

---

# Transformer les images

4/12 et 11/12

## 1 Travail à faire

1. Lire et comprendre les exercices corrigés proposés.
2. Traiter les autres exercices.
  - (a) Certains sont à déposer dans les casiers numériques de vos enseignants pour le 16/12 : exercices 3, 6.
  - (b) Un corrigé pour les autres exercices sera envoyé dans vos casiers après la séance du 11/12. Ils devront être travaillés et les méthodes utilisées devront être maîtrisées.

## 2 Découverte du module PIL

Le module PIL permet de manipuler un fichier image (reconnaissance automatique de la largeur et de la hauteur en pixels de l'image, création d'une grille de pixels, chaque ligne de la grille correspondant à une ligne de pixels, idem pour les colonnes).

### Exercice avec corrigé 1

Dans le dossier `enonces/B`, on trouvera une image au format `pgm` binaire.  
Utiliser le module PIL pour inverser les niveaux de gris de cette image.

#### Une résolution

Le code du programme utilisant le module PIL est donné ci-dessous. Vous testerez ce programme avec le fichier `enonces/B/inversionPIL.py`.

## Python

```
1 # -*- coding: utf-8 -*-
2
3 from PIL import Image
4
5 # ouverture d'une image au format pgm binaire :
6 imageSource=Image.open("Lyceebinaire.pgm")
7 # remarque : imageSource est un nom de variable, vous pouvez mettre un autre nom à la place.
8 # sa largeur et sa hauteur en pixels :
9 largeur , hauteur=imageSource.size
10 # ouverture d'une nv image
11 # pour l'option "L", voir http://www.pythonware.com/library/pil/handbook/concepts.htm
12 # et pour Image.new(), voir http://www.pythonware.com/library/pil/handbook/image.htm
13 imageBut=Image.new("L", (largeur , hauteur))
14 # remarque : imageBut est un nom de variable, vous pouvez mettre un autre nom à la place.
15
16
17
18 # pour chaque ligne :
19 for y in range(hauteur):
20     #pour chaque colonne :
21     for x in range(largeur):
22         # code du pixel (niveau de gris)
23         p=imageSource.getpixel((x,y))
24         # inversion du niveau de gris :
25         q=255-p
26         # création du pixel correspondant dans la nv image :
27         imageBut.putpixel((x,y),q)
28
29
30 # sauvegarde de l'image créée :
31 imageBut.save("InversionAvecPil.pgm")
32 # on montre l'image :
33 imageBut.show()
```

On pourra explorer les possibilités à partir de la page <http://www.pythonware.com/library/pil/handbook/index.htm>.

### A retenir pour la suite

Le pixel en haut à gauche correspond au pixel de coordonnées (0,0). Le pixel en haut à droite est le pixel de coordonnées (largeur-1 , 0). Le pixel en bas à gauche a pour coordonnées (0 , hauteur-1). Le pixel en bas à droite correspond au pixel (largeur-1, hauteur-1).

### A retenir pour la suite

Pour une image en niveaux de gris, le code obtenu par `imageSource.getpixel((x,y))` est un nombre compris entre 0 et 255. Ci-après, nous utilisons des images couleurs, `p=imageSource.getpixel((x,y))` sera alors un tuple : `p=(p[0],p[1],p[2])`, ses trois composantes sont les composantes RVB, comprises entre 0 et 255.

Pour les différences entre "L" et "RGB" : <http://www.pythonware.com/library/pil/handbook/concepts.htm>. Consulter aussi la page <http://www.pythonware.com/library/pil/handbook/image.htm#image-getbands-method>.

### Exercice avec corrigé 2

Dans le dossier enonces/J, vous trouverez une image au format jpg.

---

L'objectif est ici d'ajouter autour de l'image une bordure d'épaisseur  $e$  pixels ( $e$  étant une variable du programme). Attention, cette bordure vient se placer autour de l'image, elle n'écrase pas les pixels existants.

