

Les Fichiers

Fichiers texte

L'utilisation de fichiers permet de :

- ▶ **Séparer** les **données** des **programmes** qui les traitent
- ▶ Traitement d'une **importante quantité d'informations**
- ▶ **Rémanence** des informations d'une exécution à l'autre

Fichiers texte

Les fichiers que l'on peut traiter en Python sont des fichiers séquentiels :

- Les **fichiers textes** : Compréhensibles par l'utilisateur via un éditeur de texte (e.g. Spyder, bloc note, Notepad, etc.), manipulable par un programme. Ils sont constitués uniquement de caractères imprimables, d'espaces et de retours à la ligne (`\n`)
- Les **fichiers binaires** : Pas compréhensibles directement par l'utilisateur, manipulable uniquement par un programme

Fichiers texte

Les fichiers sont stockés sur l'ordinateur sous la forme :

```
nomfile = "nom_du_fichier.suffixe"
```

Les fichiers peuvent être accessibles via un **chemin d'accès** (ou path) :

e.g. : `path = "O:/Documents/Enseignements/Python/CoursS6/"`

Le nom complet du fichier est alors : `path + nomfile`

Pour plus de simplicité, il est conseillé (pour les TD) de stocker les fichiers **dans le même dossier** que le programme qui les utilise : il n'est alors pas nécessaire d'indiquer le chemin d'accès.

Le suffixe ou l'extension indique en général l'utilisation du fichier, e.g. :

.txt = Fichier texte générique

.py = Script Python

.dat ou .dta = Fichier de données (data)

.in ou .out = Données d'entrée ou de sorties

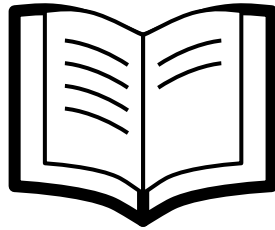
.cmd = Commandes

Manipulation de fichiers textes

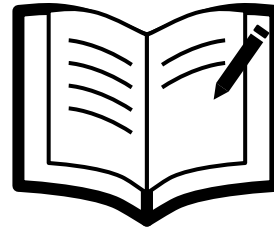
4 manipulations sont possibles :



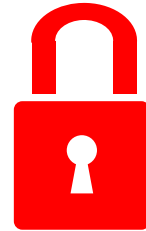
ouverture



lecture



écriture



fermeture

Opérations d'entrées-sorties

Ouverture de fichiers textes



```
objFichier = open(nom_fichier, mode_ouverture)
```

Objet fichier. C'est sur cet objet que ce feront les autres actions (lecture, écriture, fermeture)

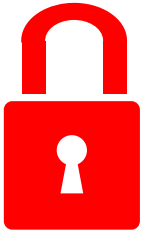
Nom du fichier, il s'agit d'une chaîne de caractères. Attention à ne pas oublier l'extension.

Mode d'ouverture

Les modes d'ouverture :

"r"	Mode lecture (read)	Le fichier doit déjà exister.
"w"	Mode écriture (write)	Si le fichier existe, il sera écrasé (i.e. vidé de son contenu), sinon il sera créé.
"a"	Mode ajout (append)	Si le fichier existe, on pourra le compléter (i.e. écrire à la fin du fichier), sinon il sera créé.

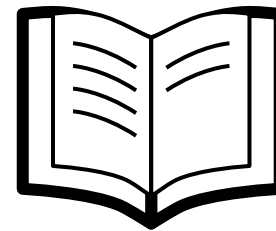
Fermeture de fichiers textes



```
objFichier.close()
```

Il faut systématiquement fermer les fichiers qui ont été ouverts !

Lecture dans un fichier texte



Pour pouvoir lire un fichier, il faut qu'il soit ouvert en mode lecture ("r").

Les fichiers en Python sont lus de manière séquentielle :

- On ne peut pas commencer à une ligne quelconque

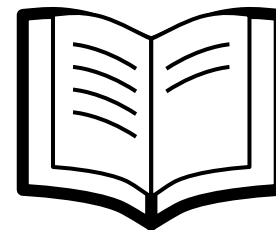
- On ne peut pas sauter de ligne

- On ne peut pas revenir en arrière

En général, les fichiers texte sont lus ligne par ligne.

