

---

## TP\_5 : Les Listes

---

### 1. Exercice

---

Soit les trois lignes de code suivantes :

```
liste1 = list(range(10, 15))  
var = 0  
var2 = 10
```

Pr disez le comportement de chaque instruction ci-dessous, sans les recopier dans un script ni dans l'interpr teur

Python :

- a) `print(liste1[2])`
- b) `print(liste1[var])`
- c) `print(liste1[var2])`
- d) `print(liste1["var"])`

Lorsqu'une instruction produit une erreur, identifiez pourquoi.

### 2. Exercice

---

Soit L une liste de N r els. Ecrire un programme qui permet de calculer le nombre des occurrences d'un nombre X (c'est- -dire combien de fois ce nombre X figure dans la liste L).

### 3. Exercice

---

L est une liste de n entiers. Proposer un programme qui permet le remplissage de la liste L, puis affiche le nombre de valeurs n gatives et le nombre de valeurs positives.

Exemple d'ex cution :

```
Donner n : 6  
Remplir L : 1 -5 -9 3 4 6  
Le nombre des nombres positifs : 4  
Le nombre des nombres n gatifs : 2
```

### 4. Exercice

---

On suppose qu'on a L une liste remplie par des entiers. On veut remplir deux listes L1 et L2   partir de la liste L tel que :

- 1- les nombres pairs dans L seront copi s dans L1.
- 2- Le reste dans L2

Exemple d'exécution :

```
Donner le nombre des éléments à insérer : 10
```

```
Résultat :
```

```
L : -9 1 6 14 -8 100 3 44 63 10
```

```
L1 : 6 14 -8 100 44
```

```
L2 : -9 1 3 63
```

## 5. Exercice

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]

## 6. Exercice

Écrire un script pour trouver le deuxième plus grand élément d'une liste sans utiliser la méthode sort.

Le second maximum d'une liste est sa deuxième plus grande valeur. Par exemple, le second maximum de [1, 2, 14, 8, 20, 7] est 14. On suppose que les listes considérées ont des éléments tous distincts.

Pour déterminer le second maximum d'une liste, on propose l'algorithme suivant :

- ▲ Initialiser deux variables **m1** et **m2** égales aux deux premiers éléments et telles que **m1** > **m2**.
- ▲ Parcourir les éléments **li** de la liste à partir du troisième élément :
  - ▲ Si **li** > **m1**, alors **li** est le nouveau maximum : mettre à jour **m2** = **m1** et **m1** = **li**.
  - ▲ Si **m2** < **li** < **m1**, alors **li** est le nouveau second maximum : mettre à jour **m2** = **li**.
- ▲ Une fois qu'on a parcouru toute la liste **m2** est le second maximum.

## 7. Exercice

1. Créez une liste de listes nommée **matrice** qui représente la matrice suivante :

```
[[1, 2, 3],
```

```
[4, 5, 6],
```

[7, 8, 9]]

2. Affichez la matrice en utilisant une double boucle for (boucles imbriquées) pour afficher chaque élément, ligne par ligne.
3. Affichez la valeur de l'élément de la deuxième ligne et de la troisième colonne.
4. Modifiez la valeur de l'élément de la première ligne et de la première colonne pour qu'il devienne 10. Affichez la matrice pour vérifier la modification.
5. Affichez l'ensemble de la deuxième ligne

## 8. Exercice

---

Écrire un programme en **Python** qui transfère les éléments d'une **liste à deux dimensions** M (L lignes et C colonnes, maximum 10×10) dans une **liste à une dimension** V contenant tous les éléments ligne par ligne.

$$M \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \Rightarrow_{to} V(a\ b\ c\ d\ e\ f\ g\ h\ i\ j\ k\ l)$$