TP n ° 7
Les Fichiers

Avant de commencer

Pour ce TP vous allez avoir besoin de certains fichiers texte ainsi que de bouts de programmes. Ils sont disponibles dans le dossier de partage. Il vous suffit d'aller les récupérer, de les copier et de les coller dans un dossier de votre compte qui pourrait s'appeler par exemple `TP7_fichiers`. Les programmes que vous écrirez alors seront tous enregistrés dans ce dossier. Enfin lors de la première exécution il vous faudra lancer avec Pyzo `Exécuter / Démarrer le script`

C'est parti !

Exercice 1 : *Traitement d'un fichier de mesures biologiques*

On considère le fichier texte `mesuresBCPST.txt` contenant les masses en grammes et les tailles en centimètres de follicules de *laurier rose, glycine blanche, glycine violette* et *bignone*. Deux valeurs successives sont séparées par un espace et chaque ligne se termine par `'\n'`, caractère non imprimable de fin de ligne.

`mesuresBCPST.txt`

```
masse taille espece
28.6 19.1 glycine blanche
20.6 14.8 glycine blanche
............
26.4 18.7 glycine violette
............
10.9 12.8 bignone
............
4.9 15.3 laurier rose
```

1. Écrire un script affichant les différentes lignes du fichier texte `mesuresBCPST.txt`
2. Afin de ne pas confondre l'espace entre les différentes colonnes « masse, taille, espèce » et l'espace dû au nom de la fleur (glycine blanche par exemple) on va reformater les lignes de la manière suivante :

   `masse\ttaille\tespèce\n`

   Écrire un script Python, créant un fichier `mesures.txt` dans lequel vous recopierez les données du fichier `mesuresBCPST.txt` reformatées comme demandé.

   Procédure :
   - ⋆ On va parcourir le fichier ligne par ligne.
   - ⋆ Sur une ligne donnée on va transformer la chaine de caractère en la liste des mots avec `.split(' ')` on appellera tout simplement la liste `liste`.
   - ⋆ Si cette liste est de longueur 4, c'est que la fleur a un nom en deux mots. On va fusionner les deux mots de la manière suivante :

     `fleur=liste[2]+' '+liste[3]`

     On crée alors une nouvelle liste `listebis=[liste[0],liste[1],fleur]` qu'on transforme en chaine de caractères séparés par une tabulation avec `'\t'.join(listebis)`
   - ⋆ Sinon on transforme directement la ligne de la manière suivante

     `newline='\t'.join(ligne.split(' '))`
   - ⋆ Enfin on écrira cette nouvelle ligne dans le fichier créé pour cela.
3. Écrire une fonction `lecture(fichier)` qui ouvre un `fichier` structuré comme `mesures.txt` et qui retourne deux listes de flottants : l'une contenant les masses et l'autre contenant les tailles. Il faut penser à la conversion des chaînes de caractères comme `'28.6'` en flottant avec `float('28.6')`.

   Votre fonction débutera par ces lignes

   ```
   1 def lecture(fichier):
   2     f=open(fichier,'r')
   ```

   vous l'exécuterez avec la ligne suivante

   ```
   >>> lecture('mesures.txt')
   ```

4. Ajouter au fichier `mesures.txt` une ligne formatée comme les précédentes avec la masse moyenne et la taille moyenne des follicules, écrites avec deux décimales, puis le mot `'Moyennes'`. On pourra commencer par écrire une fonction `moyenne` qui retourne la moyenne des éléments d'un tableau de flottants.
5. Copier et coller la fonction `diagramme` fournie et exécuter la sur le fichier `mesures.txt`
6. Créer à partir du fichier `mesures.txt` un fichier `mesures_lauriers.txt` contenant uniquement les mesures des masses et des tailles des follicules de lauriers roses.
7. Déterminer la masse moyenne et la taille moyenne des follicules de lauriers roses.
8. Copier et coller la fonction `bilan_regression`, lire la documentation et compléter les lignes manquantes.
9. Exécuter la fonction `bilan_regression` avec le fichier `mesures_lauriers`. Ouvrir avec le bloc note le fichier `regression.txt` et vous devriez obtenir quelque chose comme ceci :

regression.txt

```
Totaux
sum(x) : 312.80, sum(y) : 986.60, sum(x^2) : 1451.04, sum(y^2) : 14006.96, sum(xy)
      : 4470.36
Moyennes
moyenne x : 4.41,moyenne y 13.90
Ecarts-types
sigma x : 1.01, sigma y : 2.05
Covariance de x et de y : 1.74
Coefficient de corrélation linéaire de x et y : 0.84
Equation de la droite de régression de y en x : y=1.70x+6.42
```

Exercice 2 : *Format `fasta` de fichiers de séquences génétiques*

1. La fonction suivante prend pour paramètres formels deux chaînes de caractères. Tester cette fonction pour les paramètres effectifs `sequence='an'` et `chaine='Bonne anniversaire'`.
   Quel est le traitement réalisé par cette fonction ? Expliquer le fonctionnement de l'algorithme.

```
1  def f(sequence,chaine):
2      lchaine,lsequence = len(chaine),len(sequence)
3      i = 0
4      while i<lchaine-lsequence+1:
5          j = 0
6          while j<lsequence and chaine[i+j] == sequence[j]:
7              j += 1
8          if j== len(sequence):
9              return i
10         i += 1
11     return None
```

2. Ecrire une fonction `recherche_fasta(fichier,sequence)` qui recherche une séquence de nucléotides dans un fichier au format `fasta`[1] contenant une séquence génétique. Faire un test avec le fichier `exemple_fasta.txt`.
   On pourra se servir des fonctions `sous_mots` créées dans le TP précédents.

exemple_fasta.txt

```
1  >Sequence mutee - gene KCNJ13
2  ATGGACAGCAGTAATTGCAAAGTTATTGCTCCTCTCCTAAGTCAAAGATA
3  CCGGAGGATGGTCACCAAGGATGGCCACAGCACACTTCAAATGGATGGCG
```

1. Pour plus d'infos sur le format fasta voir http://fr.wikipedia.org/wiki/FASTA_format_de_fichier