

Langages de scripts (LS4)

TP 6 – Manipulation de fichiers textuels en PYTHON

Exercice 1 – Statistiques sur un fichier textuel

1. Définir une classe `analyse` avec une fonction d'initialisation qui, prenant une chaîne de caractères, ouvre le fichier de ce nom. Si le fichier n'existe pas, un message d'erreur doit s'afficher. Lorsqu'une instance de la classe `analyse` est détruite, le fichier correspondant doit être fermé.
2. Écrire une méthode `words` calculant la liste des mots du fichier.
3. Écrire une méthode `count_words` comptant les occurrences de tous les mots dans le texte (une table associant à chaque mot le nombre de ses occurrences).
4. Écrire une méthode `sum_up` qui affiche le nombre de mots différents apparaissant dans le texte, puis la liste de ces mots ainsi que leurs nombres d'occurrences.

Exercice 2 – Cent mille milliards de poèmes de Raymond Queneau

Le fichier `~monitLS4/vers-queneau.txt` contient une séquence de 14 groupements de 10 vers séparés par des lignes vides.

1. Écrire un script PYTHON chargeant le fichier `vers.txt` dans un dictionnaire associant aux entiers de 0 à 13, la liste des vers de chaque groupement.
2. Modifier le script pour générer un poème de 14 vers en prenant un vers aléatoirement dans chaque groupement. Utiliser pour cela le module `random` de la bibliothèque standard de PYTHON (<http://docs.python.org/lib/module-random.html>). Ne pas oublier de signer Raymond Queneau à la fin de votre poème.

Exercice 3 – Une infinité de poèmes de Victor Hugo ?

Le fichier `~monitLS4/vers-hugo.txt` contient des vers de Victor Hugo, mais ils n'ont pas été classés comme dans l'exercice précédent. Nous allons *essayer* de générer des poèmes par un autre processus.

1. Écrire un script PYTHON qui charge l'ensemble des vers du fichier dans une liste. On filtrera les entrées suivant le critère suivant : un vers est formé d'au moins 20 caractères et le rapport entre sa taille et le nombre de caractères en capitale est supérieur à 5.
2. Modifier le script pour qu'il génère un poème composé de $2n$ vers en utilisant l'algorithme suivant (n étant un paramètre du script) :
 - (a) Choisir un vers s_1 au hasard et l'afficher.

- (b) Tant que $i < 2n$, choisir le vers s_{i+1} dans la liste qui est le plus proche de s_i par rapport à la distance :

$$d(s, t) = \frac{20}{1 + d_1(s, t)} + d_2(s, t)$$

où $d_1(s, t)$ désigne le nombre de lettres identiques à la fin de s et de t , et $d_2(s, t)$ la valeur absolue des différences des longueurs de s et de t . Afficher s_{i+1} .

Exercice 4 – Modifications de l'Éducation Sentimentale

Nous allons travailler sur le fichier `~monitLS4/educ.txt`. Pour cet exercice et le suivant, utiliser le module `re` (<http://docs.python.org/lib/module-re.html>).

1. Ce texte est formé de chapitres annoncés par une ligne contenant `Ch pitre N` où N est le numéro du chapitre. Écrire un script chargeant le fichier en mémoire et créant un fichier pour chaque chapitre intitulé `ch pitre-i.txt`.
2. Modifier le script pour qu'il remplace le prénom "Frédéric" par "Gustave".
3. Modifier le script pour qu'il remplace tous les mots commençant par "aim" suivi d'un suffixe par le mot "détest" suivi du même suffixe.
4. Modifier le script pour qu'il extraie les dialogues de chaque chapitre dans un fichier `ch pitre-di logues-i.txt`.

Exercice 5 – Résolution naïve de mots croisés

Le fichier `~monitLS4/french` contient une longue liste de mots français.

1. Écrire un script chargeant le fichier `french` en mémoire sous la forme d'une liste.
2. Écrire une fonction prenant en paramètre une expression régulière et cherchant l'ensemble des mots du dictionnaire appartenant au langage qu'elle décrit.
3. Modifier le script pour qu'il charge un fichier dont le nom est passé en paramètre au script et contenant une grille rectangulaire dont chaque ligne et chaque colonne correspond à un mot à trous. Voici deux exemples :

lu.	donne	lue	.uo	donne	duo
..		ns	..		ns
			n.e		nie

Ces grilles seront stockées sous la forme d'une liste d'expressions régulières. Si la grille n'a pas la bonne forme, lancer une exception `inv lid_grid`.

4. Écrire une fonction qui recherche une solution pour la grille. Attention, le calcul pourrait être très long si la grille d'entrée est trop compliquée (les exemples se trouvent assez rapidement).