Informatique SV L2
TP 3

Collections en Python

Python définis 3 types de collection qui permet dans stocker des objets (nombre, chaine de caractère,ect...) :

- La chaine de caractère : Collection de caractère ordonnée accesible par la position du caractère
- La liste : Collection d'objets ordonné accesible par la position du caractère
- Le dictionnaire : Collection d'objets non ordonné accesible par une clé

Example d'une liste :

```
notes=[10,14,18,8,9,15]    #initialisation de la liste
print notes[2]             #affiche le 3émé élement (ici 18)
notes.append(13)           #ajoute une note à la liste
n=notes[0]                 #affecte la 1er note à la varaible n
```

Example d'un dictionnaire :

Ici on crée un annuaire avec comme clé un nom et comem valeur un numéro de téléphone. Ainsi on associe un nom à un numéro de téléphone.

```
annuaire={'Sandrine' :  '0145478596' , 'Mathieu' : '0354789568', 'Tom':'0541257136'}
print annuaire['Mathieu']          #affiche 0354789568
annuaire['Laure'] = '0547854874'   #ajoute un numéro associé à un nouveau nom
```

Enumération d'une collection

Pour énumérer les éléments d'une liste on peut utiliser l'instruction while :

```
notes=[10,14,18,8,9,15]    #initialisation de la liste
i=0
while i<len(notes) :       #len() renvoit le nombre d'élément dans la liste
     print notes[i]        #affiche la i-éme note
     i+=1
```

Mais on peut aussi utiliser l'instruction for pour énumerrer une collection en général (chaine de caractère, liste, dictionnaire) :

```
notes=[10,14,18,8,9,15]    #initialisation de la liste
for n in notes:
     print n               #n prend succesivement les valeurs de la liste notes
```

Créer une collection vide

```
chaine_vide=''             #crée une chaine de caractère vide
liste_vide=[]              #crée une liste vide
dictionnaire_vide={}       #crée un dictionnaire vide
```

1 Statistique sur des nombres et sur un texte

1.1 Statistiques sur des nombres

Écrire des fonctions qui calculs la somme,la moyenne, le minimum, le maximum à partir d'une liste de nombres.

Tester ces fonctions avec le programme principal contenu dans le fichier
http ://www.ief.u-psud.fr/~kaisser/cours/sv/stat_notes.py

1.2 Statistique sur un texte

1.2.1 Compter le nombre d'occurrence

Écrire une fonction occurrences(caractere,texte), qui renvoi le nombre d'occurrence d'un caractère dans un texte. Exemple : Trouver le nombre de *a* dans la phrase « Mon ordinateur est planté ». Tester cette fonction.

1.2.2 Compter le nombre de caractère, ligne et mot

Écrire une fonction nbLigne(texte), qui compte le nombre de ligne dans un texte, et une fonction nbMot(texte) qui compte le nombre de mot (séparé par un espace). Aide : une ligne est séparé par le caractère '\n'

Tester ces fonctions avec le texte contenu dans le fichier http ://www.ief.u-psud.fr/~kaisser/cours/sv/stat_chaine.py

1.2.3 Compter le nombre d'occurrence pour chaque caractère

1) Soit le programme suivant qui stocke dans un dictionnaire le nombre de 'a' dans un texte :

```
dict = {}                              #création du dictionnaire
texte = « Mon ordinateur est planté »
for c in texte :
    if c=='a' :
         if dict.has_key('a') :        #si la clé a est déjà dans le dictionnaire
              dict['a']+=1             #on ajoute une occurence de plus
         else :
              dict['a']=1              #sinon c'est la premiere fois qu'on voit un 'a'
#Affiche le dictionnaire
for cle in dict :
    print cle,' : ',dict[cle]
```

Note : La méthode d.has_key(cle) (où d est le dictionnaire) renvois vrai si une clé existe déjà, faux sinon.

2) En s'inspirant du programme en 1), écrire un programme qui compte le nombre d'occurrence pour chaque caractère et qui stocke ce nombre d'occurence dans un seul dictionnaire. Exemple avec « Mon ordinateur est planté », on doit trouver le nombre de a, de b, de c, ect...

3) Ecrire une fonction toutOccurrence(texte), qui reprend ce programme. Tester la fonction avec le texte de stat_chaine.py

2 Découper une phrase en mots

2.1 Créer une liste de mots

2.1.1 Afficher les mots d'un texte

Écrire un programme qui affiche tous les mots d'un texte. Ex : Avec « Mon ordinateur est planté » :

'Mon'

'ordinateur'

'est'

'planté'

Note : On utilisera texte[j :k] qui permet d'obtenir une chaine de caractère qui commence à la position j et qui finit à la position k.

2.1.2 Créer la liste de mot

Utiliser le programme précédent pour définir une fonction liste_mot(texte) qui renvoit une liste de mot à partir d'un texte.

Avec « Mon ordinateur est planté », on doit obtenir la liste ['Mon', 'ordinateur', 'est','planté']

Tester cette fonction avec le texte de l'exercice 1.2.2

Note : on utilisera la fonction l.append(element) (où l est une liste), pour ajouter un nouvelle élement dans la liste.

2.2 Trier une liste

Écrire une fonction trier(liste) qui tri une liste par ordre croissant, et retourne une nouvelle liste triée. La fonction retournera une nouvelle liste. Avant de commencer à écrire la fonction, pensez à la strategie de tri que vos souhaitez utiliser.

Tester la fonction avec la liste de mot généré précédemment.

Utiliser également cette fonction pour écrire une fonction mediane(liste) qui calcul la médiane des notes de l'exercice 1.1

Remarque : la médiane est la valeur qui est situé au milieu d'une suite de nombre trié dans un ordre croissant. Si il y a un nombre paire de valeur, on prend la moyenne entre les deux valeurs du milieu. Ex : Avec 1 ; 2 ; 5 ; 7 ; 8 la médiane est 5. Avec 1 ; 4 ; 4 ; 5 ; 8 ; 9 la médiane est $\frac{4+5}{2} = 4,5$