

Lecture d'images et compression

juillet 2020

1 Boîte à outils

On rappelle le minimum vital pour ouvrir et lire des fichiers en Python, et effectuer des traitements simples sur des chaînes de caractères.

Manipulations de base

Quoi	Comment
Ouverture en lecture	<code>monfichier = open('lenom.txt', 'r')</code>
Ouverture en écriture	<code>monfichier = open('lenom.txt', 'w')</code>
Fermeture	<code>monfichier.close()</code>
Lire les n caractères suivants (renvoie une chaîne vide en fin de fichier)	<code>chaîne = monfichier.read(n)</code>
Stocker la <i>ligne</i> suivante dans une chaîne de caractères	<code>ligne = monfichier.readline()</code>
Parcourir toutes les lignes	<code>for ligne in monfichier:</code> <code>...</code>
Écrire dans le fichier... (et avec un retour à la ligne)	<code>monfichier.write('plof')</code> <code>monfichier.write('plof\n')</code>

Split et strip

Quoi	Comment
Séparer une chaîne selon les symboles ':' ou selon les tabulations	<code>toto.split(':')</code> <code>toto.split('\t')</code>
Supprimer les blancs en début et en fin de ligne	<code>toto.strip()</code>

Pour aller plus loin :

Comme d'habitude, <https://docs.python.org> et/ou `help(str)` pour aller plus loin.

2 Lecture et écriture dans un fichier PBM

Consultez la spécification du format d'images PBM, sur Wikipedia par exemple.

EXERCICE 1 (LECTURE D'UN FICHIER PBM)

Écrivez une fonction Python qui ouvre un fichier au format PBM et construit la matrice d'entiers (0 ou 1) correspondant.

EXERCICE 2 (UNE TRANSFORMATION SIMPLE)

Écrivez une fonction Python qui prend en argument une image représentée par une matrice d'entiers, et transforme cette image en son négatif.

EXERCICE 3 (ÉCRITURE D'UN FICHIER PBM)

Écrivez une fonction Python qui prend en arguments un nom de fichier et une matrice d'entiers (0 ou 1), et qui écrit l'image correspondante au format PBM dans le fichier demandé.

Testez vos fonctions sur quelques images PBM.

3 Compression Run Length Encoding

On s'intéresse ici à la compression RLE d'un fichier PBM.

On va profiter du caractère très particulier des données utilisées ici : comme on n'a qu'une alternance de 0 et de 1, il suffit de mémoriser la taille des blocs successifs de valeurs identiques.

EXERCICE 4 (COMPRESSION RLE)

Écrire une fonction de compression d'une image PBM suivant ce principe : chaque suite de bits identiques sera remplacée par un unique entier représentant sa longueur.

Il peut être utile pour commencer « d'aplatir » la matrice sous forme d'une liste.

EXERCICE 5 (ENREGISTREMENT DE L'IMAGE COMPRESSÉE)

Écrire une fonction permettant d'enregistrer votre image compressée dans un fichier au format de votre choix.

Attention :

- il faudra tout de même stocker les dimensions de l'image (sous une forme qu'on ne cherchera pas à compresser) ;
- il faut trouver un moyen de signaler si le premier bit de l'image est un 0 ou un 1 ;
- dans un premier temps (pour mettre au point votre programme), il est conseillé d'écrire en clair (sous forme ASCII) les longueurs des séquences de bits identiques, séparées par des espaces.

EXERCICE 6 (LECTURE DE L'IMAGE COMPRESSÉE)

Écrivez également une fonction qui permet de lire un fichier compressé selon votre format.

EXERCICE 7 (UN FORMAT PLUS COMPACT)

Améliorez la méthode précédente pour représenter la longueur d'une séquence non plus en ASCII mais à l'aide de la valeur de chaque octet.

- Comment procéderez-vous pour une séquence de longueur supérieure à 255 ?
- Pour écrire une liste de valeurs entières entre 0 et 255 chacune sur un octet, en Python vous pouvez utiliser les lignes de code suivantes :

```
>>> L = [42, 17, 250]
>>> f = open('monfichier', 'wb')
>>> f.write(bytes(L))
```

Notez le mode d'ouverture 'wb' pour le fichier (écriture binaire). On peut de même ouvrir un fichier en lecture binaire avec 'rb'.

4 Pour aller plus loin

EXERCICE 8 (NIVEAUX DE GRIS)

Reprenez le travail des deux sections précédentes, et adaptez-le pour le format PGM, qui permet de représenter des images en niveaux de gris.

Quelques points de vigilance :

- Le négatif d'une image dépend de la valeur maximale autorisée pour les niveaux de gris.
- Il faudra effectuer une « vraie » compression RLE pour tenir compte des différentes valeurs de niveaux de gris possibles.

EXERCICE 9 (INDEXATION)

Écrivez une série de fonctions permettant de :

- lire ou écrire une image dans un fichier au format PPM (en couleurs donc)
- déterminer les 256 couleurs les plus fréquentes dans une image donnée sous forme matricielle
- compresser une telle image par indexation et l'enregistrer sous un format de votre choix
- décompresser une image représentée sous votre format