

Exercices en Python traitement des listes série 01

1. Objectifs

- Savoir utiliser les listes sous Python

2. Exercice 01

○ Énoncé

- Écrire 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 effectuer les actions suivantes :
 - Afficher la liste semaine
 - Afficher la valeur de semaine[4]
 - Échanger les valeurs de la première et de la dernière case de cette liste
 - Afficher 12 fois la valeur du dernier élément de la liste

3.

Exercice 02

○ Énoncé

- Écrire une fonction supprimeDoublon(liste) qui supprime les doublons d'une liste saisie par l'utilisateur.
- Si l'utilisateur à saisie la liste suivante: ([3,4,5,3,4,5,1]) l'appel de la fonction renvoie [3,4,5,1].

○

4. Exercice 03

○ Énoncé

- Écrire un programme qui demande à l'utilisateur de saisir une liste d'entiers, puis à l'aide de parcours successifs de la liste effectuer les actions suivantes :
 - Afficher la liste
 - Afficher la liste en colonne de manière à afficher l'index et sont contenu
 - Additionner tous les éléments de la liste.
 - Créer une nouvelle liste qui sera le multiple (**3**) de tous les éléments de la liste.
 - Obtenir le plus grand nombre de la liste.
 - Obtenir le plus petit nombre de la liste.
 - Compter le nombre des nombres pairs présents dans la liste

- Calculer la somme de tous les nombres impairs de la liste

5. Exercice 04

- Énoncé

- Écrivez un programme qui trouvera tous les nombres divisibles par 7 mais non multiples de 5 et 2, entre 700 et 1099 (les deux inclus). Les nombres obtenus doivent être imprimés dans une liste.
- Affichez le nombre des nombres obtenus

○

6. Exercice 05

- Énoncé

- Soit la liste suivante:
stock = ["Ordinateur de bureau", "Ordinateur portable", 100, "Caméra", 310.28, "Haut-parleurs", 27.00, "Télévision", 1000, "Cartes mères", "souris", "clavier", 500, "barrettes de mémoire"]
 - Afficher la liste "**stock**"
 - Créer des listes séparées composées de chaînes et de nombres.
 - Compter le nombre d'élément de chaque liste
 - Trier la liste de chaînes par ordre croissant
 - Trier la liste de chaînes par ordre décroissant
 - Trier la liste des numéros du plus petit au plus grand
 - Trier la liste des numéros du plus grand au plus petit

○

7. Exercice 06

- Énoncé

- La liste suivante représente les moyennes d'une classe
- **moyennes=[14.84,14.14,16.22,86,85,85,14.84,13,15.85,9.99,12.04,15.03,16.22,12,84,10.20,11.03,11.03]**
 - Afficher les trois bonnes moyennes
 - Afficher les trois mauvaises moyennes (triées de plus petites au plus grandes)

○

8. Exercice 07

- Énoncé

- **Supprimer les éléments dupliqués d'une liste Python**
- Écrire un programme Python qui permet de :

- Demander à l'utilisateur de saisir une liste d'entiers (en cas d'erreur le programme redemande la saisie sans être planté), la liste sera imprimée devant l'utilisateur chaque fois qu'il réussit d'ajouter un élément à la liste.
- Compter le nombre des éléments dupliqués de la liste.
- Supprimer les éléments dupliqués de la liste et afficher la nouvelle liste.

○

9. Exercice 08

○ Énoncé

- Pour le code suivant quel est le résultat de chaque impression (**print**) après exécution

```
numList = [1,2,3,4,5]
alphaList = ["a","b","c","d","e"]
print(numList is alphaList)
print(numList == alphaList)
numList = alphaList
print(numList is alphaList)
print(numList == alphaList)
```

○

2. Exercice 09

○ Énoncé

- Vous développez une application Python pour votre entreprise.
- Une liste nommée `employés` contient 200 noms d'employés, les cinq derniers étant des membres de la direction de la société. Vous devez découper la liste en tranches pour afficher tous les employés, à l'exception de la gestion.
- Quels deux segments de code devriez-vous utiliser ? Chaque réponse correcte présente une solution complète. (Choisissez deux.)
- A. `employees [1:-4]`
- B. `employees[:-5]`
- C. `employees [1:-5]`
- D. `employees [0:-4]`
- E. `employees [0:-5]`

○

3. Exercice 10

○ Énoncé

- Vous trouvez des erreurs lors de l'évaluation du code suivant. Les numéros de ligne sont inclus pour référence seulement.

```

1 numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
2 index = 0
3 while (index < 10)
4     print (numbers [index])
5
6     if numbers(index) = 6:
7         break
8     else :
9         index += 1

```

- Vous devez corriger le code aux lignes 03 et 06.
- Comment devriez-vous corriger le code ? Utilisez les menus déroulants pour sélectionner le choix de réponse qui répond à chaque question en fonction des informations présentées dans le segment de code.
- Zone de réponse proposée :

Quel segment de code devez-vous utiliser à la ligne 03?

	▼
while (index < 10) :	
while [index < 10]	
while (index < 5) :	
while [index < 5]	

Quel segment de code devez-vous utiliser à la ligne 06 ?

	▼
if numbers[index] == 6	
if numbers[index] == 6 :	
if numbers(index) = 6 :	
if numbers(index) != 6	

○

4.

5.

Exercice 11

○ Énoncé

- Définir la liste suivante: liste =[17, 38, 10, 25, 72], puis effectuez les actions suivantes :
 1. Triez et affichez la liste ;
 2. Ajoutez l'élément 12 à la liste et affichez la liste ;

3. Renversez et affichez la liste ;
4. Affichez l'indice de l'élément 17 ;
5. Enlevez l'élément 38 et affichez la liste ;
6. Affichez la sous-liste du 2e au 3e élément ;
7. Affichez la sous-liste du début au 2e élément ;
8. Affichez la sous-liste du 3e élément à la fin de la liste ;
9. Affichez la sous-liste complète de la liste ;
10. Affichez le dernier élément en utilisant un indicage négatif

○

6. Exercice 12

○ Énoncé

- Initialisez **lstVide** comme une liste vide, et **lstFlottant** comme une liste de cinq flottants nuls.
- Affichez ces listes.
- Ajouter à la liste **lstVide** les nombres entre 0 et 1000 avec "step" de 200
- Utilisez la fonction range() pour afficher :
 1. Les entiers de 0 à 3 ;
 2. Les entiers de 4 à 7 ;
 3. Les entiers de 2 à 8 par pas de 2.
- Définir **lstElmnt** comme une liste des entiers de 0 à 5 et testez l'appartenance des éléments 3 et 6 à chose.
- Ajouter le contenu des deux listes() à la fin de la liste **lstElmnt**.

○