TD 7 : Lecture et écriture dans un fichier

# Lecture d’un fichier .txt ou .csv

Pour lire ou écrire dans un fichier, on doit d’abord créer un « objet-fichier » par la commande open, dont le type est \_io.TextIOWrapper .

>>> f=open(os.listdir()[0])

>>> type(f)

<class '\_io.TextIOWrapper'>

Bien noter qu’il est **indispensable** de refermer lorsque la lecture ou la transformation en écriture est terminée, comme on le fait dans les exemples au-dessous.

La fonction open a deux arguments : le nom du fichier (ou le chemin menant au fichier, chemin absolu ou relatif), le mode d’ouverture du fichier (lecture 'r', écriture 'w', ajout en écriture 'a' – pour append, ou lecture et écriture 'r+') dont la valeur par défaut est 'r' (read).

* Ouverture/fermeture et lecture complète :

>>> f=open('Exemple.txt')

>>> f.read()

'riri\nfifi loulou\ndonald\n'

>>> f.close()

* Ouverture/fermeture et lecture ligne par ligne :

|  |  |
| --- | --- |
| >>> f=open('exemple.txt','r')  >>> f.readline()  'riri\n'  >>> f.readline()  'fifi loulou\n' | >>> f.readline() ##la fin de fichier se signale ##par une ligne vide  ''  >>> f.close() #note on a omis de lire la dernière ligne |

* Création/fermeture et écriture :

>>> f=open('nouveau.txt','w') #paramètre 'w’ nécessaire car égal à 'r' par défaut

>>> f.write('titi')

>>> f.write('rominet\n') #insérer \n si on souhaite un changement de ligne

>>> f.write('Fin')

>>> f.close()

Note : pour stocker des objets créés dans un programme Python, il existe le module pickle qui permet d’enregistrer les objets, ainsi que leur attributs, et non pas seulement leurs valeurs, comme on peut le faire avec les méthodes ci-dessus [<https://docs.python.org/3.3/library/pickle.html#module-pickle>]

**Q6.1.** Créer à l’aide d’un tableur, un tableau de valeurs quelconques, flottantes ou entières, en utilisant éventuellement des formules. Enregistrer ce tableau dans un fichier au format .csv (« comma separated values »). Ouvrir à l’aide de Python le fichier créé et expliciter ce qui code la structure de tableau.

Inversement, écrire un script Python qui génère une liste de 101 abscisses (entières, de 0 à 100) et une liste de 101 ordonnées, choisies aléatoirement dans l’intervalle et qui écrit ces valeurs dans un fichier, d’une façon compatible avec le format .csv, en indiquant des titres de colonnes (« abscisses » et « ordonnées »).

Vérifier la lisibilité du fichier créé en demandant son ouverture à partir d’un tableur.

Ouvrir à l’aide du bloc-notes le fichier TP16\_titrage.txt. Ce fichier correspond aux données utilisées dans l’exercice 1 du DS3 et contient des informations sur le titrage d’un acide. La première ligne de ce fichier contient la description du titrage, la seconde ligne indique le nombre de mesures effectuées, les lignes suivantes indiquent, pour chaque prise de mesure, le volume de soude versé en mL et la mesure de pH correspondante.

Écrire un script Python qui lit les informations dans ce fichier et qui écrit, dans un fichier .csv ayant pour nom le descriptif de l’expérience, ces informations sous la forme deux intitulées « Volume de soude versé (mL) » et « pH mesuré », la virgule étant codée par une virgule et non un point.

Écrire un script Python demandant à l’utilisateur une liste de mots (saisie en une fois, chaque mot étant séparé de son suivant par une virgule) et renvoyant le nombre d’occurrences de ce mot dans un fichier (indépendamment de la casse et du fait que le mot soit « coupé » par un passage à la ligne). On utilisera des fichiers de tests pour valider le script.

Écrire un script Python prenant en argument le nom d’un fichier texte et renvoyant le nombre d’occurrences de chaque lettre de l’alphabet dans ce texte. On pourra utiliser le fichier Notre-Dame de Paris.txt et comparer les fréquences obtenues aux fréquences données sur la page <https://fr.wikipedia.org/wiki/Fr%C3%A9quence_d'apparition_des_lettres_en_fran%C3%A7ais> .