

Laboratoire 5

- **Durée 3 heures**

EXERCICE 1

Écrivez une fonction **selecte_third** qui permet de retourner le troisième élément d'une liste donnée (comme paramètre). Si la liste n'a pas de troisième élément, elle renvoie **None**.

EXERCICE 2

Étant donné une liste d'équipes, où chaque équipe est une liste de noms ou l'entraîneur est le premier nom de la liste, le capitaine est le deuxième nom de la liste, et les autres joueurs sont listés après. Écrivez une fonction qui renvoie le 2e joueur (capitaine) de la dernière équipe listée.

EXERCICE 3

Étant donné une liste de candidats, mettez le premier candidat (au début de la liste) à la dernière place et vice versa.

Console

```
7  myList = ["Ali", "Pierre", "Ribal"]
8  reverse_candidate(myList)
9  print(myList) # ['Ribal', 'Pierre', 'Ali']
10
```

EXERCICE 4

Nous utilisons des listes pour enregistrer les élèves qui ont assisté à notre cours et l'ordre dans lequel ils sont arrivés. Par exemple, la liste suivante représente une classe avec 7 invités, dans laquelle Adela est arrivée en premier et Ford en dernier :

```
student_attendees = ['Adela', 'Fleda', 'Owen', 'May', 'Mona', 'Gilbert', 'Ford']
```

Un élève est considéré comme "en retard" s'il est arrivé après au moins la moitié des élèves de la classe. Toutefois, il ne doit pas être le tout dernier élève (ce serait aller trop loin). Dans l'exemple ci-dessus, Mona et Gilbert sont les seuls étudiants à être en retard.

Écrivez une fonction ci-dessous qui prend une liste des étudiants dans une classe ainsi que le nom d'un étudiant, et nous dit si cet étudiant est en retard ou non.

EXERCICE 5

Programme une fonction `rotation(liste)` qui décale d'un rang tous les éléments d'une liste (le dernier élément devenant le premier). La fonction renvoie une nouvelle liste. Par exemple `rotation([1,2,3,4])` renvoie la liste `[4,1,2,3]`.

EXERCICE 6

Programme une fonction `inverser(liste)` qui inverse l'ordre des éléments d'une liste. Par exemple `inverser([1,2,3,4])` renvoie la liste `[4,3,2,1]`.

EXERCICE 7

Programme une fonction `supprimer_rang(liste,rang)` qui renvoie une liste formée de tous les éléments, sauf celui au rang donné. Par exemple `supprimer_rang([8,7,6,5,4],2)` renvoie la liste `[8,7,5,4]` (l'élément 6 qui était au rang 2 est supprimé).

EXERCICE 8

Programme une fonction `supprimer_element(liste,element)` renvoyant une liste qui contient tous les éléments sauf ceux égaux à l'élément spécifié. Par exemple `supprimer_element([8,7,4,6,5,4],4)` renvoie la liste `[8,7,6,5]` (tous les éléments égaux à 4 ont été supprimés).

EXERCICES SUPPLÉMENTAIRES

EXERCICE 1

Écrivez une fonction qui reçoit deux listes en entrée et vérifie si elles ont au moins un membre en commun.

Console

```
listA = [1, 5, -7]
listB = [1, 0, 2]
listC = [0, 0, 2]

flag = common(listA, listB)
print(flag) # True
flag = common(listA, listC)
print(flag) # False
```

EXERCICE 2

Écrivez une fonction qui prend une liste comme paramètre. Elle divise cette liste en deux, stocke les éléments dans deux listes différentes et les renvoie.

Console

```
myList = [1, 5, 8, 9, 7]
listA, listB = split_list(myList)
print(listA) # [1, 5]
print(listB) # [8, 9, 7]
```

EXERCICE 3

Écrire une fonction qui supprime plusieurs chaînes vides d'une liste de chaînes de caractères.

Console

```
6  myList = [1, 5, 8, 9, 7, ""]
7  remove_empty(myList)
8  print(myList) # [1, 5, 8, 9, 7]
```

EXERCICE 4

Écrivez une fonction qui prend deux paramètres : une liste (myList) et un entier(seuil). Cette fonction retourne une liste qui contient tous les nombres dans myList dont la fréquence est supérieure à une valeur donnée (seuil).

Console

```
myList = [1, 5, 8, 9, 7, 7, 2, 7, 1]
listA = more_frequent(myList, 2) # displays [7]
print(listA)
```

EXERCICE 5

Écrire une fonction qui prend une liste en paramètre. Cette fonction met au carré chaque élément de la liste et puis elle retourne cette liste dans l'ordre inverse.

Console

```
myListA = [1, 5, 8, 9, 7]
myList = square_reverse(myListA)
print(myList) # [49, 81, 64, 25, 1]
```

EXERCICE 6

Écrire une fonction qui prend une liste et un entier en paramètre puis elle supprime toutes les occurrences de cet entier et puis elle retourne *True* et la liste si l'entier est dans la liste. Sinon elle retourne *False* et la liste.

Console

```
myListA = [1, 5, 8, 9, 7]
flag, test_list = remove_an_item(myListA, 5)
print(flag, test_list) # True [1, 8, 9, 7]

flag, test_list = remove_an_item(myListA, 2)
print(flag, test_list) # False [1, 8, 9, 7]
```

EXERCICE 7

Écrire une fonction qui prend une liste des listes. Puis elle crée une liste plate à partir de ces listes qui ne sont vides.

Console

```
myList1 = [[1, 2, 3], [4, 5], [6, 7, 8]]
result_list = flat_list(myList1)
print(result_list ) # [1, 2, 3, 4, 5, 6, 7, 8]

myList1 = [[1, 2, 3], [4, 5], [6, 7, 8], []]
result_list = flat_list(myList1)
print(result_list ) # [1, 2, 3, 4, 5, 6, 7, 8]
```