

## Exercices d'application sur les listes (apcpedagogie.com)

### Exercice 1 :

Ecrire un programme python qui crée une liste semaine qui comprend les jours de la semaine, puis à l'aide de parcours successifs de la liste effectue les actions suivantes :

- 1) Affiche la liste semaine
- 2) Affiche la valeur semaine[4]
- 3) Echange les valeurs de la première et de la dernière case de la liste
- 4) Affiche 6 fois les valeurs du dernier terme de la liste.

### Exercice 2 :

Ecrire une fonction `supprimeDoubleton(liste)` qui supprime les doublons d'une liste saisie par l'utilisateur (on créera une nouvelle liste).

**Exemple** : si l'utilisateur saisit la liste suivante : [2,4,6,2,4,7,1], l'appel de la fonction renvoie : [2,4,6,7,1]

### Exercice 3 :

Ecrire un programme qui demande à l'utilisateur de saisir une liste d'entiers, puis à l'aide de parcours successifs de la liste effectue les actions suivantes :

- 1) Affiche la liste
- 2) Affiche la liste en colonne de manière à afficher l'index et son contenu
- 3) Additionne tous les éléments de la liste
- 4) Crée une nouvelle liste qui sera le multiple (3) de tous les éléments de la liste
- 5) Obtenir le plus grand nombre de la liste
- 6) Obtenir le plus petit nombre de la liste
- 7) Effectue la somme des nombres pairs présents dans la liste
- 8) Effectue la somme des nombres impairs présents dans la liste

### Exercice 4 :

Ecrire un programme qui trouvera tous les nombres divisibles par 7 mais non multiples de 5 et de 2, entre 700 et 1099 (les deux inclus). Les nombres obtenus doivent être imprimés dans une liste que l'on affichera.

### Exercice 5 :

Soit la liste suivante :

```
stock = ["Ordinateur de bureau", "Ordinateur portable", 100, "Caméra", 310.28, "Haut-parleurs", 27.00, "Télévision", 1000, "Cartes mères", "souris", 500, "Barrettes de mémoire"]
```

- 1) Afficher la liste "stock"
- 2) Créer des listes séparées composées de chaînes et de nombres
- 3) Compter le nombre d'éléments de chaque liste
- 4) Trier la liste des chaînes par ordre croissant
- 5) Trier la liste des chaînes par ordre décroissant
- 6) Trier la liste des numéros du plus petit au plus grand
- 7) Trier la liste des numéros du plus grand au plus petit

CORRIGES : <http://apcpedagogie.com/exercices-en-python-sur-les-listes/>

## CORRIGE – Notre Dame de La Merci – Montpellier

### Exercice 1 :

Ecrire un programme python qui crée une liste semaine qui comprend les jours de la semaine, puis à l'aide de parcours successifs de la liste effectue les actions suivantes :

- 1) Affiche la liste semaine
- 2) Affiche la valeur semaine[4]
- 3) Echange les valeurs de la première et de la dernière case de la liste
- 4) Affiche 6 fois les valeurs du dernier terme de la liste.

```
semaine = ["lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"]
print(semaine)
print(semaine[4])
semaine[0], semaine[6] = semaine[6], semaine[0]
for i in range(6):
    print(semaine[6])
```

### Exercice 2 :

Ecrire une fonction *supprimeDoublon(liste)* qui supprime les doublons d'une liste saisie par l'utilisateur.

**Exemple** : si l'utilisateur saisit la liste suivante : [2,4,6,2,4,7,1], l'appel de la fonction renvoie : [2,4,6,7,1]

```
liste_nb = [2,4,6,2,4,7,1]
print(liste_nb)
long=len(liste_nb)
liste_nb_db=[]
for i in range(long):
    if liste_nb[i] not in liste_nb_db:
        liste_nb_db.append(liste_nb[i])
print(liste_nb_db)
```

→ [2, 4, 6, 2, 4, 7, 1]  
[2, 4, 6, 7, 1]

