



TD : Les listes de dictionnaires

On va travailler ici avec une structure de données bien pratique, les listes de dictionnaires.

Pour créer une telle structure, il suffit de créer une liste, chaque élément étant un dictionnaire .

Exemple :

Voici le tableau initial :

Nom	Points
Steeve	25
Kevina	12
Jacky	20
Momo	23
Steeve	14
Johanna	22
Momo	35
Kevina	14

On veut une liste *resultats* qui contient les résultats de différents joueurs et leur score :

1/ Combien y a-t-il de clés et quelles sont elles ?

2/ Créer une liste *resultats* et ajouter les différentes lignes, chaque ligne correspondant à un dictionnaire.

3/ Classer cette liste par ordre alphabétique. On a vu que les méthodes **sort** et **sorted** « faisaient le travail »

4/ Classer cette liste par score décroissant. En cas d'égalité, c'est l'ordre alphabétique qui prévaut.

Pour le double tri, on utilise les mêmes méthodes (sort ou sorted) et presque la même syntaxe

resultats.sort(key=lambda s:{s["nom"],s["points"]}).

Au lieu de taper les clés « à la main », on peut importer ces données et créer directement une liste de dictionnaire. Voici la syntaxe :

```
def import_fichier_csv(nom_fichier):
    """
    Retourne une liste de dictionnaires contenant les données du fichier CSV
    passées en paramètre.

    Le fichier CSV doit être placé dans le même dossier que ce module et
    utiliser l'encodage UTF-8.

    Les clés du dictionnaire sont données par la première ligne du fichier CSV

    Paramètre
    -----
    nom_fichier : str
        Une chaîne de caractères représentant le nom du fichier CSV à importer

    Valeur de retour
    -----
    list
        Une liste de dictionnaire, chaque élément de la liste représentant une
        ligne du fichier CSV sous forme d'un dictionnaire.

    """

    liste_enregistrements = []

    # Ouverture du fichier
    with open(nom_fichier, 'r', newline='', encoding="utf-8") as fichier_csv:
        lecteur_csv = csv.DictReader(fichier_csv, delimiter=";")

        # Lecture de chaque ligne et conversion en dictionnaire
        # (par défaut, la lecture retourne OrderedDict qu'on ne pourrait pas
        # modifier)
        for ligne in lecteur_csv:
            liste_enregistrements.append(dict(ligne))

    return liste_enregistrements
```

Tester cette méthode avec le fichier csv donné.