

Affichage print

Exemple :

```
print( i , " fois " , j , " = " , i*j)
```

Après chaque instruction `print`, il y a un passage à la ligne suivante

Lecture de fichier externe

`fichier = open(nom_fichier, mode)` ouvre le fichier, `mode` vaut "r", "w" ou "a".

"r" : Ouvre un fichier en lecture uniquement. Le pointeur est placé au début du fichier.

"w" : Ouvre un fichier en écriture uniquement. 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 uniquement. Le pointeur est placé à la fin du fichier s'il existe.

Exemple :

```
f = open("test.txt" , "r" ) # ouverture
```

L'ouverture d'un fichier se réalise grâce à la fonction `open` qui a deux paramètres :

- Le premier paramètre est une chaîne de caractères contenant le **nom** (avec un éventuel suffixe) du fichier.
- Le deuxième paramètre est aussi une chaîne de caractères contenant le **mode** de traitement du fichier. La valeur de ce paramètre est "r" pour la lecture d'un fichier. La fonction `open` retourne un descripteur de fichiers qui est un objet particulier. Ce descripteur doit être récupéré lors de l'appel dans une variable (ici `f`) pour pouvoir être manipulé par la suite. Ainsi, sous Python, l'accès aux fichiers est assuré par l'intermédiaire d'un objet-fichier (ici `f`) que l'on crée à l'aide de la fonction.

```
texte = f.read() # récupération du texte dans une variable
```

```
ligne = f.readline() # récupération de la ligne suivante
```

(les lignes sont séparées dans le fichier par le caractère spécial `\n`)

`ligne.split(sep)` découpe la chaîne de caractères `ligne` selon le séparateur `sep` : par exemple, si la ligne vaut "42,43,44", alors `ligne.split(",")` renvoie la liste ["42", "43", "44"].

`lignes = fichier.readlines()` donne la liste des lignes suivantes.

```
f.close ( ) # fermeture
```

La méthode conseillée en général est d'utiliser `f` comme itérateur, par exemple :

```
f = open("nom_du_fichier")
```

```
L = []
```

```
for ligne in f :           # on utilise f comme itérateur
```

```
    L.append = ligne.split(",")
```

```
f.close( ) # fermeture
```

A la sortie, `L` est une liste de listes, chacune des listes représente une ligne (découpée) du fichier.

`#` pour convertir par exemple la liste des chaînes en entier, (resp. flottant) avec `int` (resp `float`) :

```
L = [ int(x) for x in L ]
```

◀ Flottants :

3.14159, 2.014e10, a+b, a*b , a/b , a**b (exponentiation), max(a,b), abs(a)

Remarque : Pour chaque nombre flottant (en précision usuelle, dite double-précision), 52 bits sont utilisés pour stocker la mantisse, 1 bit pour le signe et 11 bits pour la puissance, donc au total **64 bits** (c'est-à-dire 8 octets).

L'infini (entier ou flottant) est codé par `inf` et est compatible avec les opérations : `max(2,inf)` et `2+inf` renvoient `inf`.

Fonctions classiques : `sqrt`, `sin`, `exp`, `log`, `abs`, `pi`, `floor` et `ceil` (à valeurs flottantes).

◀ Entiers :

`a//b` et `a%b` renvoient le quotient et le reste de a par b . Par exemple, `n%2 == 0` ssi n est pair.

Si x est un flottant, `int(x)` renvoie la partie entière de x (à distinguer de `floor(x)` qui renvoie un flottant).

Il faut 1 octet (8 bits) pour stocker un entier compris compris entre 0 et $255 = 2^8 - 1$. Ainsi, une image 2D en 3 couleurs (RGB) de taille 1000×1000 où l'intensité des couleurs est comprise entre 0 et 255 demande 3 MégaOctets.

Remarque : Le type `int` correspond à des entiers de 32 bits

En PYTHON3, on peut utiliser des entiers illimités de plus de 32 bits avec le type `long`.

◀ **La mémoire vive** d'un ordinateur est donné en Mo ou Go (mégaoctets ou gigaoctets).

On peut identifier les gigaoctets (10^9 octets) et les gibioctets ($2^{30} \simeq 10^9$).

La puissance est donnée en Hertz (fréquence de l'horloge) : elle correspond au nombre d'opérations élémentaires par seconde ; on peut aussi donner la puissance par exemple en donnant le nombre d'opérations flottantes par seconde.

Par exemple, un ordinateur de 1GHertz et de mémoire 32Go sature sa mémoire en quelques minutes.