



TRAITEMENT DES DONNÉES EN TABLES (1)

[Frédéric PEURIERE]

Savoir lire et traiter les données d'un fichier texte

Mises en pratique de l'ensemble de nos connaissances : création d'un jeu.

LECTURE DE FICHIERS TEXTE:

Une manière de stocker des données de manière pérenne est de les stocker dans des fichiers.

Pour **éditer un fichier** en **python** on utilise la fonction `open`.

Cette fonction prend en premier paramètre le chemin du fichier (relatif ou absolu) et en second paramètre le type d'ouverture.

Il existe plusieurs modes d'ouverture de fichier:

r, pour une ouverture en lecture (READ). **w**, pour une ouverture en écriture (WRITE), à chaque ouverture le contenu du fichier est écrasé. Si le fichier n'existe pas python le crée. **a**, pour une ouverture en mode ajout à la fin du fichier (APPEND). Si le fichier n'existe pas python le crée.

✓ **ACTIVITES:** Le fichier *dico.txt* contient plus de 800 mots de la langue française. Téléchargez le fichier *dico.txt* sur le site puis créez un fichier *fichier-texte.py* dans le même répertoire et testez les codes suivants:

Deux méthode d'accès au fichier:

```
fichier = open("dico.txt", "r")
line=fichier.read()
print (line)
fichier.close()
```

On lit l'intégralité du fichier. On peut aussi utiliser la syntaxe suivante qui permet de ne pas avoir à fermer le fichier:

```
with open("dico.txt", "r") as fichier:
    line=fichier.read()
    print (line)
```

La fonction write:

```
with open("dico.txt", "a") as fichier:
    line=fichier.write("\n une ligne à la fin")
# Une ligne est ajoutée dans le fichier, vérifiez-le
```

La fonction readline:

```
with open("dico.txt", "r") as fichier:
    line=fichier.readline()
    print (line)
# Lit la première ligne du fichier.
```

```
with open('dico.txt','r') as fichier :
    for n in range(10):
        line=fichier.readline()
        print(line)
# Parcourt les dix premières lignes du fichier
```

```

with open('dico.txt','r') as fichier:
    line = fichier.readline()
    cpt=0
    while line:
        line=fichier.readline()
        cpt+=1
print (cpt)
# Parcourt et compte les lignes.

```

✓ EXERCICE: LE JEU DU PENDU

Téléchargez maintenant le fichier *pendu.py* sur le site et placez-le dans le même répertoire.

- Ouvrez le fichier *pendu.py*. Faites quelques appels de la fonction `dessinPendu()` pour comprendre son fonctionnement...
- Créons maintenant un nouveau fichier **jeu-pendu.py** toujours dans le même répertoire et importons-y la fonction `dessinPendu()`. Vous êtes prêts à créer votre jeu du pendu !

OBJECTIF 1 : Initialisation du jeu

Créer une fonction `init()` qui choisit un mot au hasard dans le fichier texte dont le nom de variable est `mot`.

`mot=fichier.readline().strip()` (la méthode `strip()` supprime le caractère `'\n'` correspondant au saut de ligne dans le fichier.

OBJECTIF 2 : Savoir si la lettre est présente dans la variable `mot`.

Il est conseillé d'écrire le code du jeu dans une fonction `jeu()`.

- Afficher le premier pendu.
- Créer une liste (variable `liste`) ne contenant que les caractères `'-'`. Cette liste a bien sûr le même nombre d'éléments que le mot à deviner (`mot`). Afficher cette liste.
- Demander au joueur de saisir une lettre.
`lettre=input("Entrez une lettre: ").upper()`
La méthode `upper()` convertit le caractère saisi en majuscule.
- Si la lettre saisie n'est pas dans le mot, informer l'utilisateur et affichez le pendu suivant et la liste du mot à deviner. Au dernier pendu, le joueur aura perdu...
- Si elle y est, informer le joueur et afficher la liste mise à jour avec la lettre trouvée placée au bon endroit.

```

for i,j in enumerate(mot):
    if j == lettre:
        liste[i]=j

```

Le code au dessus parcourt la chaîne de caractère et place dans la liste le caractère trouvé à la bonne place (i est le caractère et j son index).

OBJECTIF 3 : Finir votre jeu !