

[{m.  aly}]

# Python Niveau 2

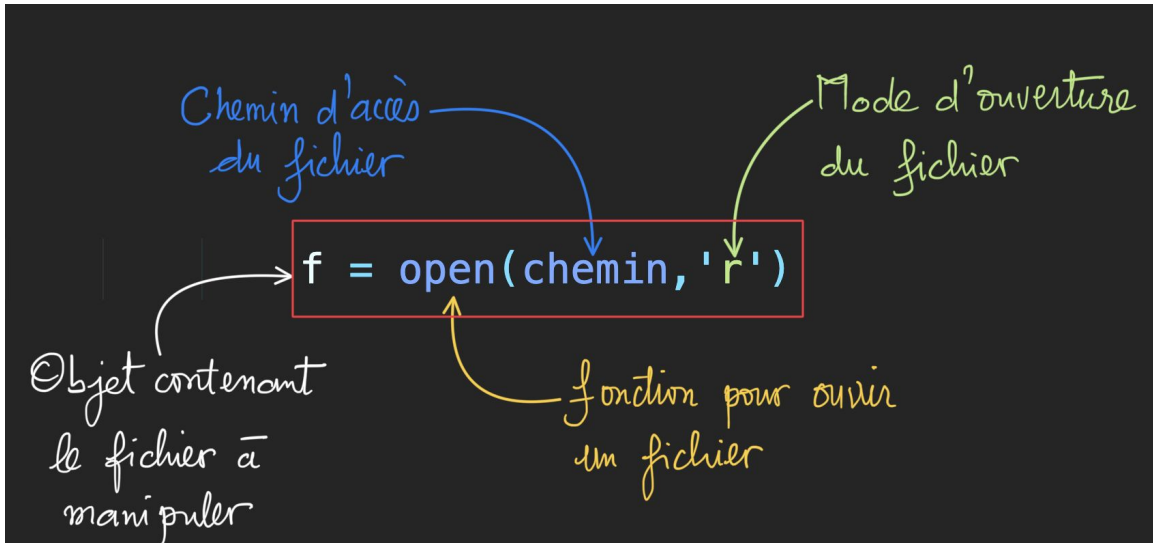
Aly Tall NIANG

# Les Fichiers

# Les fichiers

- Ouverture d'un fichier (1) :

Avant de manipuler un fichier, il faudra l'ouvrir.



# Les fichiers

- Ouverture d'un fichier (2) :

- **OPEN()** : Cette fonction renvoie un **objet de type** fichier permettant la liaison avec le fichier.
- **CHEMIN** : C'est une chaîne de caractères représentant le nom du fichier avec son extension (précède éventuellement du chemin - **absolu** ou **relatif**).
- **MODE** : C'est une chaîne de caractère qui indique le mode d'ouverture du fichier (les manipulations à faire avec le fichier).

Il existe trois principaux modes d'ouverture :

- **r** : Lecture (Read)
- **w** : Écriture (Write)
- **a** : Ajout (Append)

D'autres modes plus complexes existent aussi (voir tableau).

# Les fichiers

Mode d'ouverture	Description
r	Ouvre un fichier en lecture seule. Le pointeur est placé au début du fichier. C'est la valeur par défaut.
w	Ouvre un fichier en écriture seule. Écrase le contenu du fichier s'il existe déjà, crée un nouveau fichier sinon. Le pointeur est placé au début du fichier.
a	Ouvre un fichier en écriture seule. Le pointeur est placé à la fin du fichier s'il existe, sinon un nouveau fichier est créé.
rb , wb , ab	Ouvre le fichier en mode binaire (lecture des octets au lieu des caractères).
r+ , w+ , a+	Ouvre le fichier en lecture et en écriture.
rb+ , wb+ , ab+	Combinaison des deux précédents.

