
Correction TP1 Python

Exercice 1

1) On affiche L par la commande `print(L)`, la moins bonne note par la commande `min(L)` et la meilleure note par la commande `max(L)`.

2) programme :

```
for i in range(len(L)):
    if L[i]<10:
        L[i]=L[i]+1
print(L)
```

Exercice 2

1) La liste contient tous les termes consécutifs d'une suite arithmétique de premier terme 5 et de raison 3, sans dépasser la valeur 100.

Donc `nombres=[5,8,11,...,98]`.

2)i)

```
print(nombres)
```

2)ii)

```
nombres.append(40)
```

2)iii)

```
print(nombres[4])
```

2)iv)

```
print(nombres[4:10])
```

2)v)

```
print(nombres[-1])
```

2)vi)

```
del nombres[6]
```

Exercice 3

1)a) `len(L)` représente la longueur de la liste L, c'est-à-dire le nombre d'éléments qu'elle possède.

k prend toutes les valeurs entières entre 0 et `len(L)-1`.

1)b) Ce programme renvoie le nombre d'occurrences du chiffre 4 dans la liste, c'est-à-dire le nombre de fois qu'il y figure.

2) La commande `L.count(4)` renvoyait le même résultat.

3) Prenons par ex. `L=[2,5,4,6,4,7,8,4]`. La commande `L.count(4)` renvoie 3.

Exercice 4

1) Le programme crée une liste intitulée `pos`, vide initialement.

Dans la boucle, il regarde si la valeur d'index k de la liste L vaut 7.

Si c'est le cas, il ajoute cette valeur à la liste `pos`.

Le programme renvoie donc finalement la liste de tous les index de la liste L où la valeur fait 7.

Remarque

La commande hors programme `L.index(7)` renvoie l'indice de la première occurrence de 7.

2) Avec `L=[1,3,8,7,4,5,6,7,2]`, on obtient `pos=[3,8]`.

Avec `L=[5,4,6,3,1,2]`, on a `pos=[]` puisqu'il n'y a pas de 7 dans la liste L.

Exercice 5

1) On cherche dans les entiers i compris entre 700 et 999 ceux qui sont multiples de 7 et impairs.

On les met dans une liste, notée L.

On affiche cette liste ainsi que le nombre d'éléments de cette liste.

2) Le programme affiche :

[707, 721, 735, 749, 763, 777, 791, 805, 819, 833, 847, 861, 875, 889, 903, 917, 931, 945, 959, 973, 987]

Le nombre cherché est : 21.

Exercice 6

1)a) Le mot `rd` est un alias de `numpy.random`, une abréviation autrement dit dont on choisit le nom comme on veut.

`randint(1,11)` désigne un entier aléatoire entre 1 et 10.

1)b) La liste est construite en compréhension. Elle est formée de 5 entiers pris aléatoirement entre 1 et 10.

2)a) Les erreurs de type `IndexError` sont l'un des types d'erreurs d'exécution les plus courants en Python. Elles se produisent lorsque qu'on essaie d'accéder dans une liste à une valeur dont l'index n'existe pas.

Prenons un exemple :

Imaginons que la liste initiale vaut : `liste=[2,5,10,4,7]`.

Lors des deux premiers passages dans la boucle (`i=0` et `i=1`), il ne se passe rien car `liste[i]≠10`.

Lors du troisième passage dans la boucle, on a `i=2` et `liste[2]=10`.

Python va donc supprimer de la liste l'élément `liste[2]`.

La liste devient donc `liste=[2,5,4,7]`.

Ensuite, lors du quatrième passage dans la boucle, on a `i=3` et `liste[3]=7`.

Il ne se passe rien.

On fait ensuite un dernier passage dans la boucle, on a `i=4`. Le test demande de regarder si `liste[4]` fait 10. Là survient l'erreur car `liste[4]` n'existe pas !

Le programme de Tom va donc entraîner une erreur d'index dès lors que la liste initiale possède un 10.

b) Tom souhaitait enlever les 10 de la liste initiale. Une solution possible :

```
import random as rd
liste=[rd.randint(1,11) for k in range(5)]
print(liste)
nouvelle_liste=[ ]
for i in range(5):
    if liste[i]!=10:
        nouvelle_liste.append(liste[i])
print(nouvelle_liste)
```

Exercice 7

1) Cette fonction renvoie la liste de départ, mais écrite dans le sens inverse.

Remarque

La commande `old_list.reverse()` menait au même résultat, mais n'est pas au programme de la classe.

2) Par exemple, si `old_list=[2,5,3,7,9]`, la fonction renvoie la liste `[9,7,3,5,2]`.

Exercice 8

1) On obtient :

```
max=L[0]
min=L[0]
for i in range(len(L)):
    if L[i]>max:
        max=L[i]
    if L[i]<min:
        min=L[i]
print(max,min)
```

2) Les commandes `min(L)` et `max(L)` renvoient le plus petit élément et le plus grand élément de `L`.

Exercice 9

Tant que `ma_liste` n'est pas vide, on ajoute un élément à `new_liste` et on en retire un à `ma_liste`.

Cet élément est le plus petit élément de `ma_liste`.

Au final, `new_liste` contient tous les éléments de `ma_liste` ordonnés dans l'ordre croissant, c'est donc la liste `[2,2,3,3,4,5,7,8,11,15]`.

Remarque

La commande hors programme `ma_liste.sort()` a même effet que le programme.