https://mesmaths.com/spip.php?article416

5-au cas où

- SNT - 7-Photographie numérique -

Date de mise en ligne : samedi 4 mai 2019

si le programme ne fonctionne pas correctement sous Edupython (pour un problème de bibliothèque non importée) ... une alternative

code à copier/coller

```
"""important : l'image traitee doit se trouver dans le meme repertoire que le programme"""

 """avec cette methode, les composantes r, v, et b sont des nombres compris entre 0 et 1"""

 #importation de bibliotheques necessaires au fonctionnement du programme
 import matplotlib.image as mpimg
 import numpy as np
 import matplotlib.pyplot as plt

 name = ("monchat.png") #name prend le nom de l'image à traitée (image en format .png) ; le nom contient
l'extension
 im = mpimg.imread(name)#on lit le document saisi par la fonction mp.img()
 im = im[:,:,:3] # pour faire disparaitre le 4eme plan -> chaque pixel est constitue de 3 valeurs qui
correspondent au code rgb
 for i in range(im.shape[0]):#le long des colonnes
 for j in range(im.shape[1]):#le long des lignes
 r, v, b = im[i, j] #ecriture etrange ... car im[i,j] est une liste de trois valeurs -> que l'on stocke dans
r, v et b
 #le traitement se fait ici -> c'est là que va intervenir
 im[i, j] = (0,0,b)#on ne garde que la composante bleue

 plt.imshow(im)#on cree l'image traitee
 plt.show()#on affiche l'image traitee -> pour la conserver, il faudra l'enregistrer manuellement
```

deux images à télécharger

si on utilise le programme ci-dessus, il faut obligatoirement prendre l'image en .png.

si on utilise le programme initialement prévu, on prend a le choix entre .png et .jpg