### Variante : on laisse l'utilisateur créer une liste de nombres :

```
print("Veuillez saisir vos valeurs, pressez simplement ENTREE lorsque vous avez terminé")
liste_nb = []
while True :
    valeur = input("Saisir une valeur")
    if valeur:
        try:
            liste_nb.append(int(valeur))
        except ValueError:
            print("Cette saisie n'est pas valide")
    else :
        break
print("Valeurs saisies :",liste_nb)
long=len(liste_nb)
liste_nb_db=[]
for i in range(long):
    if liste_nb[i] not in liste_nb_db:
        liste_nb_db.append(liste_nb[i])
print(liste_nb_db)
```

### Exercice 3 :

Ecrire un programme qui demande à l'utilisateur de saisir une liste d'entiers, puis à l'aide de parcours successifs de la liste effectue les actions suivantes :

- 1) Affiche la liste
- 2) Affiche la liste en colonne de manière à afficher l'index et son contenu
- 3) Additionne tous les éléments de la liste
- 4) Crée une nouvelle liste qui sera le multiple (3) de tous les éléments de la liste
- 5) Obtenir le plus grand nombre de la liste
- 6) Obtenir le plus petit nombre de la liste
- 7) Effectue la somme des nombres pairs présents dans la liste
- 8) Effectue la somme des nombres impairs présents dans la liste

Notre liste de départ : liste\_nb = [3, 17, 21, 14, 19, 14, 30, 22]

```
liste_nb = [3, 17, 21, 14, 19, 14, 30, 22]
```

```
# afficher la liste
```

```
print(liste_nb)
```

```
#afficher les indes et les valeurs de la liste en colonne
```

```
long=len(liste_nb)
```

```
for i in range (long) :
```

```
    print (i,liste_nb[i])
```

```
#calcul de la somme de tous les termes
```

```
somme=0
```

```
for i in range(long):
```

```
    somme+=liste_nb[i]
```

```
print("La somme des nombres est :",somme)
```

```
#calcul de la somme de tous les termes : VARIANTE
```

```
somme=0
```

```
for x in liste_nb:
```

```
    somme+=x
```

```
print("La somme des nombres TEST est :",somme)
```

```
#création d'une liste contenant les triples des valeurs initiales
```

```
liste_triple=[]
```

```
for x in liste_nb:
```

```
    liste_triple.append(3*x)
```

```
print("La liste triple est:",liste_triple)
```

```
#détermination du maximum
```

```
max=liste_nb[0]
```

```
for x in liste_nb:
```

```
    if x>max:
```

```
        max=x
```

```
print("Le maximum de la liste est :",max)
```

```
#somme des termes pairs de la liste
```

```
somme paire=0
```

```
for x in liste_nb:
```

```
if x%2==0:
    sommepaire+=x
print("La somme des nombres pairs est :",sommepaire)
```

→ [3, 17, 21, 14, 19, 14, 30, 22]

0 3

1 17

2 21

3 14

4 19

5 14

6 30

7 22

La somme des nombres est : 140

La somme des nombres TEST est : 140

La liste triple est: [9, 51, 63, 42, 57, 42, 90, 66]

Le maximum de la liste est : 30

La somme des nombres pairs est : 80

#### **Exercice 4 :**

Ecrire un programme qui trouvera tous les nombres divisibles par 7 mais non multiples de 5 et de 2, entre 700 et 1099 (les deux inclus). Les nombres obtenus doivent être imprimés dans une liste que l'on affichera.

#### **Exercice 5 :**

Soit la liste suivante :

```
stock = ["Ordinateur de bureau", "Ordinateur portable", 100, "Caméra", 310.28, "Haut-parleurs",
27.00, "Télévision", 1000, "Cartes mères", "souris", 500, "Barrettes de mémoire"]
```

- 1) Afficher la liste "stock"
- 2) Créer des listes séparées composées de chaînes et de nombres
- 3) Compter le nombre d'éléments de chaque liste
- 4) Trier la liste des chaînes par ordre croissant
- 5) Trier la liste des chaînes par ordre décroissant
- 6) Trier la liste des numéros du plus petit au plus grand
- 7) Trier la liste des numéros du plus grand au plus petit