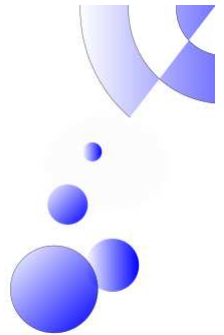
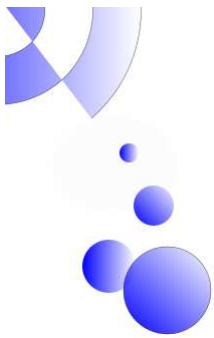




Table des Matières

I. Introduction	1
II. Comprendre les données d'une image avec Gimp	1
III. Couleurs	2
III. A Palette de couleurs	2
III. B Inverser les couleurs	4
III. C Niveaux de gris	4
III. C. 1 Définition	4
III. C. 2 Conversion d'une image couleur en niveaux de gris	4
IV. construire une image	5
V. Retoucher une image	6
VI. Projet	8





I. Introduction

Activité 1

Regarder la vidéo : [Photo numérique, du réel aux pixels ?](#)

MOOC SNT

1. Qu'est ce qu'une image numérique ?
2. Qu'est qu'une définition d'une photo ? Donner le calcul permettant de savoir la définition, mesurée en pixels, d'une image de 4000×6000 pixels.
3. Qu'est qu'une résolution d'une photo ?
4. Combien de couleurs sont nécessaires pour colorer un pixel ?

II. Comprendre les données d'une image avec Gimp

Activité 2

1. Ouvrir le logiciel Gimp, puis l'image ChuteEau.jpg disponible sur l'ENT.
2. Les métadonnées d'une image : Image > Métadonnées > Afficher les métadonnées
 - (a) Que signifient les initiales EXIF ?
 - (b) Quelle est la taille de l'image en pixels ? Quelle est la définition de l'image ?
 - (c) Comment se traduit l'unité Pouce en anglais.
 - (d) Donner la résolution de l'image en Dpi soit en pixel/Pouces.
 - (e) À quelle date et à quelle heure a été prise la photo ?
 - (f) Quelle est la marque et la modèle de l'appareil utilisé pour la photo ?
 - (g) Est-ce que le flash a été utilisé .
3. Définition et résolution de l'image :
 - (a) Faire un zoom de l'image (8:1). Comment est composée l'image ?
 - (b) Image > Échelle et taille de l'image : retrouver la définition de l'image et convertir la résolution de l'image en Pixels/cm.
 - (c) Si une résolution des de 300 dpi, quel calcul permet de retrouver la résolution en pixel/cm ?
 - (d) Image > Impression de l'image : quelle est la taille, en cm, de l'impression de l'image ? Retrouver le résultat affiché par des calculs, connaissant la définition de l'image et la résolution de l'image.
4. Couleur de l'image :
 - (a) Afficher la pipette de couleur : Outil > Pipette de couleur.
 - (b) Appuyer sur la touche MAJ puis n'importe où sur l'image.
 - (c) Pour un pixel donné de l'image donner :

- Sa position dans l'image,
 - le nombre de bits pour chaque des couleurs et le pourcentage auquel il correspond.
- (d) Que fait la commande inversée de l'image ? Couleur > Inversée.
Revenir à l'image initiale.
- (e) Que fait le mode niveau de gris ? Image > ? Image > Mode > niveau de gris.
Revenir à l'image initiale.

🔗 Exercice 1

1. Soit une image I de définition 8×8 pixels et de résolution 2 pixels/cm. Quelle est la taille de la largeur et de la longueur, en cm, de cette image ?
2. Si on double la définition de la largeur et la longueur de cette image, pour une même taille de l'image Calculer :
 - (a) le nombre de pixels de la nouvelle image I_1 .
 - (b) le nombre de pixels à retoucher à partir de l'image initiale.
 - (c) la résolution de l'image en pixel/cm.

III. Couleurs

III. A. Palette de couleurs

🔗 Activité 3

1. Une couleur est constituée d'un niveau de rouge, d'un niveau de vert, d'un niveau de bleu, variant chacun sur les entiers de 0 à 255.
Donner le nombre de couleurs possible.
2. Recopier et compléter le tableau suivant en tenant compte de l'exemple pour la couleur noire :

Couleur	Red	Green	Blue	Pourc. Rouge	Pourc. Vert	Pourc. Bleu	Octet 1	Octet 2	Octet 3	Code HEX-adécimal
Noir	0	0	0	0%	0%	0%	0000 0000	0000 0000	0000 0000	# 000000
Blanc										
Rouge										
Vert										
Bleu										

Note : Suivant votre niveau : vous pouvez les codes binaires ou les codes hexadécimaux sur Internet ou les calculer à la main, ou les déterminer sur Tableur à partir de formule tableur (Technologie), ou faire un programme Python de conversion (spécialité Mathématiques - NSI).

