

PYTHON

Présenté par :
Xavier TABUTEAU

Les fichiers textes

Les fichiers textes

Un fichier texte est un fichier contenant des caractères lisibles (comme .txt, .csv, .html, etc.). Python permet de lire, écrire, ajouter ou modifier ces fichiers facilement.

Ecriture

```
>>> f = open('Monfichier', 'w')
>>> f.write('Bonjour, fichier !')
>>> f.write("Quel beau temps!")
>>> f.close()
```

Lecture

```
>>> f = open('Monfichier', 'r')
>>> t = f.read()
>>> print(t)
Bonjour, fichier !Quel beau temps !
>>> f.close()
```

- Liste des modes d'ouverture des fichiers textes.

'r'	Read (lecture)	Ouvre un fichier en lecture seule (le fichier doit exister).
'w'	Write (écriture)	Ecriture seule ; crée ou écrase le fichier s'il existe.
'x'	Exclusive	Créer le fichier. provoque une erreur s'il existe déjà.
'a'	Append (ajout)	Ajoute à la fin ; crée le fichier s'il existe.
'r+'	Read / Write	Lecture / écriture ; ne crée pas ; n'efface rien ; curseur au début.
'w+'	Read / Write (écrase)	Lecture / écriture ; crée ou écrase le fichier.
'a+'	Read / Append	Lecture / ajout à la fin ; curseur au début pour lecture, fin pour écriture.
't'	Text	Mode texte. Valeur par défaut de open().
'b'	Binary	Mode binaire. Pour lire/écrire d'autres choses que du texte.

Les modes «t » et « b » se combine aux autres modes. Exemple : « rt », « rt+ », etc...

Les fichiers textes

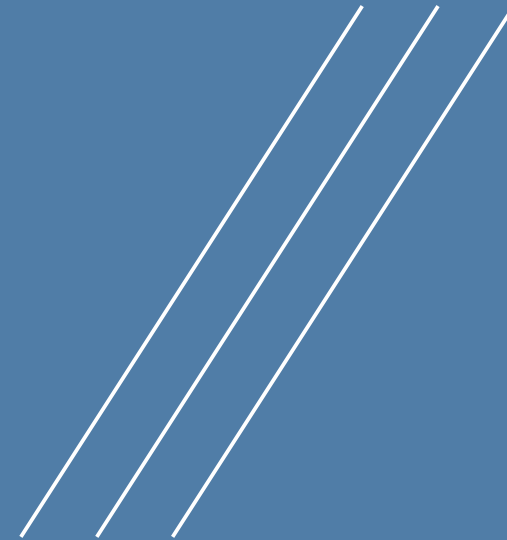
Les fichiers textes

- Les fonctions pour manipuler les fichiers

<code>open()</code>	: Ouvre le fichier. Attend deux arguments minimums, qui doivent tous deux être des chaînes de caractères. Le premier argument est le nom du fichier, et le second est le mode d'ouverture.
<code>write()</code>	: Réalise l'écriture proprement dite. Les données à écrire doivent être fournies en argument. Chaque nouvel appel de <code>write()</code> (en mode <code>a</code>) continue l'écriture à la suite de ce qui est déjà enregistré.
<code>writelines()</code>	: Accepte un itérable (liste, tuple, etc...) contenant des strings. La fonction écrira tous les éléments de l'itérable les uns à la suite de autres. Il ne faut pas oublier d'insérer les « <code>\n</code> » à la fin des strings si on veut des sauts de lignes à ce moment là.
<code>read()</code>	: Permet de lire le contenu d'un fichier dans son ensemble.
<code>readline()</code>	: Lit qu'une ligne par appel de cette instruction.
<code>readlines()</code>	: Lit les lignes individuellement dans une liste.
<code>seek(offset, whence)</code>	: Déplace le curseur dans le fichier (<code>whence=0</code> : début, <code>1</code> : position actuelle, <code>2</code> : fin). Si <code>whence</code> n'est pas spécifié le curseur est déplacé de la valeur de l'offset à partir de sa position actuelle.
<code>close()</code>	: Referme le fichier dans n'importe qu'elle mode.

`fichier_texte.py`
`ex_fichier`
`ex_multi_fichiers`
`ex_fichier_mini_bdd`

PYTHON



Présenté par
Xavier TABUTEAU