

1 Accès à un fichier

1.1 Création d'un objet-fichier

En python, création et manipulation d'un fichier se font par l'intermédiaire d'un objet-fichier, généré par la fonction :

```
objet_fichier = open(nom du fichier, mode d'accès).
```

Les paramètres sont des chaînes de caractère :

– **nom du fichier** est le nom du fichier, avec son extension.

– **mode d'accès** peut être :

'w' : (write) ouverture pour écriture seule. Lorsque le fichier n'existe pas il est créé dans le répertoire courant; lorsque le fichier existe il est écrasé.

'a' : (append) ouverture pour écriture seule. Lorsque le fichier n'existe pas il est créé dans le répertoire courant; lorsque le fichier existe les données écrites le seront à la suite.

'r' : (read) ouverture pour lecture seule. Le fichier doit exister dans le répertoire courant.

1.2 Où se trouve le fichier ?

Pour créer un objet-fichier :

– Le fichier doit être présent dans le dossier courant (*current working directory* ou *répertoire de travail*) : en général c'est le dossier utilisateur (celui de même nom que le compte utilisateur).

– Sinon on peut faire précéder le nom du fichier par le **chemin d'accès** :

```
obj_fich = open('/Users/Moi/Documents/Fichier.ext', 'r')
```

1.3 Obtention et changement du répertoire courant

Pour obtenir le répertoire de travail, utiliser la fonction `getcwd()` du module `os` :

```
>>> import os
>>> os.getcwd()
'/Users/Moi'
```

• Pour modifier le répertoire de travail, utiliser la fonction `chdir` du module `os` :

```
>>> os.chdir('/Users/Moi/Documents')
>>> os.getcwd()
'/Users/Moi/Documents'
```

1.4 Fermeture

Une fois la lecture et l'écriture dans le fichier terminés il faut refermer l'objet-fichier avec l'instruction :

```
objet_fichier.close()
```

- Les modifications apportées au fichier ne seront prises en compte qu'après la fermeture.
- Tant qu'un fichier n'a pas été fermé, son ouverture provoquera une erreur.



Ne jamais oublier de finir par refermer un objet-fichier par `objet_fichier.close()`

1.5 Méthodes des objets-fichiers

Un objet-fichier admet les méthodes :

- Lorsque **ouvert en écriture** :
`write()` écrit une chaîne de caractère en fin de fichier ouvert en écriture.
- lorsque **ouvert en lecture** :
`read()` lit l'intégralité du fichier. `read(n)` lit `n` caractères.

1.6 Exemples

On suppose que le fichier texte suivant se trouve dans le répertoire courant :



```
>>> objet = open('monfichier.txt', 'r')
>>> lect = objet.read()
>>> print(lect)
Premiere ligne
Deuxieme ligne...Suite deuxieme ligne
Troisieme ligne
>>> objet.close()
```

On ajoute du texte en fin de fichier grâce au code suivant :

```
>>> objet = open('monfichier.txt', 'a')
>>> objet.write("Quatrieme ligne rajoutee")
>>> objet.close()
```



```

monfichier.txt
Premiere ligne
Deuxieme ligne...Suite deuxieme ligne
Troisieme ligne
Quatrieme ligne rajoutee|

```

2 Exercices

Exercice 1. Création de fichier.

1. Écrire et exécuter le code suivant :

```

nom = input("Saisir le nom du fichier : ")
texte = input("Saisir le texte du fichier : ")
nom = nom + ".txt"
objet_fichier = open(nom, 'w')
objet_fichier.write(texte)
objet_fichier.close()

```

Trouver votre fichier, et l'ouvrir en double-cliquant avec la souris. Modifier son contenu dans le bloc-note avant de le sauvegarder.

2. Écrire le code permettant de récupérer dans une variable `nouveau_texte` de type chaîne de caractère tout le texte contenu dans le fichier.
3. Écrire le code permettant de créer une copie de votre fichier texte sur votre "bureau".

Exercice 2. Cryptage de César

1. Le cryptage de César est une méthode de cryptographie utilisée par Jules César durant la guerre des Gaules (1er siècle av. J.C.). Elle consiste à dissimuler le contenu d'un message en décalant chaque lettre de l'alphabet d'un rang fixé (ce rang est la clé). En fin d'alphabet on reprend au début de l'alphabet.

Par exemple si A est changé en C, alors B est changé en D, C est changé en E, etc. et Z est changé en B.

(D'après l'oral de l'ENSAM)

Soit n un entier vérifiant $n \leq 26$. On souhaite écrire un programme qui code un mot en décalant chaque lettre de l'alphabet de n lettres. Par exemple pour $n = 3$, le décalage sera le suivant :

avant décalage	a	b	c	d	x	y	z
après décalage	d	e	f	g	a	b	c

Le mot `oralensam` devient ainsi `rudohqvdp`.

- (a) Définir une chaîne de caractères contenant toutes les lettres dans l'ordre alphabétique (caractères en minuscule).
- (b) Écrire une fonction `decalage`, d'argument un entier n , renvoyant une chaîne de caractères contenant toutes les lettres dans l'ordre alphabétique, décalées de n , comme indiqué ci-dessus.
- (c) Écrire une fonction `indices`, d'arguments un caractère x et une chaîne de caractères `phrase`, renvoyant une liste contenant les indices de x dans `phrase` si x est une lettre de phrase et une liste vide sinon.
- (d) Écrire une fonction `codage` d'arguments un entier n et une chaîne de caractères `phrase`, renvoyant `phrase` codé avec un décalage de n lettres.
- (e) Comment peut-on décoder un mot codé ?

1. Nous allons maintenant créer un fichier crypté.

- (a) Créer dans votre répertoire courant, à l'aide du logiciel bloc-note, un fichier texte contenant le texte de votre choix.
- (b) Appliquer la fonction `codage()` de l'exercice TP10-3) pour écrire un script qui crée un nouveau fichier contenant votre fichier texte crypté par le cryptage de César.
- (c) Écrire le code permettant de créer un nouveau fichier texte contenant le texte après décryptage.