

Exercice - Tri par sélection sur dictionnaire

Le professeur principal d'une classe souhaite manipuler les notes de ses élèves afin de pouvoir suivre l'évolution des résultats de ses élèves. Les résultats de chaque élève sont contenus dans un dictionnaire nommé avec le nom de famille. Par exemple pour l'élève Dupont Sami, on aurait :

```
>>> dupont
{'maths': 12, 'philo': 5, 'anglais': 16, 'espagnol': 3,
'eps': 18, 'his-géo': 8, 'nsi': 11}
```

Pour créer ce dictionnaire, le professeur a utilisé les commandes suivantes :

Création du dictionnaire
avec initialement déjà 3
éléments en contenu

```
dupont={'maths':12,'philo':5,'anglais':16}
dupont['espagnol'] = 3
dupont['eps'] = 18
dupont['his-géo'] = 8
dupont['nsi'] = 11
```

Ajout de 4 éléments
supplémentaires

Clé | Valeur

L'objectif de cet exercice est de créer une fonction qui permettra d'afficher les valeurs triées d'un dictionnaire en se basant sur les clés ou sur les valeurs.

1- TRAVAIL PRELIMINAIRE :

On demande dans un premier temps d'écrire le code d'une fonction nommée *dicListe()*. Elle prend en argument un dictionnaire et renvoie une liste composés des tuples (clé,valeur) des différents éléments du dictionnaire. Par exemple, l'exécution de cette fonction dans la console, avec en argument le dictionnaire *dupont*, donnera :

```
>>> dicListe(dupont)
[('maths', 12), ('philo', 5), ('anglais', 16),
 ('espagnol', 3), ('eps', 18), ('his-géo', 8),
 ('nsi', 11)]
```

Conseil : Avant de commencer, relisez le [cours](#) sur les dictionnaires et posez-vous les questions suivantes :

- Comment crée-t-on un dictionnaire ? avec des accolades : `dic = {}` ou `dic = {cle1 : valeur1 , cle2 :valeur2, ... }`
- Comment ajoute-t-on des éléments ? `dic[cle3] = valeur3`
- Comment accède-t-on aux éléments qu'il contient pour les lire ? pour les modifier ? `dic[cle3]` ou `dic[cle3] = autrevaleur`
- Comment parcourt-on un dictionnaire ? `for cle in dic :`
- Comment supprime-t-on des éléments ? `del(dic[cle3])`

Ne pas hésiter à tester des lignes de commandes dans la console pour s'appropriier tout cela.

```
def dicListe(dic) : CORRIGE  
    l = []  
    for cle in dic :  
        l.append((cle,dic[cle]))  
    return l
```

2- FINALISATION :

Ecrire le code d'une fonction nommée *triDic()*. Elle prend en argument un dictionnaire et un entier *k* égal à 0 ou 1. Elle ne renvoie rien, mais permet d'afficher dans la console le dictionnaire trié par clés si *k = 0* ou par valeurs si *k = 1*.

Le tri sera réalisé en réutilisant les fonctions *echange()* et *triSelection()* mises au point dans ce chapitre. Le fonction *triSelection()* sera à modifier légèrement.

Tri par clés : $k = 0$	Tri par valeurs : $k = 1$
<pre>>>> triDic(dupont,0) dic[anglais] = 16 dic[eps] = 18 dic[espagnol] = 3 dic[his-géo] = 8 dic[maths] = 12 dic[nsi] = 11 dic[philo] = 5</pre>	<pre>>>> triDic(dupont,1) dic[espagnol] = 3 dic[philo] = 5 dic[his-géo] = 8 dic[nsi] = 11 dic[maths] = 12 dic[anglais] = 16 dic[eps] = 18</pre>

```
def triDic(dic,k) : CORRIGE  
    l = dicListe(dic)  
    l = triSelection(l,k)  
    for i in range(len(l)) :  
        print(f"dic[{l[i][0]}] = {l[i][1]} ")
```

```
def triSelection(liste,k): CORRIGE  
    n = len(liste)  
    for i in range(n-1):  
        jMin = i  
        for j in range(i+1,n):  
            if liste[j][k]<liste[jMin][k]:  
                jMin = j  
        echange(liste,i,jMin)  
    return liste
```

Le test est fait sur 1
des 2 éléments du
tuple

C'est tout le tuple
qui est échangé