**Une résolution**

Le code du programme ci-dessous peut être testé directement avec le fichier [enonces/J/cadre.py](#).

---

 Python

```
1 # -*- coding: utf-8 -*-
2
3 from PIL import Image # importation d'une partie du module PIL
4
5
6 bord=50 # épaisseur e de la future bordure
7 pixelbord=(255,160,0) # couleur RVB des pixels de la future bordure
8
9
10 imageSource=Image.open("bouleau.jpg") # ouverture de l'image
11 largeur, hauteur=imageSource.size # largeur et hauteur en pixels de l'image
12
13
14 lf=largeur+2*bord # largeur de l'image avec bordure
15 hf=hauteur+2*bord # hauteur de l'image avec bordure
16
17 imageBut=Image.new("RGB", (lf, hf)) # ouverture d'une nouvelle image
18
19
20
21 # on commence d'abord par recopier l'image dans le nouveau fichier
22 # en tenant compte des décalages dus aux bords que l'on va ajouter :
23 for y in range(hauteur): # pour chaque ligne du fichier source
24     for x in range(largeur): # pour chaque colonne du fichier source
25         p=imageSource.getpixel((x,y)) # code du pixel source
26         imageBut.putpixel((x+bord,y+bord),p) # création du pixel correspondant dans la nv
           image
27
28
29 # les boucles qui suivent correspondent aux bordures haute, basse, droite, gauche.
30 for y in range(bord):
31     for x in range(lf):
32         imageBut.putpixel((x,y), pixelbord)
33
34
35 for y in range(hauteur+bord, hf):
36     for x in range(lf):
37         imageBut.putpixel((x,y), pixelbord)
38
39
40 for x in range(bord):
41     for y in range(hf):
42         imageBut.putpixel((x,y), pixelbord)
43
44
45 for x in range(largeur+bord, lf):
46     for y in range(hf):
47         imageBut.putpixel((x,y), pixelbord)
48
49
50 imageBut.save("bouleaucadre.jpg") # sauvegarde de l'image créée
51 imageBut.show() # affichage de l'image
```

□

### 3 Quelques transformations d'images

#### Exercice avec corrigé 3

Transformer l'image de gauche (cf dossier enonces/D) en l'image de droite.



#### Une résolution

Symétrie axiale (cf fichier `enonces/D/symetrie.py`) :

#### Python

```
1  # -*- coding: utf-8 -*-
2
3  from PIL import Image
4  import pylab as pyl
5
6  # ouverture d'une image au format jpg :
7  imageSource=Image.open("crabe.jpg")
8  # largeur et hauteur en pixels de l'image
9  largeur , hauteur=imageSource.size
10
11 #définition de la nouvelle image
12 imageBut=Image.new("RGB" ,(largeur , hauteur))
13
14
15
16 # pour chaque ligne :
17 for y in range(hauteur):
18     #pour chaque colonne :
19     for x in range(largeur):
20         # code du pixel (niveau de gris)
21         p=imageSource.getpixel((x,y))
22         # création du pixel correspondant dans la nv image :
23         imageBut.putpixel((x,-y+hauteur-1),p)
24
25
26 # sauvegarde de l'image créée :
27 imageBut.save("sym_axe.jpg")
28 # on montre l'image :
29 imageBut.show()
```

#### Exercice 1

A l'aide des outils présentés ci-dessus, transformer l'image de gauche (cf dossier enonces/D) en l'image de droite.



### Exercice 2

Passer de l'image de gauche (cf dossier enonces/E) à l'image de droite.



## 4 Agir sur les couleurs.

### Exercice avec corrigé 4

Passer de l'image de gauche (cf dossier enonces/E) à l'image de droite.



#### Une résolution

Ce qu'il faut saisir ici, c'est que l'on doit agir sur tout un intervalle. Certains pixels dont le noir n'est pas nécessairement indiscernable à l'oeil n'ont pas le même code couleur.

Fichier [enonces/F/redcat.py](#).