3. Ouvrir le programme Python PaletteCouleur.py, puis recopier et compléter le tableau suivant :

Couleur	Red	Green	Blue	Pourc. Rouge	Pourc. Vert	Pourc. Bleu	Octet 1	Octet 2	Octet 3	Code HEX-adécimal
Noir	0	0	0	0%	0%	0%	0000 0000	0000 0000	0000 0000	# 000000
	255	0	120							
	255	255	0							
	120	120	120							
	180	120	180							

Note : Suivant votre niveau : vous pouvez les codes binaires ou les codes hexadécimaux sur Internet ou les calculer à la main, ou les déterminer sur Tableur à partir de formule tableur (Technologie), ou faire un programme Python de conversion (spécialité Mathématiques - NSI).

4. *Facultatif*: spécialité NSI recopier et interpréter chaque ligne du programme proposé :

```
1 from tkinter import*
2
3 global rectangle
4
5 def couleur () :
6     rectangle ["background"] = "#%02x%02x%02x" % (int(varR.get()),int(varG.get()),int(varB.get()))
7
8 #fenêtre application
9 app=Tk()
10 app.title("Couleur RGB (Red - Green - Blue)")
11
12 #frames
13 Frame0= Frame(app,borderwidth=2, relief=GROOVE)
14 Frame0.pack(side=LEFT, padx=10, pady=10)
15 Frame1 = Frame(Frame0, borderwidth=2, relief=GROOVE)
16 Frame1.pack(side=LEFT, padx=10, pady=10)
17 Label(Frame1, text="Rouge").pack(padx=10, pady=10)
18 Frame2 = Frame(Frame0, borderwidth=2, relief=GROOVE)
19 Frame2.pack(side=LEFT, padx=10, pady=10)
20 Label(Frame2, text="Vert").pack(padx=10, pady=10)
21 Frame3 = Frame(Frame0, borderwidth=2, relief=GROOVE)
22 Frame3.pack(side=LEFT, padx=10, pady=10)
23 Label(Frame3, text="Bleu").pack(padx=10, pady=10)
24 Frame4 = Frame(app, borderwidth=2, relief=GROOVE)
25 Frame4.pack(side=LEFT, padx=10, pady=10)
26 Label(Frame4, text="Couleur").pack(padx=10, pady=10)
27
28 #scale
29 varR = DoubleVar()
30 varG = DoubleVar()
31 varB = DoubleVar()
32 scale1=Scale(Frame1,from_=0,to=255, variable=varR)
33 scale2=Scale(Frame2,from_=0,to=255,variable=varG)
34 scale3=Scale(Frame3,from_=0,to=255,variable=varB)
35 scale1.pack()
36 scale2.pack()
37 scale3.pack()
38
39 #bouton valider
40 ButtonValider = Button(Frame0, text ="Valider", command = couleur)
41 ButtonValider.pack(side=LEFT, padx=10, pady=10)
42
43 #rectangle de couleur
44 rectangle = Canvas(Frame4, width=150, height=150, background="black")
45 rectangle.pack()
46
47
48
49 app.mainloop()
```

PaletteCouleur.py

III. B. inverser les couleurs

🔗Activité 4

La couleur inversée du triplet RGB (120 ; 60 ; 180) est le triplet RGB (255 – 120 ; 255 – 60 ; 255 – 180) soit le triplet (135 ; 195 ; 75).

1. Recopier et compléter les deux tableaux de l'activité précédente en donnant les codes des couleurs inversées de chacune des couleurs proposées.
2. *Facultatif : spécialité NSI*, compléter le programme PaletteCouleur.py pour obtenir dans un Frame 5 le code RGB inverse et le rectangle de la couleur associé.

III. C. Niveaux de gris

III. C. 1. Définition

🔗Activité 5

1. En reprenant les activités précédentes, combien d'octets sont nécessaires pour coder la couleur d'un pixel ?
2. Sur la palette de couleurs PaletteCouleur.py, quelles couleurs obtenez vous si vous choisissez le même niveau de rouge, de vert et de bleu ?
3. Définir une image en niveau de gris, combien de couleurs y a-t-il en niveau de gris ?
4. Combien d'octet sont nécessaires pour coder une image en niveau de gris ?
5. Une image couleur contient
6. *Facultatif : spécialité NSI* À partir du fichier PaletteCouleur.py, déterminer une échelle définissant les niveaux de gris.

III. C. 2. Conversion d'une image couleur en niveaux de gris

🔗Activité 6

Trois méthodes peuvent être utilisées pour convertir une couleur RGB $L(varR ; varV ; varB)$ en niveaux de gris :

- Méthode 1 : chaque valeur du triplet de gris reçoit $\frac{max(L) + min(L)}{2}$
 - Méthode 2 : chaque valeur du triplet de gris reçoit $\frac{varR + varV + varB}{3}$
 - Méthode 3 : chaque valeur du triplet de gris reçoit $0,2125.varR + 0,7154.varV + 0,0721varB$.
Il s'agit de la recommandation CIE 709 utilisée comme standard de la télévision HDTV.
Cette recommandation remplace la recommandation CIE 601 utilisée pour la télévision analogique Hertzienne (pour information $0,299.varR + 0,587.varV + 0,114.varB$, il semble que Gimp utilise cette recommandation).
1. Reprendre les 2 tableaux de l'activité 3 et donner le niveau de gris associé à chaque couleur, avec la méthode de votre choix, le tableau ne comportera que le niveau de gris et l'octet associé et son code HEXadécimal.
 2. *Facultatif : spécialité NSI* Compléter le fichier PaletteCouleur.py par une Frame6 donnant le code et un rectangle de la couleur de niveau de gris (méthode de votre choix) associé à la couleur choisit par l'utilisateur.

