



ACTIVITÉ 2 : L'IMAGE NUMÉRIQUE

Capacités attendues

Contenus

Traitement d'image

Capacités attendues

Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.

Note d'intention

Cette activité aborde la notion d'image numérique. À partir d'un logiciel comme Gimp, l'élève comprend la notion de composantes d'un pixel et ensuite la représentation d'une couleur. La manipulation des images se fait en utilisant Python. La première partie est débranchée. Un script Python est proposé. L'élève doit colorier un quadrillage représentant les pixels de l'image et répondre à des questions. Ensuite il vérifie son travail en recopiant son script et en l'exécutant. Il termine en créant son propre script afin d'obtenir le drapeau Danois. Dans une deuxième partie, le traitement d'une image est abordé en proposant une fonction « FiltreRouge ». Quelques questions sont posées pour vérifier la compréhension de certaines lignes (notamment la triple affectation qui est un choix pédagogique). Pour finir, l'élève modifie le script pour créer un filtre vert puis un filtre bleu et inverser les quantités de rouge et de bleu.

L'image numérique

Formats d'image

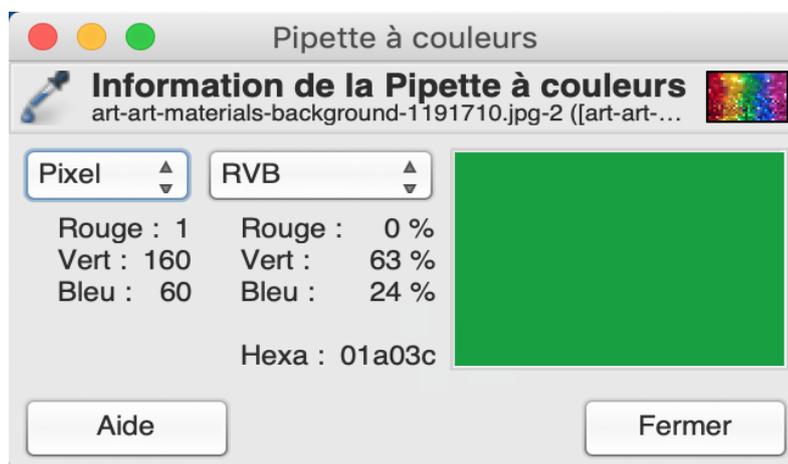
Le type d'un fichier dans l'ordinateur se reconnaît en examinant son extension (les caractères situés après le . dans son nom). Par exemple, un fichier PDF aura un nom du type NomFichier.pdf. Cette extension permet à l'ordinateur d'utiliser automatiquement le bon logiciel pour ouvrir le fichier et le visualiser à l'écran.

1. Citer plusieurs formats d'images et préciser l'extension associée.
2. Ouvrir le fichier `art-art-materials-background-1191710.jpg` (photo by Sharon)

Retrouvez éduscol sur :



- McCutcheon from Pexels) fourni avec le logiciel *Gimp*. Indiquer la définition de cette image.
3. Aller dans le menu *Outils* de *Gimp* et sélectionner *Pipette à couleurs*. Le pointeur de la souris est alors une pipette. En pressant les touches *Maj*, *Shift* et en cliquant dans l'image en même temps, une fenêtre d'information s'ouvre donnant les intensités de rouge, vert et bleu du pixel sélectionné.



Représentation de la couleur

Nous avons vu dans l'activité précédente que chaque pixel de l'écran étant composé de trois sous-pixels rouge, vert et bleu. La couleur d'un pixel d'une image est alors définie par trois valeurs, chacune associée à l'intensité d'une des trois couleurs primaires.

