

TP5 – Python

Pr. Houda BENHAR

Exercice 1.

Executer le code suivant:

```
x = [1, 2, 3]
y = x
```

Afficher le contenu de y.

Maintenant exécuter l'instruction suivante :

```
x[1] = -15
```

Afficher le contenu de y et de x.

Maintenant testez le code ci-dessous :

```
x = [1, 2, 3]
y = x[:] #ou y = list(x)
x[1] = -15
print(x)
print(y)
```

Quelle est votre remarque?

Exercice 2.

Écrivez un programme permettant de rechercher tous les diviseurs d'un nombre et de les stocker dans une liste. Les diviseurs d'un nombre correspondent à tous les nombres par lesquels on peut le diviser sans laisser de reste lors de la division.

Exercice 3.

Définir la liste: liste =[17, 38, 10, 25, 72, -1, 16, 82, 0.5], puis effectuez les actions suivantes :

- Triez et affichez la liste
- Ajoutez l'élément 12 à la liste et affichez la liste
- Renversez et affichez la liste
- Affichez l'indice de l'élément 17
- Enlevez l'élément 38 et affichez la liste
- Affichez la sous-liste du 2e au 3e élément
- Affichez la sous-liste du début au 2e élément
- Affichez la sous-liste du 3e élément à la fin de la liste
- Affichez la sous-liste complète de la liste

- Affichez le premier et le dernier élément

Exercice 4.

Le but de cet exercice est d'ajouter plusieurs éléments dans une liste en une seule fois ! Vous devez ajouter plusieurs éléments dans la liste originale, et non pas imbriquer une liste dans une autre. La liste de départ contient les éléments 1, 2 et 3. La liste finale doit contenir les éléments 1, 2, 3, 4, 5 et 6. Vous devez donc ajouter les éléments 4, 5 et 6 à la liste originale.

Exercice 5.

Ecrire un programme qui permet d'enlever les doublons d'une liste en utilisant l'opérateur **in**.

Exemple :

```
list_doublons = [1, 2, 2, 3, 4, 5, 5, 5, 6, 7, 7, 8, 9, 10]
resultat: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Réécrire le programme en utilisant les méthodes de liste.

Exercice 6.

Écrire une fonction qui prend en argument un tuple et un élément et qui renvoie la position de cet élément dans le tuple. Elle renverra -1 si l'élément n'est pas trouvé.

Exercice 7.

Écrire une fonction qui prend en argument un tuple composé de nombres entiers et renvoie un tuple contenant le plus grand et le plus petit des entiers.

Exercice 8.

Créer une fonction `cumul_valeurs` qui permet de renvoyer une liste contenant le cumul de toutes les valeurs d'une liste récupérée en paramètre. C'est-à-dire que chaque valeur de la liste renvoyée contient la somme de toutes les valeurs précédentes de la liste récupérée en paramètre (la somme se fait de gauche à droite).

Exemples d'utilisation :

```
cumul_valeurs([1, 4, 7]) → [1, 5, 12]
cumul_valeurs([1, 2, 3, 4, 5, 6]) → [1, 3, 6, 10, 15, 21]
cumul_valeurs([]) → []
```

Exercice 9.

Créez une fonction `moyenne_eleves` qui prend en paramètre un dictionnaire avec en clé les noms des élèves et en valeur une liste contenant leurs différentes notes. La fonction devra renvoyer un dictionnaire en associant les noms des élèves en clé avec leur moyenne.

Exercice 10.

Supposons que vous avez une liste de dictionnaires comme suit :

```
livres = [  
    {"titre": "Dune", "auteur": "Frank Herbert"},  
    {"titre": "1984", "auteur": "George Orwell"},  
    {"titre": "Le Meilleur des mondes", "auteur": "Aldous Huxley"},  
    {"titre": "Fondation", "auteur": "Isaac Asimov"},  
    {"titre": "La Ferme des animaux", "auteur": "George Orwell"},  
]
```

une fonction qui prend en argument une liste de dictionnaires représentant des livres (voir liste livres au-dessus) et renvoie un dictionnaire dont les clés sont les noms des auteurs et les valeurs sont des listes de titres de leurs livres.