

Interrogation : Boucles en python

L'évaluation porte sur 3 exercices indépendants.

Les exercices sont notés sur 18 et la rigueur, rédaction et justifications sont notés sur 2 points.

Exercice 1 : Questions de cours (2 points)

1. On peut itérer sur 2 propriétés d'une liste à l'aide d'une boucle `for`. Quelles sont ces 2 propriétés ?
2. Une propriété précédente permet de localiser la position dans une liste. Quelle est la position de l'élément `l[0]`? `l[4]`?
3. Peut-on modifier les éléments d'une liste ? d'un tuple ?

Exercice 2 : Complétion de programme (4 points)

Compléter la fonction `rangement_valeurs` qui prend en paramètre une liste et un élément, et renvoie 3 listes : une liste contenant les valeurs inférieures à l'élément, une liste avec l'élément si présent, et une liste avec les valeurs supérieures.

Exemple :

```
rangement_valeurs([1, 7, 4, 3, 6, 2, 8], 5) # Renvoie: [1, 4, 3, 2], [], [7, 6, 8]
rangement_valeurs([1, 2, 4, 3, 6, 2, 8], 2) # Renvoie: [1], [2, 2], [4, 3, 6, 8]
```

```
def rangement_valeurs(liste : list, valeur : int) -> tuple(list):
    inferieurs = []
    superieures = []
    egales = []
    for ... :
        if ... > valeur :
            superieures.append(valeur)
        elif ... == ... :
            egales = egales + [...]
        else:
            inferieurs = inferieurs + [...]
    return ...
```

Exercice 3 : Écriture de programme (12 points)

Écrire une fonction `compter_pairs` qui prend une liste d'entiers et renvoie le nombre d'éléments pairs dans cette liste.

Exemple :

`compter_pairs([1, 2, 3, 4, 5, 6])` doit renvoyer 3.

Écrire une fonction `moyenne` qui prend en paramètre une liste d'entiers et renvoie la moyenne de tous les nombres présents dans cette liste.

Exemple :

`moyenne([1, 2, 3, 4, 5])` doit renvoyer 3.0.

Écrire une fonction `somme_elements` qui prend une liste de nombres en paramètre et renvoie la somme de tous les éléments.

Exemple :

`somme_elements([1, 2, 3, 4])` doit renvoyer 10.

Écrire une fonction `separer_pairs_impairs` qui prend une liste d'entiers et renvoie deux listes : une avec les éléments pairs et une autre avec les éléments impairs.

Exemple :

`separer_pairs_impairs([1, 2, 3, 4, 5])` doit renvoyer `([2, 4], [1, 3, 5])`.