

```

1 import csv
2
3 def charger_liste(nom_fic):
4     """
5     Permet de charger un fichier CSV
6     paramètre : nom_fic une chaîne de caractères contenant le nom du
    fichier csv
7     résultat : la liste des enregistrements dans le fichier (une liste
    de listes)
8     """
9     liste_enreg = []
10    with open (nom_fic, "r", newline="", encoding = "utf-8") as
    csvfile :
11        # création du lecteur csv
12        fich_reader = csv.reader(csvfile , delimiter = ";")
13        for enreg in fich_reader :
14            #enreg est de type list
15            #Le premier étant l'entête sous forme d'une liste
16            liste_enreg.append(enreg)
17    return liste_enreg
18 jo = charger_liste("JO_Liste_Medaille_2012-2014.csv")
19
20 def appartient(nom, table):
21     for ligne in table:
22         if ligne[3] == nom:
23             return True
24     return False
25 appartient("LAVILLENIE", jo)
26 appartient("NAT", jo)
27
28 def nombre_medailles(athlete, table):
29     """Donne le nombre de médailles obtenues par l'athlète
30     paramètres : athlete, une chaîne de caractère correspondant au nom
    de l'athlète
31     et table, la liste des médaillés aux JO
32     résultat : un entier égal au nombre de médailles obtenues par l'
    athlète"""
33     nombre_medailles = 0
34     for ligne in table:
35         if ligne[3] == athlete:
36             nombre_medailles += 1
37     return nombre_medailles
38 nombre_medailles("LAVILLENIE", jo)
39 nombre_medailles("LE FUR", jo)

```

```

1 def select(table):
2     new_liste = []
3     for ligne in table:
4         if ligne[6] == "Or":
5             new_liste.append(ligne)
6     return new_liste
7 select(jo)
8
9 def select(table):
10    new_liste = []
11    for ligne in table:
12        if ligne[5] == "F" and ligne[6] == "Argent":
13            new_liste.append(ligne)
14    return new_liste
15
16 select(jo)
17
18 def trier_liste(table):
19     return sorted(table, key=lambda ligne: ligne[3])
20 trier_liste(jo)
21
22 def trier_liste2(table):
23     return sorted(table, key=lambda ligne: ligne[4])
24 trier_liste(jo)
25
26 def charger_liste_dico(nom_fic):
27     """Permet de charger un fichier CSV
28     paramètre : nom_fic une chaine de caractères contenant le nom du
29     fichier csv, sans son extension
30     résultat : la liste de tous les enregistrements du fichier (une
31     liste de dictionnaires)"""
32     table = []
33     # ouverture du fichier CSV
34     with open(nom_fic+".csv", "r", newline="", encoding = "utf-8") as
35     csvfile :
36         # création du lecteur csv indiquant le caractère séparateur
37         # la ligne d'entête est utilisée pour créer les clés des
38         dictionnaires
39         element_reader = csv.DictReader(csvfile, delimiter = ",")
40         for enreg in element_reader :
41             # enreg est une liste de str contenant chaque champ de l'
42             enregistrement
43             # ajout de l'enregistrement dans la liste sous forme d'un
44             dictionnaire
45             table.append(dict(enreg))
46     return table
47 heros = charger_liste_dico("superHeros")

```

```

1 def extraire(table):
2     new_heros =[]
3     for dico in table:
4         new_heros.append({ "Nom" : dico["Nom"], "Pseudo" : dico["Pseudo
5         " ]})
6     return new_heros
7
8
9 def extraire2(table,liste_cles):
10    new_heros =[]
11    for dico in table:
12        new_dico = {cle : dico[cle] for cle in dico if cle in
13        liste_cles}
14        new_heros.append(new_dico)
15    return new_heros
16
17
18 extraire2(heros ,["Nom","Année_apparition"])
19
20 def fusion_listes_memes_champs(tab1,tab2):
21     """Fusion de tables identiques
22     paramètres : les deux tables à fusionner (liste de dictionnaires)
23     sortie : nouvelle liste constitué des deux autres"""
24     return tab1+tab2
25
26 fusion_listes_memes_champs(heros,heroines)
27
28 def recherche_doublons(tab):
29     """Rechercher d'éventuels doublons et les supprimer
30     paramètre : tab, une liste de dictionnaires
31     sortie : une liste sans doublons"""
32     for i in range(len(tab)):
33         present = False
34         for j in range(i+1,len(tab)):
35             if tab[i]["nom"] == tab[j]["nom"]:
36                 present = True
37         if present == True:
38             tab.remove(tab[i])
39     return tab

```

```

1 liste=[{"nom":"bubulle","categorie":"élève","age":15},{ "nom":"nat",
    "categorie":"alien","age":18},
2         {"nom":"bubulle","categorie":"élève","age":15},{ "nom":"nat",
    "categorie":"alien","age":18},
3         {"nom":"nat","categorie":"vivante","age":18},{ "nom":"toto",
    "categorie":"singe","age":25}]
4 liste_essai = list(liste) #on copie la liste dans une nouvelle liste
    pour la modifier
5 recherche_doublons(liste_essai)
6 liste_eleves = [{"nom" : "Dupond", "prénom" : "Jacques", "age" : 16},
7                 {"nom" : "Bourgeois", "prénom" : "Nicolas", "age" :
    16},
8                 {"nom" : "Etoile", "prénom" : "Caroline", "age" : 17},
9                 {"nom" : "Trunc", "prénom" : "Jessie", "age" : 15},
10                {"nom" : "Kennedy", "prénom" : "Maelle", "age" : 16}]
11 liste_notes = [{"nom" : "Dupond", "prénom" : "Jacques", "maths" : 12,
    "français" : 9},
12                {"nom" : "Bourgeois", "prénom" : "Nicolas", "maths" :
    19, "français" : 7},
13                {"nom" : "Etoile", "prénom" : "Caroline", "maths" :
    12, "français" : 9},
14                {"nom" : "Trunc", "prénom" : "Jessie", "maths" : 5, "
    français" : 12},
15                {"nom" : "Kennedy", "prénom" : "Maelle", "maths" :
    10, "français" : 17}]
16
17 def creation_fusion(enreg1,enreg2):
18     """Fonction créant un dictionnaire eleve avec tous les champs de
    la fusion
19     paramètres : enreg1 et enreg2 sont des éléments des deux tables à
    fusionner ;
20     sortie : on retourne le dictionnaire complet représentant l'élève.
    """
21     return {"nom" : enreg1["nom"], "prénom" : enreg1["prénom"], "age"
    : enreg1["age"],
22            "maths" : enreg2["maths"], "Français" : enreg2["français"]
    }
23
24 def fusion(tab1,tab2):
25     tableau_complet = []
26     for dico1 in tab1:
27         for dico2 in tab2:
28             if dico1["nom"] == dico2["nom"]:
29                 tableau_complet.append(creation_fusion(dico1,dico2))
30     return tableau_complet
31 fusion(liste_eleves,liste_notes)

```