Le dictionnaire en python

Un dictionnaire en python est une liste de couples (clef : valeur) :

```
print ("Dictionnaire avec python ")
data = {}   # initialiser dictionnaire

data = {"Jean Pierre" : 25 , "Bertrand" : 49 , "Jérome" : 50 , "David" : 12}
```

CLEF VALEUR

Note : clef et valeur peuvent être du texte ou des nombres

Manipulation du dictionnaire :

```
#
# Utilisation d'un dictionnaire - Résumé
#
print ("Dictionnaire avec python ")
data = {}   # initialiser dictionnaire

data = {"Jean Pierre" : 25 , "Bertrand" : 49 , "Jérome" : 50 , "David" : 12 , "Jerémy" : 32 , "Antoine" : 69 , "Luc" : 55 , "Emilie" :37 , "Sébastien" : 13 }
```

Manipulation d'un dictionnaire :

```
# Afficher tout le dictionnaire :
print (data)

# Ajouter des valeurs à un dictionnaire :
data["Isabelle"] = 35


# On veut afficher toutes les clefs du dictionnaire :
for clef in data:
    print (clef)

# On veut afficher toutes les valeurs du dictionnaire :

for clef in data:
    print (data[clef])

# On veut afficher le couple clef , valeur

for clef in data:
    print (clef , data[clef])

# Vérifier si une clef est présente dans un dictionnaire

if "Bertrand" in data:
    print ("La clef Bertrand est bien présente")

# Supprimer une clef (et valeur associée) à un dictionnaire
del data["Isabelle"]
```

On testera le code pour s'approprier la notion de dictionnaire --> on rajoutera des couples de clefs/valeur

```
# On veut afficher uniquement les membres de plus de 30ans :

print ("Membres de plus de 30 ans : ")
for clef in data:
    if data[clef] >= 30:
        print (clef , data[clef])

# On veut additionner les ages des membres de moins de 30ans :
somme = 0
print ("Somme des âges des membres de moins de 30 ans : ")
for clef in data:
    if data[clef] < 30:
        somme = somme + data[clef]

print ("Somme =", somme )
```

Tester le code et vérifier qu'il fonctionne correctement.

Opérations de tri :

```
56  print ("Tri par clef :")
57  for clef in sorted(data,reverse=False):
58      print (clef)
59  print ()
60
61  print ("Tri sur valeur ou clef :")
62  for clef,valeur in sorted(data.items(), key=lambda x: x[1],reverse=True ):
63      print (clef,valeur)
```

X[0] : le tri se fait sur la clef
X[1] : le tri se fait sur la valeur
Reverse = True : le classement est inversé

Exercices d'application

1- On veut créer un dictionnaire avec comme clef le nom du client et comme valeur le montant de son compte bancaire.

banque = {nom1 : valeur1 , nom2 : valeur2 , nom3 : valeur3 …. etc }

- Créer un dictionnaire banque avec une dizaine de clients.

- Faire une fonction affiche_clients(banque) qui va afficher tous les clients de la banque

- Faire une fonction affiche_compte(client,banque) qui affiche le montant du compte du client. Attention, si le client n'existe pas, le programme ne doit pas planter, vous devez tester si le client existe.

- Faire une fonction ajout(client ,depot, banque) qui devra :
  - Si le nom existe déjà, ajouter une valeur dépôt au montant du compte
  - Si le nom n'existe pas, ajouter un nouveau compte client (nom – dépôt)

- Faire une fonction supprime(client,banque) qui supprime le client de la banque.
Attention, si le client n'existe pas, le programme ne doit pas planter !

2- On veut maintenant créer un dictionnaire « epargne » qui contient le nom du client et le montant de son épargne.
Important : les clients de la banque n'ont pas tous une épargne.

- Créer un dictionnaire epargne avec 4 clients (à reprendre parmi ceux de banque)

Faire une fonction tester(client,epargne) qui renvoie True si le client a un compte epargne et False sinon.

Faire une fonction total(client,banque,epargne) qui affiche le montant total du compte client (compte_courant + epargne si existe)

Faire une fonction classe_client (banque) qui classe les clients par ordre alphabetique
Faire une fonction classe_montant(banque) qui classe les clients en fonction du montant de leur compte en banque.