

Exercice 1.10 - Solution :

Liste des employés

a) Initiales : pour chaque nom de la liste, il faut afficher le premier caractère du nom

b) Si un nom commence et se termine par la même lettre, il faut l'afficher

```
liste=['anna', 'bernard', 'marie', 'emile', 'joe']
```

```
print("\n Liste : ", liste)
```

```
for nom in liste:
```

```
    print("\n initiale : ", nom[0])
```

```
    debut=nom[0]
```

```
    fin=nom[-1]
```

```
    if debut==fin:
```

```
        print("\n nom qui commence et se termine par la même lettre: ", nom)
```

```
print("\n ----- ")
```

Exercice 1.10 - Solution :

c) Définir une fonction qui retourne une liste composée des initiales (1er caractère de chaque nom)

```
def initiales(Li):
```

```
    ini = []
```

```
    for e in Li:
```

```
        ini.append(e[0])
```

```
    return ini
```

```
LiA = initiales(liste)
```

```
print("\n Liste des initiales : ", LiA)
```

d) Définir une fonction qui affiche les noms commençant et se terminant par la lettre passée en paramètre

```
def f1(c, Li):
```

```
    for p in Li:
```

```
        if (p[0]==c) and (p[-1]==c):
```

```
            print("\n nom qui commence et se termine par la lettre {} : {}".format(v, p))
```

```
var = input("\n entrer char :")
```

```
f1(var, liste)
```

Exercice 1.10 - Solution :

Annuaire des employés

"""

a) Associer à

- l'équipe-1 la liste1 avec les noms : 'anna', 'bernard', 'marie-antoinette', 'emile'
- l'équipe-2 la liste2 avec les noms : 'Joe', 'michel'
- l'équipe-3 la liste3 avec les noms : 'frederique', 'sylvie'

"""

liste1=['anna', 'bernard', 'marie-antoinette', 'emile']

liste2=['joe', 'michel']

liste3=['frederique', 'sylvie']

di={}

di["equipe1"]=liste1

di["equipe2"]=liste2

di["equipe3"]=liste3

Exercice 1.10 - Solution :

b) Mettre tous les noms de l'équipe-1 en majuscule

```
Li = di["equipe1"]  
for i in range(len(Li)):  
    Li[i] = Li[i].upper()  
  
print("\n Liste mise en Majiscule : ", Li)
```

Exercice 1.10 - Solution :

c) Afficher l'employé qui a le nom le plus long ainsi que son équipe

Méthode-1: utiliser `Liste.sort(key = len)`

```
taille_finale=0
```

```
employe_final=None
```

```
equipe_finale=None
```

```
for eq in di:
```

```
    li = di[eq]
```

```
    li_triee = li.copy()
```

```
    li_triee.sort(key = len)
```

```
    emp = li_triee[-1]
```

```
    t_emp = len(emp)
```

```
    if t_emp > taille_finale:
```

```
        taille_finale = t_emp
```

```
        employe_final = emp
```

```
        equipe_finale = eq
```

```
print("\n employé qui a le nom le plus long : {}".format(employe_final) )
```

```
print("\n son équipe   : {}".format(equipe_finale) )
```

```
print("\n taille du nom : {}".format(taille_finale) )
```

Exercice 1.10 - Solution :

Méthode-2: utiliser la fonction tri_str créée dans un exercice précédent

```
from Module1 import tri_str # importer la fonction tri_str du fichier Module1.py
```

```
taille_finale=0
```

```
employe_final=None
```

```
equipe_finale=None
```

```
for eq in di:
```

```
    li_triee = tri_str(di[eq])
```

```
    emp = li_triee[-1]
```

```
    t_emp = len(emp)
```

```
    if t_emp > taille_finale:
```

```
        taille_finale = t_emp
```

```
        employe_final = emp
```

```
        equipe_finale = eq
```

```
print("\n employé qui a le nom le plus long : {}".format(employe_final) )
```

```
print("\n son équipe   : {}".format(equipe_finale) )
```

```
print("\n taille du nom : {}".format(taille_finale) )
```

Exercice 1.10 - Solution :

Méthode-3: Passer par un dico intermédiaire

```
from Module1 import tri_str      # importer la fonction tri_str du fichier Module1.py
```

```
taille_finale=0
```

```
employe_final=None
```

```
equipe_finale=None
```

```
di_inter = {}
```

```
for eq in di:
```

```
    li_triee = tri_str(di[eq])
```

```
    emp = li_triee[-1]
```

```
    di_inter[eq] = emp
```

```
for eq, emp in di_inter.items():
```

```
    t_emp = len(emp)
```

```
    if t_emp > taille_finale:
```

```
        taille_finale = t_emp
```

```
        employe_final = emp
```

```
        equipe_finale = eq
```

```
print("\n employé qui a le nom le plus long : {}".format(employe_final) )
```

```
print("\n son équipe   : {}".format(equipe_finale) )
```

```
print("\n taille du nom : {}".format(taille_finale) )
```

Exercice 1.10 - Solution :

d) Ecrire une fonction mettant les noms d'une equipe en majuscule ou en minuscule

```
def upp_low(equipe, choix):  
    if (choix != "upper") and (choix != "lower"):  
        print ("Error choix")  
        exit()  
  
    Li = di[equipe]  
    for i in range (len(Li)):  
        if choix == "upper":  
            Li[i] = Li[i].upper()  
        else:  
            Li[i] = Li[i].lower()  
  
    print("\n équipe AVANT mise en miniscule : ", di["equipe1"])  
    upp_low("equipe1", 'lower')  
    print("\n équipe APRES mise en miniscule : ", di["equipe1"])  
  
    print("\n équipe AVANT mise en majuscule : ", di["equipe1"])  
    upp_low("equipe1", 'upper')  
    print("\n équipe APRES mise en majuscule : ", di["equipe1"])
```


Exercice 1.10 - Solution :

e) Ecrire une fonction qui retourne l'employé qui a le nom le plus long ainsi que son équipe

```
def func(dico):
    taille_finale=0
    employe_final=None
    equipe_finale=None
    for eq in dico:
        Li_triee = tri_str(dico[eq])
        emp = Li_triee[-1]
        t_emp = len(emp)
        if t_emp > taille_finale:
            taille_finale = t_emp
            employe_final = emp
            equipe_finale = eq
    return employe_final, equipe_finale, taille_finale

emp, equipe, taille = func(di)

print("\n employé qui a le nom le plus long : {}".format(emp) )
print("\n son équipe   : {}".format(equipe) )
print("\n taille du nom : {}".format(taille) )
```