and element not in communs:
            communs.append(element)
    return communs
```

### 3. Éléments pairs d'une liste :

```
def elements_pairs(liste):
    pairs = []
    for element in liste:
        if element % 2 == 0:
            pairs.append(element)
    return pairs
```

### 4. Vérification si une liste est triée :

```
def est_triee(liste):
    for i in range(len(liste) - 1):
        if liste[i] > liste[i + 1]:
            return False
    return True
```

### 5. Moyenne des éléments d'une liste :

```
def moyenne_liste(liste):
    somme = 0
    for element in liste:
        somme += element
```

```
return somme / len(liste)
```

## 6. Suppression des doublons d'une liste :

```
def sans_doublons(liste):  
    sans_doublons = []  
    for element in liste:  
        if element not in sans_doublons:  
            sans_doublons.append(element)  
    return sans_doublons
```

## 7. Mots dont la longueur est supérieure à un nombre donné :

```
def mots_longueur_superieure(liste_mots, longueur):  
    mots_filtres = []  
    for mot in liste_mots:  
        if len(mot) > longueur:  
            mots_filtres.append(mot)  
    return mots_filtres
```

## 8. Vérification d'éléments communs entre deux listes :

```
def elements_communs_present(liste1, liste2):  
    for element in liste1:  
        if element in liste2:  
            return True  
    return False
```