

TP4

I Listes, Tuples, Dictionnaires

Exercice 1.1

Écrire un programme en python sous forme de fonction qui :

- Prend en paramètres une liste l
- Renvoie un tuple de deux listes (l_even, l_odd) où l_even est composé des éléments de l d'indice pair et l_odd est constitué par les éléments d'indice impair.

Exercice 1.2

Écrire un programme en python sous forme de fonction qui :

- Prends en paramètre un couple (L, a) formé d'une liste L et d'un élément a
- Renvoie la position de l'élément a dans la liste L. La fonction doit renvoyer -1 si l'élément a n'est pas présent dans la liste.

Exercice 1.3

Écrire un programme en Python qui :

- Demande à l'utilisateur de saisir un texte
- Renvoie un dictionnaire dont les clés sont les mots du texte saisi et les valeurs sont les longueurs des mots qui composent le texte.

Exemple pour le texte T = "Python est un langage de programmation" , le programme doit renvoyer le dictionnaire:

```
d={'Python': 6 , 'est': 3 , 'un': 3 , 'langage': 7 , 'de': 2 ,  
'programmation': 13 }
```

Exercice 1.4

Étant donnée un dictionnaire dict dont les valeurs des clés sont des listes.

Écrire un programme Python qui permet de transformer le dictionnaire dict en triant les listes. Exemple pour le dictionnaire :

```
dict = {'a1': [21, 17, 22, 3] , 'a2': [12, 14, 9 , 16] , 'a3': [7,  
13, 2, 11] , 'a4': [22, 14, 7, 9] }
```

Le programme doit renvoyer le dictionnaire :

```
dict = {'a1': [3, 17, 21, 22] , 'a2': [9, 12, 14 , 16] , 'a3': [7, 2,  
11, 13] , 'a4': [9, 7, 14, 22] }
```

II POO en python

Exercice 2.1

1. Tester l'exemple vu pendant la séance du cours

Exercice 2.2

2. Écrire une classe Rectangle en langage Python, permettant de construire un rectangle dotée d'attributs longueur et largeur.
3. Créer une méthode perimetre() permettant de calculer le périmètre du rectangle et une méthode surface() permettant de calculer la surface du rectangle.
4. Créer une classe fille Parallelepipede héritant de la classe Rectangle et dotée en plus d'un attribut hauteur, d'une méthode surface() qui utilise celle de la classe mère et une autre méthode volume() permettant de calculer le volume du Parallélépipède.

III Les fichiers

Exercice 3.1

- 1- Écrire un programme Python permettant de
 - a. créer un fichier sur le bureau nommé "myFile.txt"
 - b. créer un répertoire sur le bureau nommé "new" (`os.mkdir(path)`)
 - c. déplacer le fichier "myFile.txt" vers le répertoire "new". (`shutil.move(src, dest)`)

Exercice 3.2

1. Écrire un programme en Python qui permet de créer un fichier nommé myFile.txt et d'ajouter les lignes suivantes :
 - Ligne numéro 1
 - Ligne numéro 2
 - Ligne numéro 3
 - Ligne numéro 4
 - Ligne numéro 5
2. Écrire un programme en Python qui permet de remplacer la 3ème ligne par la phrase "désolé ! Le contenu de cette ligne a été changé !"

Exercice 3.3

1. Écrire un programme en Python qui permet de créer un fichier nommé myFile.txt et d'ajouter les lignes suivantes :
 - Python Programming
 - Java Programming
 - C++ Programming
2. Écrire un programme en Python qui permet d'échanger la troisième ligne avec la deuxième ligne du fichier myFile.txt.