

Créer, lire et écrire
dans un fichier en python

La fonction open

```
fichier = open("data.txt", "r")
```

Les principaux types d'ouverture

- **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.



Tout fichier doit être fermé !

```
fichier.close()
```

Lire le contenu d'un fichier

```
fichier = open("data.txt", "r")
```

Ouverture en lecture

```
f = fichier.read()
```

← lit tout le fichier

```
l = fichier.readline()
```

← lit une ligne du fichier et se positionne sur la suivante

```
ll = fichier.readlines()
```

← lit toutes les lignes du fichier et renvoie la liste des lignes

```
fichier.close()
```

Lire le contenu d'un fichier ligne par ligne

```
fichier = open("data.txt", "r")
```

```
l1 = fichier.readline() ← lit la 1e ligne
```

```
l2 = fichier.readline() ← lit la 2e ligne
```

```
fichier.close()
```

Une fois une ligne lue, le programme pointe automatiquement sur la ligne suivante.

Si la fin du fichier est atteinte la ligne est une chaîne de caractères vide : "".

Une ligne lue contient le contenu de la ligne suivi du caractère de retour à la ligne "\n" .

Ecrire dans un fichier

Ouverture en écriture

```
fichier = open("data.txt", "w")
```

*Ajout d'un
retour à la ligne*

```
fichier.write("Premiere ligne"+"\\n")
```

```
fichier.write("Deuxieme ligne"+"\\n")
```

```
fichier.close()
```

Les images
au format
Portable pixmap
(.pbm, .pgm, .ppm)

Structure des fichiers PBM, PGM et PPM

Les fichiers PBM, PGM ou PPM sont composés sur la même base :

- le **nombre magique** du format (P1, P2... P6). Il indique le type de format (PBM, PGM, ou PPM) et la variante (binaire ou ASCII) ;
- la **largeur de l'image** (nombre de pixels);
- la **hauteur de l'image** (idem);
- les **données de l'image** : Succession des valeurs associées à **chaque pixel** (l'image est codée ligne par ligne en partant du haut, chaque ligne est codée de gauche à droite.).

Toutes les lignes commençant par # sont ignorées (commentaires).

Les données précédentes sont séparées par un caractère d'espacement (espace, tabulation, nouvelle ligne) ;

Exemple de fichier PBM

Magic Number : P1 = Ascii en noir et blanc

Ligne facultative
de commentaires

Largeur
de l'image

Hauteur
de l'image

Pixels noir

Pixels blanc

```
P1
# Un exemple bitmap de la lettre "J "
7 10
0 0 0 0 0 0 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
```



Exemple de fichier PGM

Magic Number : P2 = Ascii en niveau de gris

Taille de l'image

P2

Affiche le mot "FEEP"

24 7

15

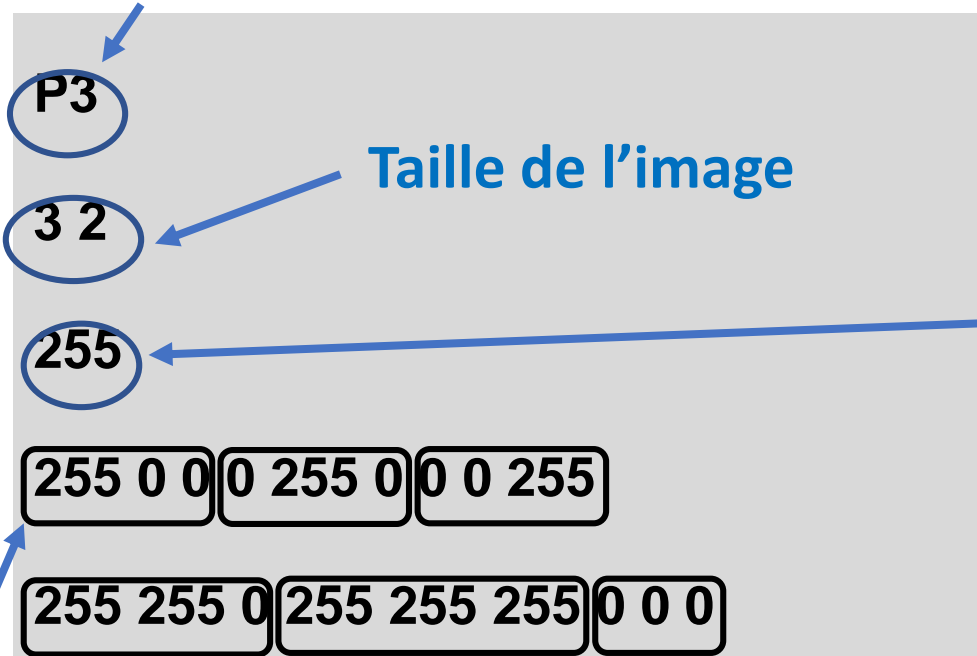
Chaque niveau de gris est codé par une valeur entre 0 et 15, proportionnellement à son intensité.
Un pixel noir est codé par la valeur 0, un pixel blanc est codé par la valeur 15.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	3	3	3	0	0	7	7	7	7	0	0	11	11	11	11	0	0	15	15	15	15	0
0	3	0	0	0	0	0	7	0	0	0	0	0	11	0	0	0	0	0	15	0	0	15	0
0	3	3	3	0	0	0	7	7	7	0	0	0	11	11	11	0	0	0	15	15	15	15	0
0	3	0	0	0	0	0	7	0	0	0	0	0	11	0	0	0	0	0	15	0	0	0	0
0	3	0	0	0	0	0	7	7	7	7	0	0	11	11	11	11	0	0	15	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FEEP

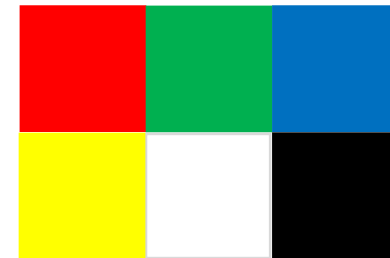
Exemple de fichier PPM

Magic Number : P3 = Ascii en couleur



Chaque pixel est codé par trois valeurs (rouge, vert et bleu).

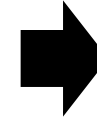
Chaque composante (rouge, verte et bleue) est codée par une valeur entre 0 et **255**, proportionnellement à son intensité.



1^{er} pixel : rouge à 100%, vert à 0% et bleu à 0% : pixel rouge

Quelques remarques

```
P3
3 2
255
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```



```
P3
3 2
255
255
0
0
0
255
255
255
0
255
255
255
0
0
0
```

Magic number

Dimension de l'image

Intensité max des couleurs

1^{er} pixel : Dans l'ordre les composantes rouge, verte et bleue

2^{er} pixel

- Dans un fichier Ascii, une ligne ne peut contenir, au maximum, que 70 caractères.
On écrit souvent le fichier avec une seule valeur par ligne.
(l'espace ou le retour à la ligne sont considérés comme des séparateurs)
- Pour provoquer un **retour à la ligne** dans le fichier après avoir écrit une valeur, on doit ajouter le caractère « retour à la ligne » : "\n".
Par exemple :
fichier.write(255)
fichier.write("\n")