

Écrire et lire dans un fichier externe

Il est possible de lire ou d'écrire des données dans un fichier sur le disque dur.

La première des choses à faire est d'ouvrir ton fichier texte. Pour cela utilise la méthode `open()` :

```
monFichier = open(nomdufichier, mode d'ouverture)
```

monFichier est un objet qui possède des attributs et des méthodes.

Pour manipuler le fichier texte, il faut manipuler l'objet **monFichier**.

Il existe 3 modes d'ouverture :

- **'r'** : ouverture en lecture,
- **'w'** : ouverture en écriture. Le contenu existant du fichier est écrasé mais s'il n'existe pas, il est créé.
- **'a'** : ouverture en écriture en mode ajout. Dans ce cas, tu peux écrire à la fin du fichier sans écraser l'ancien contenu du fichier. Si le fichier n'existe pas, il est créé.

Une fois le fichier ouvert, il est possible **d'écrire** dedans (à l'aide de la méthode **write()**) ou de **lire** son contenu (à l'aide de la méthode **read()**). Quand tu n'utilises plus le fichier pense à le fermer avec la méthode **close()**.

Teste le programme **fichier1.py** suivant :

```
nom=input('Entre un nom : ')
monFichier=open('fichier.txt','a')
monFichier.write(nom)
monFichier.close()
print('fichier enregistré')
```

Le répertoire (dossier) contenant le script **fichier1.py** doit maintenant contenir un fichier « **fichier.txt** ».

Ouvre ce fichier avec un éditeur de texte et vérifie qu'il contient bien le nom que tu as saisi.

Teste le programme **fichier2.py** suivant :

Pour cette activité, récupérer le document cigale.txt.

Utilise un éditeur de texte pour visualiser le contenu du fichier sans le modifier.

Ecris et teste maintenant le programme suivant :

```
monFichier=open('cigale.txt','r')
contenu=monFichier.read()
monFichier.close()
print(contenu)
```

Quel est le résultat obtenu après l'exécution de ce programme ?

Trouver les mots dans une chaîne

Dans un texte ordinaire, les mots se **distinguent** les uns des autres par la présence **d'espaces**, de **symboles** de ponctuation, de **parenthèses** les séparant. Nous allons considérer qu'un mot dans une chaîne sera toute partie de cette chaîne qui ne contient pas de séparateurs. La définition des mots dépend donc de celle de séparateur.

En informatique, les séparateurs sont :

- le marqueur de fin de ligne (`\n`) et la tabulation (`\t`)
- l'espace

- les symboles de ponctuation (.?!,:;)
- les tirets (-_) [1]
- les quote et double-quote (")
- les parenthèses ([] { })
- les symboles d'opérations arithmétiques (+-*/)
- les symboles de comparaison (=<>).

La suite de caractères `\n` entraîne un « **retour chariot** » comme quand on utilise la touche **Entrée** du clavier).

Exemple :

L'instruction suivante :

```
print('hello \n world')
```

n'affiche pas : hello \n world

mais :

hello

world

Pour pouvoir découper une chaîne de caractères tu peux utiliser la méthode **split()**. Mais cette méthode n'accepte qu'un seul séparateur. Il faut donc :

- **remplacer** tous les séparateurs que contient le texte par un seul d'entre eux qui sera le caractère espace, ou bien supprimer le séparateur si cela ne pose pas de problème,
- **effectuer** ensuite le découpage selon cet unique séparateur espace avec `split`.

Exemple de suppression du caractère « , » :

```
monFichier=open('cigale.txt','r')
contenu=monFichier.read()
contenu = contenu.replace('.', '')
monFichier.close()
print(contenu)
```

Exercices à faire

Programme 1 :

Avec un éditeur de texte, repère dans le texte **cigale.txt** les séparateurs à supprimer.

Ecris un programme **analyse.py** qui enlève tous les séparateurs présents de texte **cigale.txt**. afin d'obtenir la chaîne de caractères **contenu** qui ne contient que les mots.

Programme 2 :

Modifie ton programme **analyse.py** pour :

- découper la chaîne **contenu** afin d'obtenir la liste des mots,
- et compter les mots présents avec la fonction **len()**.

Programme 3 :

Recherche sur Internet un extrait numérique d'un livre étudié en cours et enregistre cet extrait dans un document au format txt.

Crée un programme **analysetexte.py** qui :

- demande le nom du document enregistré,
- puis qui affiche le nombre de mots.

Programme 4 :

Modifie ton programme **analysetexte.py** pour compter également le nombre d'occurrence de chaque mot présent.