

Thème 4. La photo numérique

source : en partie Hachette éducation " SNT seconde " programme 2019

Activité 3 – Dans l'image, les données des pixels. Corrections

Problème. Auparavant on a vu que les métadonnées sont enregistrées automatiquement dans le fichier image lors de l'enregistrement de la photo. Mais comment cette photographie est-elle codée dans son fichier ? Et comment peut-on la modifier ?

Les corrections sont en vert.

Remarques.

1- Nous allons parler dans cette activité de bits et d'octets, termes que nous avons déjà vu. Un **bit** est une unité d'information en informatique : c'est un 0 ou un 1.

Un **octet** est **8 bits**.

L'ordinateur fonctionne avec seulement avec des 0 et des 1, c'est donc un système binaire. On les regroupe par 8 pour travailler avec l'ordinateur.

2- Les **54 premiers octets** d'un fichier image correspondent aux **métadonnées** de l'image.

3- Le fichier image suivant est un fichier **bitmap**. C'est un **type de fichier image** comme jpeg, jpg ou png ou gif sont d'autres types de fichiers images.

Les deux premiers octets indiquent le type de fichier :
66 77 = fichier bitmap

Les quatre octets suivants donnent la taille du fichier :
134 000 = 134 octets

Les octets 19 à 22 donnent la largeur de l'image en pixels:
5 000 = 5 pixels

Les octets 23 à 26 donnent la longueur de l'image en pixels :
5 000 = 5 pixels

Ces octets donnent le nombre de bits pour coder chaque pixels :
24 000 = 24 bits,
soit 3 octets par couleur du système Rouge Vert Bleu

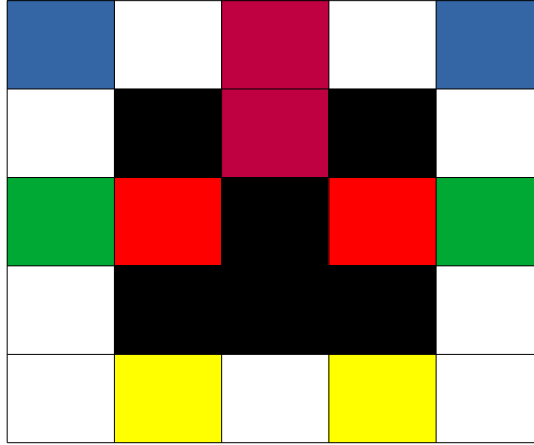
N° lignes	N° des octets															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	66	77	134	0	0	0	0	0	0	0	54	0	0	0	40	0
2	0	0	5	0	0	0	5	0	0	0	1	0	24	0	0	0
3	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	255	255	255	0	0	0	255	255	255	0
5	0	0	255	255	255	0	255	255	255	0	0	0	0	0	0	0
6	0	0	255	255	255	0	0	0	0	255	255	255	0	0	0	255
7	255	255	0	0	0	0	255	255	255	0	0	0	0	0	0	0
8	0	0	255	255	255	0	0	0	255	255	255	255	255	255	255	255
9	255	255	0	0	0	0										

Pour indiquer qu'une ligne de pixels est terminée et qu'on passe à la ligne du dessus, en repartant par la gauche, on note 0.
Il s'agit, ici, de la fin de la première ligne de pixels.

L'image en elle-même commence à l'octet 55. Dans un fichier bitmap, l'image est codée en commençant par le pixel en bas à gauche. Les octets 55, 56, 57 codent le pixel en bas à gauche de l'image. Les octets 58 à 63 codent les deux pixels suivants.

Doc 1. Le code derrière l'image

Image modifiée à partir du tableau précédent



Question 3. Modifie une image à l'aide d'un logiciel de retouche photo.

Vous pouvez vous aider des **aide 2** et **aide 3**.

a) Charger l'[image disponible ici](#) et enregistrez-la sous le nom 'chien' dans le dossier 'image' que vous venez de créer.

b) A l'aide d'un logiciel de retouche de photo, celui que vous voulez : PhotoShop, Gimp, Paint.net, PIXLR editor en ligne à l'adresse suivante, <https://pixlr.com/fr/e/> et il y en a d'autres, vous allez pouvoir agir sur les pixels de l'image pour faire une image en noir et blanc. (**Aide 3**)

Si vous ne trouvez pas. Ouvrir l'image avec le logiciel GIMP. Les actions qui suivent sont sensiblement les mêmes avec les autres logiciels.

Plusieurs manières existent pour obtenir une image en noir et blanc ou plutôt en niveau de gris. Je vous donne ici quelques suites de menus et onglets qu'il faut ouvrir :

- 1) Image / Mode / niveau de gris,
- 2) Couleurs / Teinte-Saturation / couleur primaire à ajuster : Maître et Saturation à -100 (curseur à gauche),
- 3) Couleurs / Teinte-Chroma / Chroma à -100 (curseur à gauche),
- 4) Couleurs / Saturation / Échelle à 0 (curseur à gauche)



Photo d'origine



Photo modifiée

c) Agir sur les pixels de l'image pour créer quatre photos à la manière d'Andy Warhol. (**Doc 2**)

A vous de jouer !

Pas fait !

Question 4. Modifier une image à l'aide d'un programme en Python.

Le programme suivant permet de passer d'une photo couleur à une photo en niveau de gris.

Vous pouvez aussi le [télécharger à partir de ce lien](#).

- a) Copiez et collez dans Pyzo ce programme.
- b) Enregistrez le programme sous le nom 'noir_et_blanc', par exemple, dans le dossier 'image' qui contient déjà l'image 'chien.jpg' et qui va être utilisée par ce programme.
- c) Vérifiez bien que les indentations du programme sont respectées dans Pyzo : lignes avec for j in range ... jusqu'à photo1_pil.putpixel...
- d) Puis exécutez le programme et observez le résultat.

- e) Dans le programme trouvez quelles sont les lignes qui codent pour le passage de la couleur au niveau de gris.

```
## -----Importation des librairies-----
import tkinter as tk

from PIL import Image, ImageTk
# -----
##----- Création de l'environnement -----
fenetre=tk.Tk()
fenetre.title("traitements d'images")
##----- Création des objets -----
#Un canevas pour afficher l'image
toile= tk.Canvas(fenetre,height=452, width=894, bg="black")
toile.grid()
#-----
##----- Chargement et affichage de l'image de départ -----
# Chargement image avec PIL
photo_pil = Image.open("chien.jpg")
# Longueur et hauteur de l'image
colonne, ligne = photo_pil.size
# Conversion en PhotoImage et affichage
photo = ImageTk.PhotoImage(photo_pil)
toile.create_image(10,10, anchor ="nw", image = photo)
#-----
### -----Traitement de l'image -----
#Copie de l'image de départ
photo1_pil = photo_pil.copy()
#Balayage de l'image pour prendre les couleurs des pixels puis changer les pixels de la nouvelle
image
for i in range(colonne):
    for j in range(ligne):
        r,g,b = photo_pil.getpixel((i,j))
        gris = int((r + g + b)/3)
        photo1_pil.putpixel((i,j), (gris,gris,gris))
# -----Conversion en PhotoImage et affichage -----
photo1 = ImageTk.PhotoImage(photo1_pil)
toile.create_image(452, 10, anchor ="nw", image = photo1)
# -----
## -----Maintient de la fenêtre ouverte pendant le traitement-----
fenetre.mainloop()
```



Résultat du programme Python :
photo d'origine à gauche, photo modifiée, à droite