IV. construire une image

Activité 7

Avec le module Pillow de Python, nous allons construire des images au format PNG.

1. Recopier et exécuter le code suivant :

```
1 from PIL import Image
2 im = Image.new("RGB" , (300,300) , "grey")
3 type(im)
4 im.save("ImageFond.png" , "PNG")
```

ImageFond.py

2. Compléter le code par la ligne suivante :

```
1 from PIL import Image
2 im = Image.new("RGB" , (300,300) , "grey")
3 im.putpixel( (10,10) , (0,0,0) )
4 im.save("ImageFondPixel.png" , "PNG")
```

ImageFondPixel.py

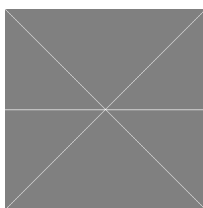
Préciser à quoi sert la méthode .putpixel et expliquer comment elle fonctionne.

3. Modifier le code pour tracer une ligne horizontale au centre de l'image :

```
1 from PIL import Image
2 im = Image.new("RGB" , (300,300) , "grey")
3 im.putpixel( (10,10) , (0,0,0) )
4 im.save("ImageFondPixel.png" , "PNG")
```

ImageFondPixel.py

Compléter le code pour obtenir la figure suivante :



4. Faire au moins un des drapeaux suivants :



V. Retoucher une image

Activité 8

Recopier et exécuter le code du programme Python suivant au fur et à mesure pour remarquer les retouches de l'image. Pour l'exécution du programme il vous faudra placer l'image PhotoRetouche.png dans le même dossier. Aussi le programme peut s'appliquer avec n'importe quelle image. Après chaque question exécuter le programme.

1. Bibliothèque Pillow Image et ImageFilter, ouvrir une image :

```
1 from PIL import Image # Import de la classe Image
2 from PIL import ImageFilter #Import de la classe ImageFilter pour filtrer une image
3
4 img = Image.open("PhotoRetouche.png") #ouverture du fichier image PhotoRetouche au format png
5 print(img.format, img.size, img.mode) #information sur l'image
6
7 img.show() #montrer l'image
```

RetoucheImage.py

Donner les informations sur l'image.
Quelle fonction permet d'afficher l'image.

2. Découper une image :

```
1 decoupe = img.crop((400, 400, 800, 800))
2 decoupe.show()
```

RetoucheImage.py

La région à découper est définie par 4 coordonnées (gauche, haut, droit, bas) en pixels. Dans le système de coordonnées d'une image, l'origine (0,0) se situe en haut à gauche.

3. Coller une image dans une autre :

```
1 img.paste(decoupe, (0,0, 400, 400))
2 img.show()
```

RetoucheImage.py

coller une découpe, l'image à coller doit être à la taille de la découpe.

4. Rotation d'une image :

```
1 img1 = img.rotate(45)
2 img1.show()
```

RetoucheImage.py

Ici la rotation est de 45 degrés.

5. Changer les dimensions d'une image :

```
1 img2 = img.resize((100, 250))
```

RetoucheImage.py

6. Transpositions d'une image :

```
1 img3 = img.transpose(Image.FLIP_LEFT_RIGHT)
2 img4 = img.transpose(Image.FLIP_TOP_BOTTOM)
3 img5 = img.transpose(Image.ROTATE_90)
4 img6 = img.transpose(Image.ROTATE_180)
5 img7 = img.transpose(Image.ROTATE_270)
6
7 img2.show()
8 img3.show()
9 img4.show()
10 img5.show()
11 img6.show()
12 img7.show()
```

RetoucheImage.py

7. Convertir une image en niveau de gris :

```
1 img_ng = img.convert("L")
2 img_ng.show()
```

RetoucheImage.py

8. Traiter une image avec un filtre :

```
1 decoupe = img.crop((850, 600, 1000, 700))
2 decoupe_net = decoupe.filter(ImageFilter.DETAIL)
3 decoupe.show()
4 decoupe_net.show()
```

RetoucheImage.py

Comparer la qualité de l'image

9. Augmenter la luminosité des pixels de l'image :

```
1 img_lum = img.point(lambda i: i * 1.5)
2 img_lum.show()
```

RetoucheImage.py

10. Traiter une image par Pixels :

```
1 pixels = img.getpixel((0,0))
2 print(pixels)
3 img.putpixel((0,0), (255, 0, 0))
4 img.show()
```

RetoucheImage.py

Que fait la fonction getpixel ? la fonction putpixel ?

VI. Projet

🔗Activité 9

À vous de jouer, utiliser le logiciel Gimp ou programmer en Python pour créer une image, faire un jeu avec des images, truquer une image etc...

