

Fichiers

Polytech Marseille, GII, 3A

Séverine Dubuisson, Simon Vilmin
severine.dubuisson@univ-amu.fr,
simon.vilmin@univ-amu.fr

2023 - 2024



Plan

Fichiers

- Généralités

- Lecture

- Écriture

- Résumé


Fichiers


- Généralités

- Lecture

- Écriture

- Résumé

 **Remarque** : Jusqu'à présent on a utilisé `print()`, `input()` ou saisi directement nos données dans les programmes.

 **Idée** : On peut utiliser *des fichiers* pour *stocker* nos données, y *accéder* et les *partager* (et bien plus encore) !

Syntaxe générale

 **Syntaxe** : on utilise le bloc `with open(...)` `as` pour ouvrir et traiter un fichier

```
with open(name, mode) as file:  
    # traitement
```

- `open` retourne un *descripteur de fichier* stocké dans la variable `file`
- `name` et `mode` sont des `str`
- `name` est le nom du fichier
- `mode` est le mode d'ouverture du fichier : lecture ("`r`"), écriture ("`w`", "`a`")
- à la fin du `with`, le fichier est *fermé*

Attention :

- ne pas oublier l'*extension* du fichier dans `name`
- si le fichier n'est *pas dans le répertoire courant*, il faut aussi préciser le *chemin d'accès* au fichier !

Exemple

```
# ==== Prog principal
with open("C:\Users\Simon\Documents\pouet.txt", "w") as file:
    file.write("1 2 3, nous irons au bois \n") # ecriture
    file.write("4 5 6, cueillir des saucisses \n")
    for i in range(5):
        file.write(f"{i * 3} ")

with open("C:\Users\Simon\Documents\pouet.txt", "r") as file:
    for line in file: # lecture
        print(line)

# ==== Resultat
1 2 3, nous irons au bois
4 5 6, cueillir des saucisses
0 3 6 9 12
```

Modes d'ouvertures

- "r" (read) : *lecture seule* (read)
- "w" ((re)write) : *écriture seule* et *écrase* le contenu si le fichier est non vide
- "a" (append) : *écriture seule* et *ajoute* ce que l'on écrit à la fin du fichier

i Remarque :

- "w" et "a" *créent un nouveau fichier* si celui précisé n'existe pas
- "r" renvoie une *erreur* si le *fichier n'existe pas*
- il existe d'autres modes ("w+", "a+", "b", ...) non vus ici

```
In []: file = open("inexistant.pouet", "r")
```

```
Out []: FileNotFoundError: [Errno 2] No such file  
or directory: 'inexistant.pouet'
```

Lecture d'un fichier

Une fois un fichier *ouvert en lecture*, plusieurs possibilités :

- `read()` lit ce qui *reste à lire* dans le fichier
- `read(i)` lit *i* caractères
- `readline()` lit *une ligne* du fichier
- `readlines()` renvoie la liste de toutes les lignes du fichier
- parcours du fichier *ligne par ligne*

```
for line in file:  
    print(line) #affichage d'une ligne
```

i Remarque : les fonctions `read()` et `readline()` renvoient des *chaînes de caractères*!

Exemple

```
def lecture_lignes(name):
    with open(name, "r") as file:
        print(file.readline())
        print(file.readline())


    print("==== STOP, changement de methode ====")

    for line in file:
        print(line, end="")

def lecture_unique(name):
    with open(name, "r") as file:
        print(file.read())

if __name__ == "__main__":
    lecture_lignes("fichiers.py")
    lecture_unique("fichiers.py")
```

À propos de `open`

 **Question** : mais `open()` ça renvoie quoi au juste ?

 **Réponse** : un *descripteur de fichier*

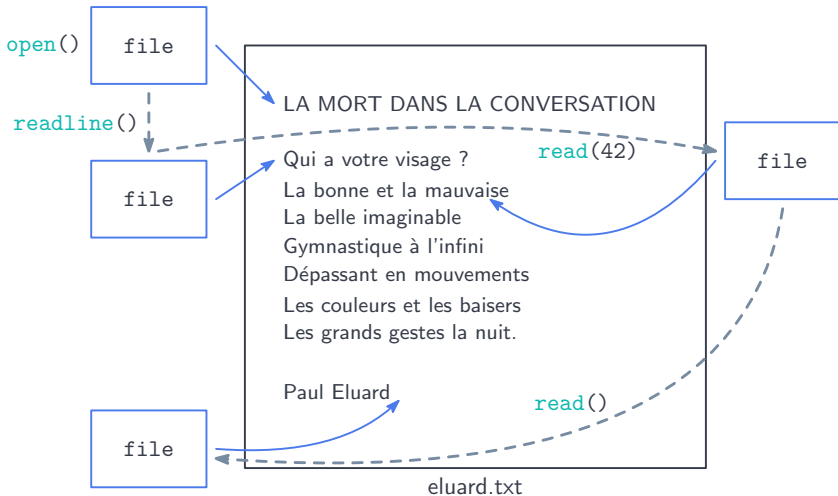
Dans les grandes lignes :

- descripteur \neq fichier !
- descripteur \simeq intermédiaire pour accéder au fichier

Fonctionnement :

- un fichier (texte) est une suite de *caractères*
- à l'ouverture le *descripteur* se positionne au *début du fichier*
- une opération de lecture (ou d'écriture) se *déplace* dans le fichier

Visuellement



Écriture


Une fois un fichier *ouvert en écriture* on écrit avec `write`

```
with open(fichier, "w") as file:  
    file.write("...")
```

! Attention : `write`

- prend en paramètre une *chaîne de caractères*
- et renvoie le *nombre d'octets écrits* (= le nombre de caractères écrits)

i Remarque : On peut écrire plein de types de fichier : du texte, du python (exécutable), des images, ...

 Rappel : il y a deux modes d'écritures, "w" et "a"

Exemple


```
def ecriture(name, mode, content):
    with open(name, mode) as file:
        for line in content:
            f.write(line + "\n")

if __name__ == "__main__":
    programme = [
        "a = 5",
        "b = 6",
        "print(a + b)",
        "print('AAAAAH')",
    ]
    ecriture("essai.py", "w", programme)
```

Résumé

Rappel : opérations sur les fichiers

- ouverture avec bloc `with`
- mode *lecture* ("`r`") ou *écriture* ("`w`", "`a`")
- manipulation avec `read()`, `readline()`, `write()`


 **Remarque** : on pourrait utiliser `open()` et `f.close()` au lieu du bloc `with` mais :

- il faut *fermer manuellement le fichier* avec `close`
- `with` préférable selon la doc !

```
with open("fichier.txt", "r") as file:
    for line in file:
        print(line)
```

Exercices

 **Exercice** : écrire une fonction `copie` qui prend en paramètres deux noms de fichiers `fic1`, `fic2` et qui recopie le contenu de `fic1` dans `fic2`.

 **Exercice** : écrire une fonction `commentaires` qui prend en paramètres un nom de fichier `python prog1` et qui écrit un fichier python `comments` qui contient le programme de `prog1` mais complètement commenté (un `#` au début de chaque ligne).