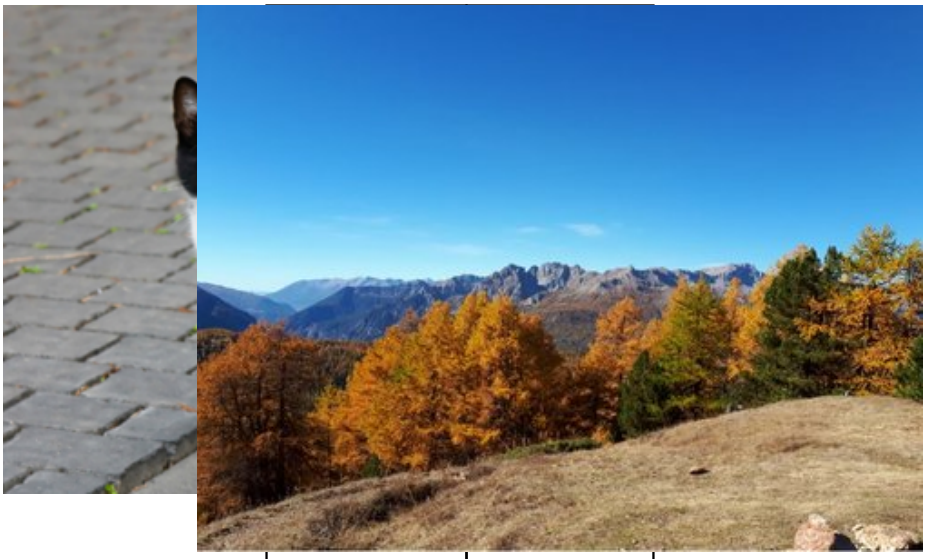

image en **.png**	image en .jpg

programme pour mélanger deux images

```
"""IMPORTANT : les images importées devront se trouver dans le même fichier que ce programme. L'image modifiée sera elle aussi enregistrée dans ce dossier. On mélange les images 1 et 2 (la plus grande est l'image 1)

 ###traitement d'une photo
 #importation obligatoire pour utiliser la fonction analysant le code RGB pixel par pixel d'une image
 from PIL import Image

 #fonction permettant de faire un melange plus ou moins marque
 coef=0.5 #coef est la proportion de la première image
 def mel(a,b):
 return int(coef*a+(1-coef)*b)

 #importation des deux images : la plus grande est image1
 imageSource1=Image.open("pays.jpg")#nom de l'image1 à saisir A MODIFIER
 imageSource2=Image.open("girod.jpg")#nom de l'image2 à saisir A MODIFIER

 #parametrages des images
 largeur1,hauteur1=imageSource1.size#prend en compte la largeur de l'image1 et sa hauteur
 largeur2,hauteur2=imageSource2.size#prend en compte la largeur de l'image2 et sa hauteur
 dech=int((largeur1-largeur2)/2)#centrage horizontal
 decv=int((hauteur1-hauteur2)/2)#centrage vertical
 #hauteur=min(hauteur1,hauteur2)

 imageModif=Image.new("RGB",(largeur1,hauteur1))#définie une nouvelle image vierge de même dimension que l'image 1
```

```
##4 zones ne faisant apparaitre que l'image la plus grande
#zone en haut sur toute la largeur
for y in range(decv): # y varie de 0 a  hauteur - 1
for x in range(largeur1):
p1=imageSource1.getpixel((x,y)) # p est la valeur RGB du pixel de l'image1
r1=p1[0]#première valeur de la liste p -> niveau de r
v1=p1[1]#deuxième valeur de la liste p -> niveau de v
b1=p1[2]#troisème valeur de la liste p -> niveau de b
imageModif.putpixel((x,y),p1) # on affecte les valeurs de de p dans la nouvelle image pour la créér

#zone en bas sur toute la largeur
for y in range(decv+hauteur2,hauteur1): # y varie de 0 a  hauteur - 1
for x in range(largeur1):
p1=imageSource1.getpixel((x,y)) # p est la valeur RGB du pixel de l'image1
r1=p1[0]#première valeur de la liste p -> niveau de r
v1=p1[1]#deuxième valeur de la liste p -> niveau de v
b1=p1[2]#troisème valeur de la liste p -> niveau de b
imageModif.putpixel((x,y),p1) # on affecte les valeurs de de p dans la nouvelle image pour la créér

#zone sur la gauche sur toute la hauteur
for y in range(hauteur1): # y varie de 0 a  hauteur - 1
for x in range(dech):
p1=imageSource1.getpixel((x,y)) # p est la valeur RGB du pixel de l'image1
r1=p1[0]#première valeur de la liste p -> niveau de r
v1=p1[1]#deuxième valeur de la liste p -> niveau de v
b1=p1[2]#troisème valeur de la liste p -> niveau de b
imageModif.putpixel((x,y),p1) # on affecte les valeurs de de p dans la nouvelle image pour la créér

#zone sur la droite sur toute la hauteur
for y in range(hauteur1): # y varie de 0 a  hauteur - 1
for x in range(dech+largeur2,largeur1):
p1=imageSource1.getpixel((x,y)) # p est la valeur RGB du pixel de l'image1
r1=p1[0]#première valeur de la liste p -> niveau de r
v1=p1[1]#deuxième valeur de la liste p -> niveau de v
b1=p1[2]#troisème valeur de la liste p -> niveau de b
imageModif.putpixel((x,y),p1) # on affecte les valeurs de de p dans la nouvelle image pour la créér

#zone ou il y a le melange
for y in range(decv,decv+hauteur2): # y varie de 0 a  hauteur - 1
for x in range(dech,dech+largeur2):
p1=imageSource1.getpixel((x,y)) # p est la valeur RGB du pixel de l'image1
p2=imageSource2.getpixel((x-dech,y-decv)) # p est la valeur RGB du pixel de l'image2
r1=p1[0]#première valeur de la liste p -> niveau de r
v1=p1[1]#deuxième valeur de la liste p -> niveau de v
b1=p1[2]#troisème valeur de la liste p -> niveau de b
r2=p2[0]#première valeur de la liste p -> niveau de r
v2=p2[1]#deuxième valeur de la liste p -> niveau de v
b2=p2[2]#troisème valeur de la liste p -> niveau de b
r=mel(r1,r2)
v=mel(v1,v2)
b=mel(b1,b2)
p=(r,v,b)
```

```
imageModif.putpixel((x,y),p)

#sauvegarde et visualisation de l'image
imageModif.save("image1_image.jpg")#on nomme la nouvelle image pour l'enregistrer
imageModif.show()#on visulaise cette nouvelle image
```