

Exercice 1.8.1 - Solution :

Créer une liste de nombres

Créer un dico avec chacun des éléments de la liste et son nb d'occurrence

```
liste1=[1, 5, 1, 3, 4, 5]
```

```
dico1={}
```

```
for e in liste1:
```

```
    if (e in dico1):
```

```
        dico1[e] += 1
```

```
    else:
```

```
        dico1[e] = 1
```

```
print (dico1)
```

Exercice 1.8.1 - Solution :

Créer une liste de prenom

Créer un dico avec chacun des éléments de la liste et son nb d'occurrence

```
liste2=["Sylvie", "Eric", "Bernard", "Eric", "Catherine", "Sylvie"]
```

```
dico2={}
```

```
for e in liste2:
```

```
    if (e in dico2):
```

```
        dico2[e] += 1
```

```
    else:
```

```
        dico2[e] = 1
```

```
print (dico2)
```

Exercice 1.8.1 - Solution :

Ecrire une fonction qui trie une liste de nombres avec la méthode de tri par selection

```
def tri_nb(liste):  
    lg = len(liste)  
    for i in range (lg-1):  
        for j in range (i+1,lg):  
            if liste[j] < liste[i]:  
                # Echanger liste[i] avec liste[j]  
                tampon=liste[j]  
                liste[j]=liste[i]  
                liste[i]=tampon  
    return liste
```

```
liste3=[100, 25, 12, 3, 44, 57]
```

```
Li = tri_nb(liste3)
```

```
print(Li)
```

Exercice 1.8.1 - Solution :

Ecrire une fonction qui trie une liste de prenomms du plus court au plus long

```
def tri_str(liste):  
    lg = len(liste)  
    for i in range (lg-1):  
        for j in range (i+1,lg):  
            if len(liste[j]) < len(liste[i]):  
                # Echanger liste[i] avec liste[j]  
                tampon=liste[j]  
                liste[j]=liste[i]  
                liste[i]=tampon  
    return liste  
  
liste2=["Sylvie", "Eric", "Bernard", "Eric", "Catherine", "Sylvie"]  
Li = tri_str(liste2)  
print(Li)
```