

## Séance 6: FICHIERS TEXTES

L1 – Université Nice Sophia Antipolis

Objectifs:

- savoir écrire un fichier texte
- savoir lire un fichier texte
- savoir utiliser les chaînes de formatage

### Exercice 1 (Watson, ★)

On considère le programme suivant

```
f_out = open('watson.txt', 'w', encoding='utf-8')
f_out.write('Élémentaire, mon cher!')
f_out.close()
```

1. Que fait ce programme ?
2. Que se passe-t-il si le fichier `watson.txt` existe déjà avant d'exécuter le programme ?
3. Que se passe-t-il si on oublie la ligne `f_out.close()` ?
4. Que se passe-t-il si on remplace le paramètre optionnel par `encoding='latin-1'` ? Et si on oublie ce paramètre ?
5. Comment faire pour ajouter le texte ci-dessus à la fin du fichier si le fichier contient déjà du texte ?
6. Que se passe-t-il si le fichier `'watson.txt'` est protégé en écriture ?

□

### Exercice 2 (Le temps qui passe, ★)

Définissez une fonction `jour(s)` qui crée un fichier `s` dans lequel elle écrit ligne après ligne les différentes heures de la journée, à intervalle de 5 minutes. Le fichier ressemblera à

```
00:00
00:05
00:10
...
23:50
23:55
```

Indication : `'{:*>10}'.format(42)` renvoie `'*****42'`, c'est à dire une chaîne de 10 caractères représentant le nombre 42 aligné à droite en utilisant le caractère `*` comme caractère de remplissage. □

### Exercice 3 (Afficher un fichier, ★)

1. Écrivez une fonction `affiche_fichier(s)` qui lit le fichier nommé `s` et l'affiche.  
RAPPEL : On peut lire un fichier `f` de quatre manières distinctes :
  - d'un seul coup tout le contenu sous la forme d'une longue chaîne avec la méthode `read`.
  - d'un seul coup sous la forme d'une liste de lignes avec la méthode `readlines`.
  - une ligne à la fois avec la méthode `readline`.

— avec une boucle `for` en itérant sur le descripteur du fichier.

2. Que se passe-t-il si l'on demande d'afficher un fichier qui n'existe pas ?
3. Modifiez votre fonction pour faire afficher le message "fichier non trouvé".

□

#### Exercice 4 (Nombre de lignes, \*)

Définissez la fonction `nb_lignes(s)` qui renvoie le nombre de lignes du fichier `s`, ou `-1` si le fichier `s` n'existe pas ou ne peut pas être lu.

□

#### Exercice 5 (Mise en majuscule, \*\*)

On suppose que l'on a créé un fichier-texte contenant plusieurs paragraphes en français. Programmez une fonction `majuscules(f)`, sans résultat, fonctionnant de la manière suivante. Elle prend en entrée une chaîne de caractères `f` représentant un nom de fichier, comme `'texte.txt'`.

1. La fonction va commencer à fabriquer un nouveau nom de fichier `f1` identique au précédent, mais en ajoutant `-maj` au nom du fichier (donc ici `f1` sera `texte-maj.txt`). Définissez la variable `f1` en fonction de la variable `f`. On rappelle que la méthode `split` permet d'obtenir la liste `['texte', 'txt']` à partir de la chaîne `'texte.txt'`.
2. Elle va ensuite ouvrir `f` en lecture et `f1` en écriture, pour envoyer tout le texte de `f` dans `f1`, en le transformant au passage en MAJUSCULES. Elle ferme ensuite les deux fichiers.

□

#### Exercice 6 (Statistiques, \*\*\*)

Définissez une fonction `statistiques(L)` qui prend en argument une liste `L` de notes entre 0 et 20 et qui affiche des statistiques sur ces notes au format suivant. Vous utiliserez la méthode `format` pour ne pas avoir à faire l'alignement.

```
Note la plus basse :      3
Note la plus haute :     19
Moyenne :                 9.83
Taux d'admission :       40.2%
```

□