

Master 1 Humanités numériques – Algorithmique et programmation

TD 6 : Fichiers textes

J. Darmont – <http://eric.univ-lyon2.fr/~jdarmont/>

L'objectif de ce TD est d'extraire le dictionnaire d'un fichier texte (l'ensemble des mots composant le texte). Les exercices sont à formuler en langage algorithmique textuel, puis en Python (Blockly ne prend pas en charge les fichiers). Ne pas oublier la déclaration des variables en algorithmique. Tester l'exécution avec Python.

Exercice 1 : Lecture d'un fichier

1. Télécharger le fichier lorem-ipsuM.txt¹ et le placer dans le répertoire de vos programmes Python.
2. En algorithmique, on suppose que la procédure afficher (une liste) écrite au TD 5 est disponible. En Python, importer la procédure afficher du module listeMod créé au TD 5.
3. Ouvrir le fichier f correspondant à lorem-ipsuM.txt en lecture.
4. Lire chaque ligne du fichier f et les placer dans une liste listeLignes.
5. Fermer le fichier f.
6. Afficher la liste listeLignes. Vous devez obtenir autant de lignes que le nombre de paragraphes dans le fichier (5).

Exercice 2 : Traitement du contenu du fichier

On souhaite extraire de la liste des lignes du fichier les mots qui les composent pour constituer un dictionnaire. Pour cela, ajouter à l'algorithme et au programme précédents les traitements suivants.

Pour chaque ligne de listeLignes :

1. supprimer successivement les points, les virgules et les retours charriot (caractère spécial `\n`) de la ligne en les remplaçant par une chaîne vide ;
2. transformer la ligne en une liste de chaînes nommée lesMots ;
3. afficher la liste lesMots pour vérification. Tester. Vous devez obtenir autant de listes de mots que le nombre de lignes.
4. remplacer l'affichage de lesMots par son ajout dans la liste de chaînes dico, qui aura dû être initialisée vide avant la boucle « pour chaque ligne ». L'union de listes se fait avec le symbole \cup en algorithmique et le + en Python.

¹ <http://eric.univ-lyon2.fr/~jdarmont/docs/lorem-ipsuM.txt>

- Après la boucle « pour chaque ligne », afficher la liste dico. Remarque ?

Afin de dédoublonner les mots du dictionnaire, nous allons utiliser des sous-programmes tout faits :

- Algorithmique : procédure Dédoublonner(uneListe) ;
 - Python : fonction set(uneListe), qui transforme une liste en *ensemble* sans doublon. Pour retransformer l'ensemble en liste, il faut lui appliquer la fonction list(). Il est possible d'appliquer une fonction à une autre fonction, par exemple : uneListe = list(set(uneListe)).
- Dédoublonner, puis trier la liste dico.
 - Afficher la liste dico.

Exercice 3 : Écriture d'un fichier

- Toujours à la suite de l'algorithme et du programme précédents, ouvrir un nouveau fichier f2 nommé dico.txt en écriture.
- Ecrire chaque mot du dictionnaire dico dans le fichier f2.
- Fermer le fichier f2.
- Consulter le fichier dico.txt pour vérifier que la liste des mots y est bien sauvegardée.

Questions subsidiaires (Python)

- Transformer le programme développé dans les exercices 1 à 3 en une fonction nommée extraireDictionnaire prenant en paramètres le nom du fichier texte à traiter en entrée et le nom du fichier dictionnaire en sortie.
- Placer la fonction extraireDictionnaire dans un module nommé dicoMod.

Correction

Algorithme lireEtTransformer

```
Var f, f2 : Fichier
Var ligne : Chaîne
Var lesLignes, dico : Liste de Chaînes
```

Début

```
{Lecture et affichage du fichier d'entrée}
f ← Ouvrir("lorem-ipsuim.txt", "lecture")
Lire(f, ligne)
Tant que non FinDeFichier(f) faire
    lesLignes( ) ← ligne
    Lire(f, ligne)
Fin tant que
Fermer(f)
afficher(lesLignes)
{Extraction des mots}
dico ← ( )
Pour ligne dans lesLignes faire
    ligne ← Remplacer(ligne, ".", "")      {Suppression des points}
    ligne ← Remplacer(ligne, ",", "")     {Suppression des virgules}
    ligne ← Remplacer(ligne, "\n", "")    {Suppression des retours chariot}
    lesMots ← Découper(ligne, " ")       {Extraction des mots}
    dico ← dico ∪ lesMots
Fin pour
afficher(dico)
{Dédoublonnage et tri du dictionnaire}
Dédoublonner(dico)
Trier(dico)
afficher(dico)
{Ecriture du dictionnaire dans un fichier de sortie}
f2 ← Ouvrir("dico.txt", "écriture")
Pour ligne dans dico faire
    Écrire(f2, ligne)
Fin pour
Fermer(f2)
```

Fin

```

from listeMod import afficher

# Lecture et affichage du fichier d'entrée
f = open("lorem-ipsuM.txt", "r")
lesLignes = f.readlines()
f.close()
afficher(lesLignes)

# Extraction des mots
dico = [ ]
for ligne in lesLignes:
    ligne = ligne.replace(".", "") # Suppression des points
    ligne = ligne.replace(",", "") # Suppression des virgules
    ligne = ligne.replace("\n", "") # Suppression des retours chariot
    lesMots = ligne.split(" ") # Extraction des mots
    dico = dico + lesMots
afficher(dico)

# Dédoublonnage et tri du dictionnaire
dico = list(set(dico))
dico.sort()
afficher(dico)

# Ecriture du dictionnaire dans un fichier de sortie
f2 = open("dico.txt", "w")
for mot in dico:
    f2.write(mot + "\n")
f2.close()

# Questions subsidiaires
# Placer la fonction ci-dessous dans un fichier nommé dicoMod.py
def extraireDictionnaire(nomFichierEntree, nomDicoSortie):
    # Reproduire ci-dessous tout le code ci-dessus :)
    # Dans la section "Lecture et affichage du fichier d'entrée",
    # remplacer f = open("lorem-ipsuM.txt", "r") par
    f = open(nomFichierEntree, "r")
    # Dans la section "Ecriture du dictionnaire dans un fichier de sortie",
    # remplacer f2 = open("dico.txt", "w") par
    f2 = open(nomDicoSortie, "w")

# Exemple d'utilisation
from dicoMod import extraireDictionnaire
extraireDictionnaire("lorem-ipsuM.txt", "dico-lorem.txt")

```