

La fonction *sorted()* et la méthode *sort()*

- Pour trier , on peut utiliser la **fonction** *sorted()*.
Par exemple , pour trier une liste:

```
>>>sorted([25,6,15,4])  
[4,6,15,25]
```

On peut aussi utiliser la **méthode** de list : *sort*

```
>>>L=[25,6,15,4]  
>>>L.sort  
>>>L  
[4,6,15,25]
```

Pour un tri décroissant , on peut utiliser *reverse*:

```
>>>L.sort(reverse=True)  
>>>L  
[25,15,6,4]
```

La méthode *sort()* est seulement définie pour les listes et renvoie **None** .

La méthode *sort()* est seulement définie pour les listes et renvoie **None** .

Au contraire, la fonction *sorted()*s'applique à tout objet itérable et retourne une liste. .

```
>>>sorted({1:'Mbappé' , 2:'Giroud' , 3:'Griezman' , 4 : 'Messi'})  
[1,2,3,4]
```

- *sort* et *sorted* utilisent une fonction **key** qui permet d'affiner les critères de comparaison .
Soit une liste *d* qui contient les tuples (3-uplets) qui représentent une équipe de ligue 1 ,le nombre de buts marqués et le nombre de buts encaissés.

```
>>>d=[('PSG',38,15),('Marseille',34,10),('Lille',28,17)]  
>>>d.sort(key=lambda equipe: equipe[1])  
>>>d  
>>>[('Lille', 28, 17), ('Marseille', 34, 10), ('PSG', 38, 15)]
```

Ici, on a utilisé une fonction "anonyme" (d'où lambda) *equipe* qui retourne la clé du tri , c'est-à-dire le deuxième élément de chaque tuple de la liste *d*.

On peut bien sûr l'utiliser avec *sorted()*:

```
>>>d=[('PSG',38,15),('Marseille',34,10),('Lille',28,17)]  
>>>sorted(d,key=lambda equipe: equipe[1])  
[('Lille', 28, 17), ('Marseille', 34, 10), ('PSG', 38, 15)]
```

- On pouvait également écrire de manière explicite (à l'aide de **def**) une fonction *equipe*:

```
def equipe (d):  
    return d[1]
```

```
>>>d.sort(key=equipe)  
>>>d  
>>>[('Lille', 28, 17), ('Marseille', 34, 10), ('PSG', 38, 15)]
```

Ou avec *sorted*

```
>>>sorted(d,key=equipe)  
[('Lille', 28, 17), ('Marseille', 34, 10), ('PSG', 38, 15)]
```