# Les fichiers

- Fermeture d'un fichier :

Après avoir fini avec un fichier il faudra le fermer. On peut fermer un fichier comme suit:

- ❖ **fichier.close()** : permet de fermer le fichier (couper la liaison de l'objet avec le fichier).
- ❖ l'instruction **WITH** : permet de fermer le fichier automatiquement après traitement.

```
f = open(chemin, 'r')
# Manipulations...
f.close();

# OU BIEN
with open(chemin, 'r') as f:
    # Manipulations ...
```

# Les fichiers

- Lecture d'un fichier texte (1) :

- **fichier.read()** : lire l'intégralité du fichier dans une chaîne de caractères.
- **fichier.readlines()** : lire l'intégralité du fichier dans une liste (séparateur = saut de ligne). son équivalent est : `line = f.read().splitlines()`
- **fichier.readline()** : lire la ligne suivante du fichier dans une chaîne de caractères.
- **NB** : Ces méthodes peuvent prendre en paramètre le nombre de caractères max que l'on veut lire. Exemple : `fichier.read(10)` => lire les 10 prochains caractères.

# Les fichiers

- Exemples de lectures de fichier texte :

**NB :**

sur window, ajouter le paramètre :

*encoding='utf-8'* à la fonction open()

```
with open(chemin,'r') as f:
    contenu = f.read()
    print(contenu)

# -----
with open(chemin,'r') as f:
    contenu = f.readlines()
    print(contenu)

# -----
with open(chemin,'r') as f:
    contenu = f.read(10)
    print(contenu)
```



# Les fichiers

- Lectures de fichier texte (2) :

- Avec la boucle FOR :

```
with open(chemin, 'r') as f:  
    for line in f:  
        print(line)
```

- Avec la boucle WHILE :

```
with open(chemin, 'r') as f:  
    line = f.readline()  
    while line:  
        print(line)  
        line = f.readline()
```

# Les fichiers

- Ecriture dans un fichier texte :
  - `fichier.WRITE(chaine)` :

Cette méthode écrit la chaîne dans un fichier puis retourne le nombre de caractères écrit.

**Attention** : si le fichier est ouvert en mode ÉCRITURE (w) alors son ancien contenu est supprimé.

```
1 chemin = './fichier.txt'
2 with open(chemin, 'w', encoding='utf-8') as f:
3     nbre = f.write("Je suis en train d'ecrire")
4     print("{} caracteres ecrits dans le fichier".format(nbre))
5
```

# Les fichiers

- Exercice 1 :

Soit une liste d'éléments, écrire un programme qui transfère le contenu de la liste dans un fichier txt dont le nom sera saisi par l'utilisateur.

```
1 liste_fruits = ["mangue", "pomme", "papaye", "banane", "coco"]
2
3 chemin = input("Entrer le nom du fichier a cree :\n")
4
5 with open(chemin, 'a', encoding='utf-8') as f:
6     for fruit in liste_fruits:
7         f.write(f"{fruit}\n")
8
```

← Remarquer le formatage

# Les fichiers

- Exercice 2 :

Soit le fichier créé précédemment, écrire un programme qui copie le contenu du fichier dans un autre fichier nommé copie-nomDuPremierFichier tout en mettant tout son contenu en majuscule.

```
3 chemin1 = "fruits.txt"
4 chemin2 = f"copie-{chemin1}"
5
6 with open(chemin1,'r') as f1, open(chemin2,'w') as f2:
7     contenu = f1.read()
8     f2.write(contenu.upper())
9
```

# Les fichiers

- Exercice 3 :

Soit le fichier créé précédemment, écrire un programme qui copie le contenu du fichier dans un autre fichier nommé copie-nomDuPremierFichier tout en mettant un numéro d'ordre devant chaque fruit.

```
3 chemin1 = "fruits.txt"
4 chemin2 = f"copie-{chemin1}"
5
6 with open(chemin1,'r') as f1, open(chemin2,'w') as f2:
7     i = 1
8     for fruit in f1:
9         f2.write(f"{i}- {fruit}")
10        i+=1
```