Une **ligne** = 1 **chaîne de caractères** se **terminant** par un passage à la ligne "**\n**"

Lecture dans un fichier texte



Lecture de l'ensemble des lignes à partir de « la position du curseur » et jusqu'à la fin du fichier :

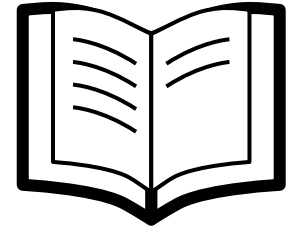
```
for ligne in objFichier :  
    ...
```

Lecture de la ligne suivante :

```
ligne = objFichier.readline()
```

=> La « position du curseur » est donc déplacée.

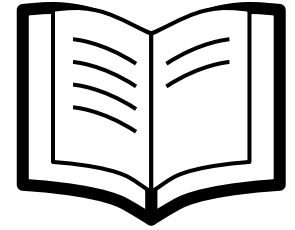
Lecture - Exemple



Écrire un programme qui lit et affiche à l'écran le contenu du fichier "fable.txt".

Qu'observez-vous ?

Lecture - Exemple



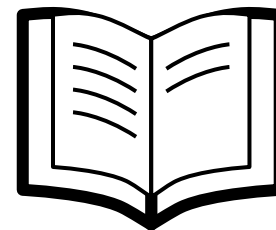
```
filename = "fable.txt"

fdonnees = open(filename, "r")

for ligne in fdonnees :
    print(ligne)

fdonnees.close( )
```

Lecture - Exemple



Fichier "fable.txt"



L'AVANTAGE DE LA SCIENCE
Jean de La Fontaine
Entre deux Bourgeois d'une Ville
S'émut (l)jadis un différend.
L'un était pauvre, mais habile,
L'autre riche, mais ignorant.
Celui-ci sur son concurrent
Voulait emporter l'avantage :
Prétendait que tout homme sage
Était tenu de l'honorer.
C'était tout homme sot ; car pourquoi révéler
Des biens dépourvus de mérite ?
La raison m'en semble petite.

Mauvaise gestion des accents
=> problème d'encodage

Apparition d'un saut de ligne

Affichage à l'écran

L'AVANTAGE DE LA SCIENCE
Jean de La Fontaine
Entre deux Bourgeois d'une Ville
S'Ã©mut (l)jadis un diffÃ©rend.
L'un Ã©tait pauvre, mais habile,
L'autre riche, mais ignorant.
Celui-ci sur son concurrent
Voulait emporter l'avantage :
PrÃ©tendait que tout homme sage
Était tenu de l'honorer.
C'Ã©tait tout homme sot ; car pourquoi rÃ©vÃ©rer
Des biens dÃ©pourvus de mÃ©rite ?
La raison m'en semble petite.

Ouverture de fichiers textes



Mauvaise gestion des accents => problème d'encodage

```
objFichier = open(nom_fichier, mode, encoding = "utf-8")
```

Permet de gérer les problèmes éventuels d'accents.

Lecture dans un fichier texte

Apparition d'un saut de ligne

- Lors de la lecture : une ligne se termine toujours par un **passage à la ligne**
- Lors de l'affichage : par défaut, print rajoute un **passage à la ligne**

Comment enlever le passage à la ligne à la fin d'une ligne à sa lecture :

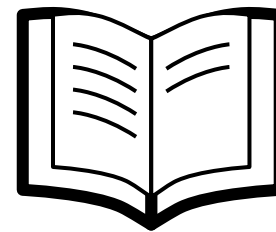
```
ligne = ligne.rstrip()
```

Comment enlever le passage à la ligne ajouté lors de l'affichage avec print :

```
print(ligne, end = "")
```

On peut choisir aussi de rajouter en fin de print d'autres caractères (e.g. alinéa : end = "\t" ; espace : end = " " ; tiret : end = "-", etc.)

