

Les fichiers

Pour manipuler les fichiers en Python Il y a un type spécial et on crée un objet de ce type de la façon suivante :

```
#!/usr/bin/python
f = open(<ref> [, mode=<opt>, encoding=<encodage>])
```

Les différents modes possibles:

- "r" ouverture en mode lecture. C'est le mode par défaut (si on indique le mode d'ouverture).
- "w" ouverture en mode écriture.
- "a" ouverture en mode écriture à la fin du fichier.

L'encodage peut-être modifier (par défaut celui du shell dans lequel est lancé l'interpréteur python)

- ex : encoding='utf-8'

```
f = open('mon_fichier')
```

Si le fichier 'mon_fichier' n'existe pas ou si l'ouverture ne peut se faire open lève une exception de type IOError.

```
try:
    f = open('mon_fichier')
except IOError:
    printf("Erreur ouverture fichier")
```

Mais Pour l'ouverture en écriture si le fichier n'existe pas, il est créé

```
f = open('mon_fichier', mode='w')
```

Si le fichier est ouvert en mode 'r' ou 'w' l'objet de type file f pointe alors au début du fichier. Pour l'ouverture avec le mode 'a' f pointe alors à la fin du fichier

```
f = open("fichier.txt")
```

Lecture

Pour lire, il y a les méthodes :

- `read()` : `f.read(n)` # on lit les `n` char(octets) à partir de la position du curseur et le place ensuite `n` char(octet) plus loin
- `readline()` : `f.readline()` # on lit une ligne de la position du curseur jusqu'à la fin de la ligne (`\n`)
- `readlines()` `for ligne in readlines():` # la fin de la boucle le curseur est placé à la fin du fichier

```
print(ligne)
```

Écriture

```
f = open("fichier.txt", "w")
```

```
f.write("blabla")    # écrit la chaîne après la position du curseur et la modifie  
f.write(str())
```

Curseur

seek:

La méthode `seek()` permet de changer la position du curseur

```
seek(<position en nombre d'octet>, constante=0)
```

SEEK_SET or 0: seek from the start of the stream (the default); offset must either be a number returned by `TextIOWBase.tell()`, or zero. Any other offset value produces undefined behaviour.

SEEK_CUR or 1: “seek” to the current position; offset must be zero, which is a no-operation (all other values are unsupported).

SEEK_END or 2: seek to the end of the stream; offset must be zero (all other values are unsupported).

```
Return the new absolute position as an opaque number.
```

tell:

```
La méthode tell() permet de connaître la position du curseur
```

ATTENTION

Ne pas oublier de fermer le fichier

```
f.close()
```