1. Trier des données

(version élève)

Nous reprenons notre fichier de joueurs de rugby du Top14.

In [ ]:
```
import csv
f = open('data/top14.csv', "r", encoding = 'utf-8')
donnees = csv.DictReader(f)
joueurs = []
for ligne in donnees:
    joueurs.append(dict(ligne))

f.close()
```

1.0.1 Créer une fonction filtre

L'objectif est de créer une fonction `joueursEquipe(equipe)` qui renvoie une liste contentant tous les joueurs de l'équipe `equipe`. Le paramètre `equipe` sera donnée sous forme de chaîne de caractères. La valeur renvoyée sera de type liste.

In [ ]:
```
def joueursEquipe(equipe):
    ret = []

    for k in joueurs :
        if k['Equipe'] == equipe :
            ret.append(k)

    return ret
```

Exemple d'utilisation :

In [ ]:
```
len(joueursEquipe("Bordeaux"))
```

Définir de la même manière une fonction `joueursPoste(poste)`.

In [ ]:
```
```

1.1 Utilisation d'une fonction de tri

Comment classer les joueurs suivant leur taille ? La fonction `sorted(liste)` est efficace sur les listes : elle renvoie une nouvelle liste triée dans l'ordre croissant.

In [ ]:
```
mylist = [4,2,8,6]
mynewlist = sorted(mylist)
print(mynewlist)
```

Mais comment trier un dictionnaire ?

In [ ]:
```
test = sorted(joueurs)
```

Il est normal que cette tentative échoue : un dictionnaire possède plusieurs clés différentes. Ici, plusieurs clés peuvent être des critères de tri : la taille, le poids.

1.1.2 un exemple de tri de dictionnaire

In [ ]:
```
Simpsons = [{"Prenom" : "Bart", "age estimé": "10"},
            {"Prenom" : "Lisa", "age estimé": "8"},
            {"Prenom" : "Maggie", "age estimé": "1"},
            {"Prenom" : "Homer", "age estimé": "38"},
            {"Prenom" : "Marge", "age estimé": "37"}]
```

In [ ]:
```
def age(personnage):
    return int(personnage["age estimé"])
```

In [ ]:
```
age(Simpsons[0])
```

La création de cette fonction `age()` va nous permettre de spécifier une clé de tri, par le paramètre `key` :

In [ ]:
```
```

In [ ]:
```
triSimpsons = sorted(Simpsons, key = age)
```

In [ ]:
```
triSimpsons
```

In [ ]:
```
triSimpsons = sorted(Simpsons, key = age, reverse = True)
```

In [ ]:
```
triSimpsons
```

1.2 Exercice

1. Trier les joueurs du top14 par taille.
2. Trier les joueurs de Bordeaux par taille.
3. Trier les joueurs de Bordeaux suivant leur Indice de Masse Corporelle (IMC)

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

1.3 Recherche des joueurs de profil physique similaire

1.3.3 Distance entre deux joueurs

Construire une fonction `distance(joueur1,joueur2)` qui renvoie la somme des carrés des différences de tailles et de poids entre les joueurs `joueur1` et `joueur2` : $$ d = (p_1-p_2)^2 + (t_1-t_2)^2$$

In [ ]:

1.3.4 Distance des joueurs avec Baptiste Serin

Retrouvons d'abord le numéro de Baptiste Serin dans notre classement de joueurs :

In [ ]:

In [ ]:

Nous pouvons maintenant classer les joueurs suivant leur distance morphologique à Baptiste SERIN :

In [ ]:
```
def distanceSerin(joueur2):
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: