

Python permet d'interagir avec des fichiers (écriture, lecture). On s'intéresse aujourd'hui aux fichiers contenant du texte. Le processus se calque sur l'utilisation d'un livre : on le trouve (à l'aide de son titre), on l'ouvre, on l'utilise (soit pour le lire, soit pour l'écrire), puis on le ferme!

I Le Poète

1. Essayer :

```
monpoeme=open("LePontMirabeau","w")
monpoeme.write("Sous le pont Mirabeau coule la Seine")
monpoeme.close()
```

- a. Quelle est l'action de la commande `monpoeme=open("LePontMirabeau","w")` ?
- b. Que signifie l'option `"w"` dans l'instruction `open` ?
- c. Quelle est la différence entre `"LePontMirabeau"` et `"monpoeme"` ?

2. Vérifier dans l'arborescence de Lubuntu que votre fichier `LePontMirabeau` a bien été créé et qu'il contient bien le premier vers.

3. L'inspiration venant, essayons d'ajouter le deuxième vers :

```
monpoeme=open("LePontMirabeau","w")
monpoeme.write("Et nos amours")
monpoeme.close()
```

4. Que contient à présent le fichier `LePontMirabeau` ?
 - a. Que signifie réellement l'option `"w"` dans l'instruction `open` ?
 - b. Essayer l'option `"a"`.

5. Un saut de ligne peut-être inséré en utilisant le caractère `\n`. Essayer :

```
monpoeme=open("LePontMirabeau","w")
monpoeme.write("Sous le pont Mirabeau coule la Seine\n")
monpoeme.write("Et nos amours\n")
```

Examiner le contenu du fichier `"LePontMirabeau"` à ce moment de l'écriture. Que constate-t-on ?

```
monpoeme.close()
```

Que contient alors le fichier `"LePontMirabeau"` ? Que s'est-il passé ?

6. Ajouter le troisième vers : *Faut-il qu'il m'en souviennne*

II Le lecteur

1 Lecture intégrale

Essayer :

```
poeme=open("LePontMirabeau","r")
t=poeme.read()
print(t)
poeme.close()
```

1. Vérifier le contenu de `t`.
2. Quelle est le type de `t` ?

2 Lecture séquentielle

1. Tâcher de prévoir le comportement du programme :

```
poeme=open("LePontMirabeau","r")
t=poeme.read(6)
print(t)
t=poeme.read(35)
print(t)
poeme.close()
```
2. L'exécuter, et confirmer ou infirmer votre conjecture.
3. Que signifie l'expression *lecture séquentielle* ?
4. Que renvoie la commande `read()` lorsque la fin du fichier est atteinte ?

3 Lecture séquentielle par lignes

1. Essayez :

```
f=open("Lignes","w")
f.write("Ceci est la ligne 1\nVoici la ligne 2\n")
f.write("Ici, la ligne 2\nEt enfin la 4!")
f.close()

f=open("Lignes","r")
t=f.readline()
print(t)
t=f.readline()
print(t)
t=f.readline(10)
print(t)
t=f.readlines()
print(t)
f.close()
```
2. Décrire le fonctionnement des commandes `f.readline()`, `f.readline(n)` et `f.readlines()`.
3. Que renvoient les commandes `f.readline()` et `f.readlines()` lorsque la fin de fichier est atteinte ?

III Je fais le point

Le fichier sera appelé "MonFichier", et le fichier-objet (sa représentation en Python) sera appelé `f`.

1. Commande pour ouvrir le fichier "MonFichier" (préciser les options) :
2. Commande pour fermer le fichier :
3. Commande pour écrire une ligne dans le fichier :
4. Commandes pour lire dans le fichier et leur effet :
 -
 -
 -
5. Le caractère de saut de ligne est :
6. `f.readline(3)` lit la troisième ligne du fichier ? Vrai ou Faux ?
7. L'auteur du *Pont Mirabeau* est :

IV L'informaticien

1. Écrire une fonction `def gestion(fich)` : prenant comme argument un nom de fichier, et proposant à l'utilisateur deux choix :
 - afficher le contenu du fichier ;
 - ajouter des lignes au fichier.Dans ce dernier cas, on demandera à l'utilisateur d'entrer ses lignes de texte successives en utilisant simplement la touche *Enter* pour les séparer les unes des autres. Pour terminer les entrées, il lui suffira d'entrer une ligne vide (c'est-à-dire d'utiliser la touche *Enter* seule).
Ne pas oublier d'insérer les sauts de ligne...
2. Écrire un script cherchant et affichant la ligne la plus longue d'un fichier texte donné.
3. Écrire une fonction `def filtre(source,destination)` : recopiant le fichier source dans le fichier destination, après avoir supprimé toutes les lignes débutant par un `#`.
4. Écrire un script qui recopie un fichier texte en triplant tous les espaces entre les mots.
5. Écrire un script déterminant le nombre de voyelles dans un texte donné.
6. Écrire un script qui compare les contenus de deux fichiers et signale la première différence rencontrée.
7. Écrire un script engendrant automatiquement un fichier texte contenant les tables de multiplication de 2 à 12 (chacune d'elle contenant 20 termes).
8. A partir de deux fichiers préexistants A et B, construire un fichier C qui contienne alternativement un élément de A, un élément de B, un élément de A, et ainsi de suite, jusqu'à atteindre la fin de l'un des deux fichiers originaux. Compléter ensuite C avec les éléments restant sur l'autre.

Remarque : le codage des caractères est généralement fait via la norme Utf-8 (Universal character set Transformation Format -8 bits). Ce format ne prend pas en compte les caractères accentués des langues latines. Si des problèmes se posent suite à l'utilisation des accents, on peut ajouter l'option d'encodage Latin-1 :

```
f=open("monFichier","w",encoding="Latin-1")
```

On précisera alors l'option pour la lecture également :

```
f=open("monFichier","r",encoding="Latin-1")
```

Le Pont Mirabeau

Sous le pont Mirabeau coule la Seine
Et nos amours
Faut-il qu'il m'en souvienn
La joie venait toujours après la peine
Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

Les mains dans les mains restons face à face
Tandis que sous
Le pont de nos bras passe
Des éternels regards l'onde si lasse
Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

L'amour s'en va comme cette eau courante
L'amour s'en va
Comme la vie est lente
Et comme l'Espérance est violente
Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

Passent les jours et passent les semaines
Ni temps passé
Ni les amours reviennent
Sous le pont Mirabeau coule la Seine
Vienne la nuit sonne l'heure
Les jours s'en vont je demeure