Lecture - Exemple



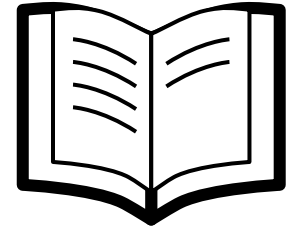
```
filename = "fable.txt"

fdonnees = open(filename, "r", encoding = "utf-8")

for ligne in fdonnees :
    print(ligne.rstrip())

fdonnees.close( )
```

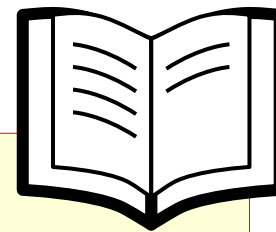
Lecture - Exemple



Écrire un programme qui, à partir du fichier "fable.txt" :

- Lit séparément le titre de la fable et le nom de l'auteur
- Les affiche en majuscule
- Lit puis affiche le contenu de la fable.

Lecture - Exemple



```
filename = "fable.txt"

fdonnees = open(filename, "r", encoding = "utf-8")

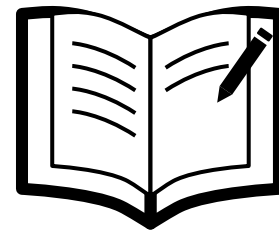
titre = fdonnees.readline().rstrip()
print(titre)

auteur = fdonnees.readline().rstrip()
print(auteur.upper())

for ligne in fdonnees :
    print(ligne.rstrip())

fdonnees.close( )
```

Écriture dans un fichier texte



On ne peut écrire que des **chaînes de caractères**, et une **unique** chaîne à la fois.

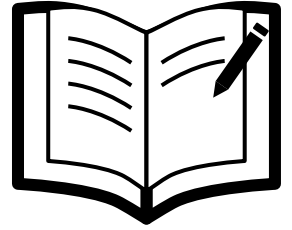
Pour pouvoir lire un fichier, il faut qu'il soit ouvert en mode écriture ("**w**") ou ajout ("**a**").

Chaque nouvelle appel de write continue l'écriture à la suite de ce qui est déjà enregistré.

Pour démarrer une nouvelle ligne, il est nécessaire d'ajouter un passage à la ligne "**\n**"

```
objFichier.write(chaine)
```

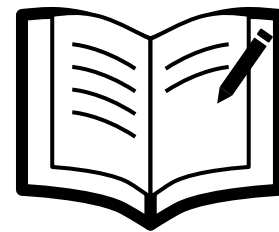
Écriture - Exemple



Écrire un programme qui, à partir du fichier "fable.txt" :

- Lit le fichier
- Écrit la fable, en majuscule, dans un nouveau fichier "fable_maj.txt"

Écriture - Exemple



```
filename = "fable.txt"
```

```
fdonnees = open(filename, "r", encoding = "utf-8")
```

```
fsortie = open("fable_maj.txt", "w", encoding = "utf-8")
```

```
for ligne in fdonnees :  
    fsortie.write(ligne.upper())
```

```
fdonnees.close()
```

```
fsortie.close()
```

Remarque : Ici ligne se termine par un passage à la ligne, il n'est donc pas nécessaire ni de l'enlever, ni d'en rajouter un dans le write.

Rappels – Chaînes de caractères

- Les lignes lues sont des **chaînes de caractères**.
- Les éléments que l'on peut écrire avec write sont des **chaînes de caractères**.

On doit donc utiliser l'ensemble des outils et fonctions vues sur les chaînes de caractères (**cf S5 - quizz**) pour pouvoir les manipuler :

Transformation nombre vers chaîne de caractères : **str(...)**

Transformation chaîne de caractère vers nombre : **int(...)**, **float(...)**

Évaluation d'une chaîne de caractère comme du code Python : **eval(...)**

Concaténation de chaînes de caractères : **+**

Séparation d'une chaîne de caractère grâce à un séparateur : **split(...)**

etc.