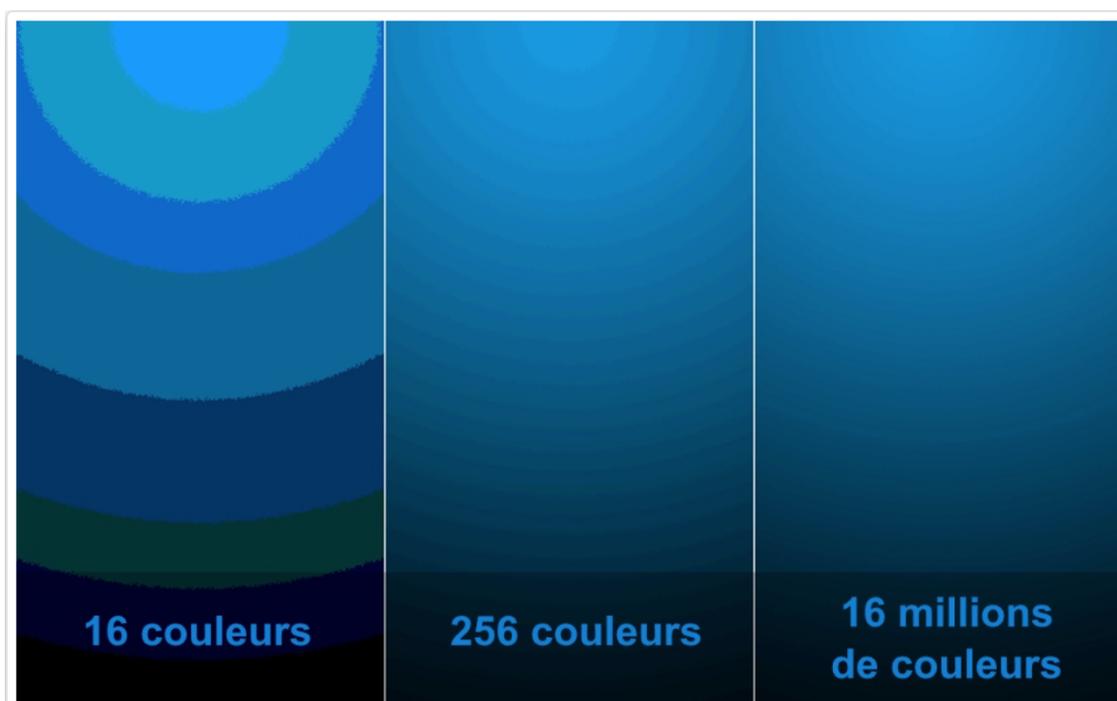
Définition

La profondeur de couleurs, dont l'unité est le nombre de bits par pixel (bpp) est un terme utilisé en informatique pour décrire le nombre de bits utilisés pour représenter la couleur d'un pixel de l'image.

Une plus grande profondeur de couleurs, ce qui nécessite un plus grand nombre de bits, permet une plus grande échelle de nuances dans les couleurs. L'illustration suivante montre l'effet produit sur l'échelle de bleu en utilisant une profondeur de plus en plus grande (on obtient 16 couleurs avec 4 bits, 256 couleurs avec 8 bits).

Retrouvez éducol sur :





La norme aujourd'hui est de coder la couleur avec 24 bits (8 bits par couleur primaire soit valeurs ce qui donne environ 16 millions de couleurs).

4. Compléter le tableau suivant des couleurs principales avec le triplet de valeurs qui convient, chaque canal couleur ayant soit l'intensité minimale (0) soit l'intensité maximale (255) (voir la synthèse additive de la couleur dans l'activité 1).

Couleur	Noir	Rouge	Vert	Bleu	Cyan	Jaune	Magenta	Blanc
Triplet	(0,0,0)	(255,0,0)					(255,0,255)	

Manipuler des images avec Python

Création d'une image pixel par pixel

5. Avec le script *Python* ci-dessous et à l'aide des explications données plus bas, colorier le quadrillage représentant les pixels de l'image construite de la couleur qui convient.

```
from PIL import Image

img = Image.new("RGB", (7,5), (255,255,255))
for x in range(7):
    img.putpixel((x,1), (255,0,0))
for y in range(5):
    img.putpixel((5,y), (0,0,255))
```

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							

Retrouvez éduscol sur :



Les principales instructions

- L'instruction `from PIL import Image` permet d'importer le module Image de la bibliothèque PIL.
- L'instruction `img = Image.new («RGB», (Largeur,Hauteur), (r,v,b))` permet de créer une image de couleur unie fixée par le triplet (r,v,b) et de dimension Largeur x Hauteur.
- L'instruction `for k in range (n)` : permet de réaliser une boucle POUR, la variable k variant de 0 à n-1, ce qui permet de répéter n fois les instructions qui la composent (l'indentation - le décalage à droite - permet d'identifier ces instructions).
- L'instruction `img.putpixel ((5,y), (r,v,b))` permet de donner la couleur définie par le triplet (r,v,b) au pixel de coordonnées (x,y).

6. Expliquer pourquoi le pixel (5,1) est-il bleu.
7. Ouvrir *EduPython* et recopier les instructions précédentes dans l'éditeur de script en ajoutant l'instruction `img.show()` qui permet d'afficher l'image obtenue. Faire interpréter le script par *EduPython* en cliquant sur la flèche verte et visualiser le résultat. Indiquer si votre coloriage est correct.
8. Modifier le code fourni afin d'ajouter une ligne de vert, une colonne de magenta et afin que le pixel soit noir.
9. Écrire un script pour produire le drapeau danois.

Filtre couleur

10. Recopier les instructions suivantes dans l'éditeur de script d'EduPython et tester les en cliquant sur la flèche verte.

```
img = Image.open ("art-art-materials-background-1191710.jpg")
img.show ()
```

L'instruction `img = Image.open(NomdeFichier)` permet de charger un fichier image par son nom et de la stocker dans la variable `img`.

11. Recopier le code suivant dans l'éditeur de script d'EduPython. Il permet de définir une fonction nommée *FiltreRouge*.

```
def FiltreRouge (NomFichier):
    img = Image.open (NomFichier)      # chargement de l'image
    Largeur, Hauteur = img.size        # récupération de ses dimensions

    # Double boucle pour parcourir tous les pixels de l'image
    for x in range (Largeur) :          # boucle sur les colonnes de l'image
        for y in range (Hauteur):       # boucle sur les lignes de l'image
            r,v,b = img.getpixel((x,y))
            img.putpixel ((x,y),(r,0,0))

    img.show ()      # affiche l'image modifiée
```

Retrouvez éducol sur :



L'instruction `img.getpixel ((x,y))` permet de récupérer la couleur d'un pixel (x, y) sous la forme d'un triplet de valeurs (les valeurs des trois composantes couleur).

11.1. Expliquer à quoi sert l'instruction `r, v, b = img.getpixel((x,y))`.

11.2. Expliquer à quoi sert l'instruction `img.putpixel((x,y), (r, v, b))`.

12. Indiquer comment modifier les deux instructions internes aux boucles pour produire un filtre vert.
13. Même question pour un filtre bleu.
14. Implémenter l'un des deux filtres.
15. Modifier le script afin d'invertir les quantités de rouge et de bleu.

Retrouvez éduscol sur :