## Python

```
1 # -*- coding: utf-8 -*-
2
3 from PIL import Image
4 import pylab as pyl
5
6 # ouverture d'une image au format jpg :
7 imageSource=Image.open("cats.jpg")
8 # largeur et hauteur en pixels de l'image
9 largeur ,hauteur=imageSource.size
10
11 imageBut=Image.new("RGB" ,(largeur ,hauteur))
12
13 # pour chaque ligne :
14 for y in range(hauteur):
15     #pour chaque colonne :
16     for x in range(largeur):
17         # code du pixel (niveau de gris)
18         p=imageSource.getpixel((x,y))
19         # transfo des noirs en rouge :
20         if 0<=p[0]<=90 and 0<=p[1]<=90 and 0<=p[2]<=90 : p=(255,0,0)
21         # création du pixel correspondant dans la nv image :
22         imageBut.putpixel((x,y),p)
23
24 # sauvegarde de l'image créée :
25 imageBut.save("redcat.jpg")
26 # on montre l'image :
27 imageBut.show()
```

□

### Exercice 3

Dans cet exercice, vous allez transformer une image en couleurs (cf l'image enonces/G/fruits.jpg) en une image en niveaux de gris, donc de passer de l'image de gauche à une image telle que celle de droite :



Vous pourrez vous aider d'une lecture du document suivant : [http://cache.media.eduscol.education.fr/file/ISN\\_Tle\\_S/27/3/lyceeGT\\_ressource\\_ISN\\_20\\_06\\_Tle\\_S\\_22\\_Traitement\\_images\\_1\\_218273.pdf](http://cache.media.eduscol.education.fr/file/ISN_Tle_S/27/3/lyceeGT_ressource_ISN_20_06_Tle_S_22_Traitement_images_1_218273.pdf).

Vous proposerez une fonction pour chaque formule proposée dans ce document et vous imaginerez (ou trouverez sur le web) également d'autres formules que vous testerez.

### Exercice 4

Essayer de passer la photo de campagne du dossier enonces/H en noir et blanc.

---

## 5 Effets sur les images

Il existe de nombreux algorithmes basés sur des outils mathématiques pour transformer les images ou obtenir certaines informations. Ces outils sortent pour l'essentiel des connaissances abordables en terminale.

Nous nous contenterons donc de donner ici un exemple d'effet simple à obtenir. Nous ne commenterons pas l'outil mathématique sous-jacent, mais vous pouvez comprendre comment la transformation se fait dans le cas particulier proposé.

### Exercice 5

Dans le dossier enonces/I, vous trouverez une image au format pgm binaire.

Chaque pixel est dans ce cas codé par un simple entier (entre 0 et 255) correspondant à un niveau de gris.

1. Si les entiers  $a, b, d, f, h, i$  sont tous compris entre 0 et 255, montrer que l'entier  $q = (-2a - b - d + f + h + 2i) // 8 + 128$  (où  $//$  permet d'obtenir le quotient de la division entière) est également compris entre 0 et 255.
2. L'objectif est maintenant de transformer l'image suivant le procédé ci-dessous.
  - (a) Tout pixel de la première ligne de pixels, de la dernière ligne de pixels, de la première colonne de pixels et de la dernière colonne de pixels sera remplacé par un pixel noir.
  - (b) Tout autre pixel sera codé par un niveau de gris calculé à partir des niveaux de gris de ses voisins.

|   |   |   |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

Le pixel central, codé dans l'image initiale par le niveau de gris  $e$ , sera codé dans l'image finale par  $q = (-2a - b - d + f + h + 2i) // 8 + 128$ .

Ce qui a été fait ici relève d'une technique plus générale pour le travail sur les images. Vous pourrez manipuler avec le logiciel gimp :

<http://docs.gimp.org/fr/plugin-convmatrix.html>.

## 6 Création d'une image simple par équations.

### Exercice 6

Créer une image carrée (par exemple de taille 100 sur 100) ayant la forme d'un X noir sur fond blanc, le X joignant les coins du carré.