TRAITEMENT D’UNE IMAGE

1 PIXEL

Une image couleur est composée d’un certain nombre de PIXEL,

Chaque pixel est un petit rectangle de votre écran.

Sa position est identifiée par un numéro de colonne et un numéro de ligne

Par exemple dans une image 1920 par 1627, il y aura 1920 colonnes numérotées de 0 à 1919 et 1627 lignes numérotées de 0 à 1626

Le pixel (325,428) se trouve sur la colonne numéro 325 et sur la ligne numéro 428

Sa couleur est identifiée grâce au mode ‘RGB’ (red, green, blue), par un triplet de nombre indiquant l’intensité de rouge, vert et bleu. Chaque intensité est un nombre entier allant de 0 à 255

Par exemple le pixel (325,428) peut avoir comme code couleur

(0,255,0) cela voudrait dire que ce pixel est vert pur

Un pixel qui a pour code couleur (255,255,255) est donc

……

Un pixel qui a pour code couleur (0,0,0) est donc

……

Un pixel qui a pour code couleur (255,0,255) est donc

……

OUVRIR des IMAGES

Aller sur le site

Python 3 - Repl.it
https://repl.it › Python-3

Vous pouvez à cette occasion vous créer un compte en ligne à l’aide de votre adresse email.

Dans l’éditeur de programme taper l’instruction

from PIL import Image, puis cliquer sur run

Quand on tente l’execution on se rend compte qu’il ne sait pas de quoi on parle.

Il nous faut cliquer à gauche sur Packages, et à la ligne « search for package » on tape pil, et dans la liste on sélectionne pil-compat

Puis on clique sur le +

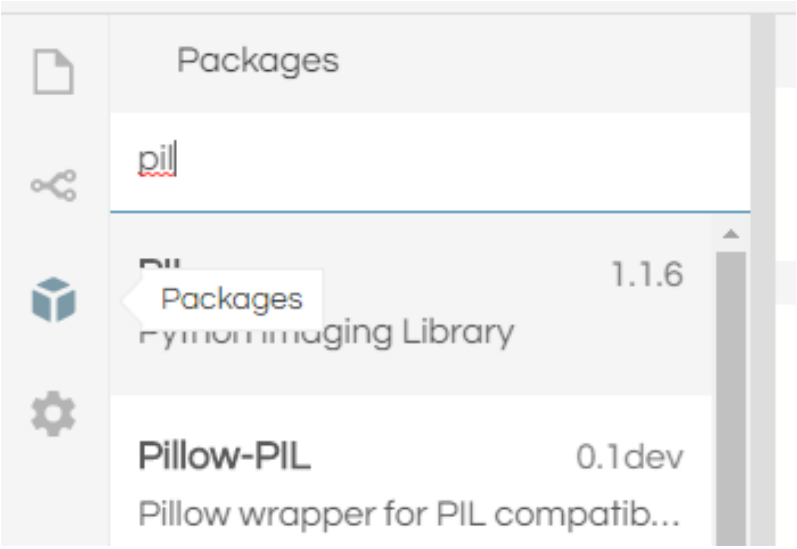
Pour que la librairie soit

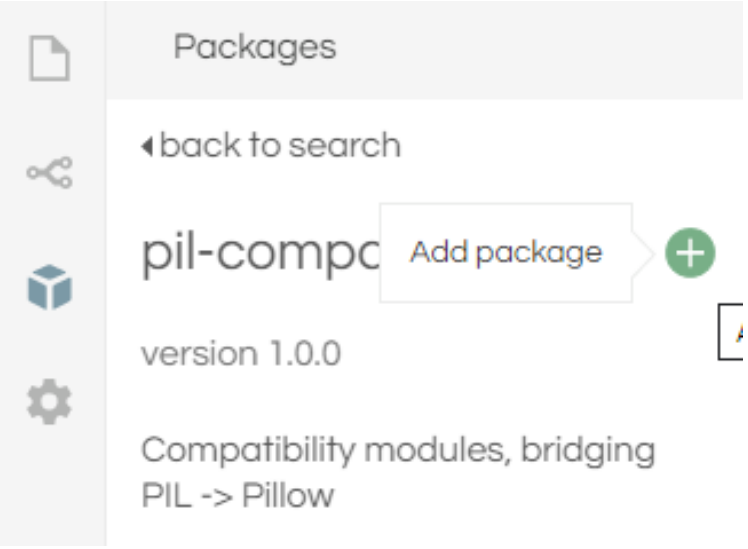
Installée sur le serveur

à partir de là on pourra

l’utiliser sur dans notre

programme.

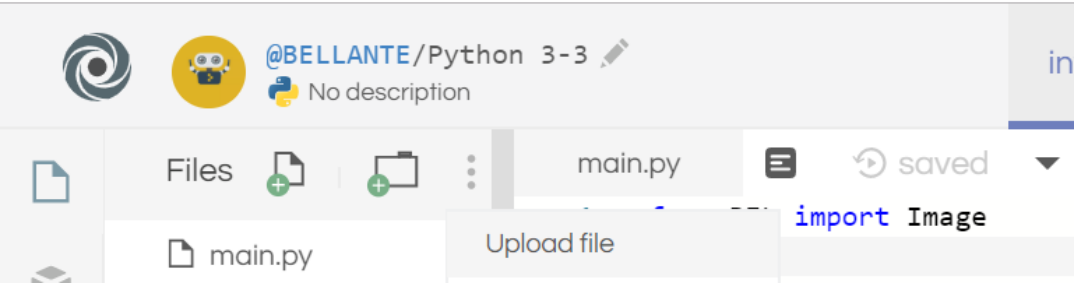




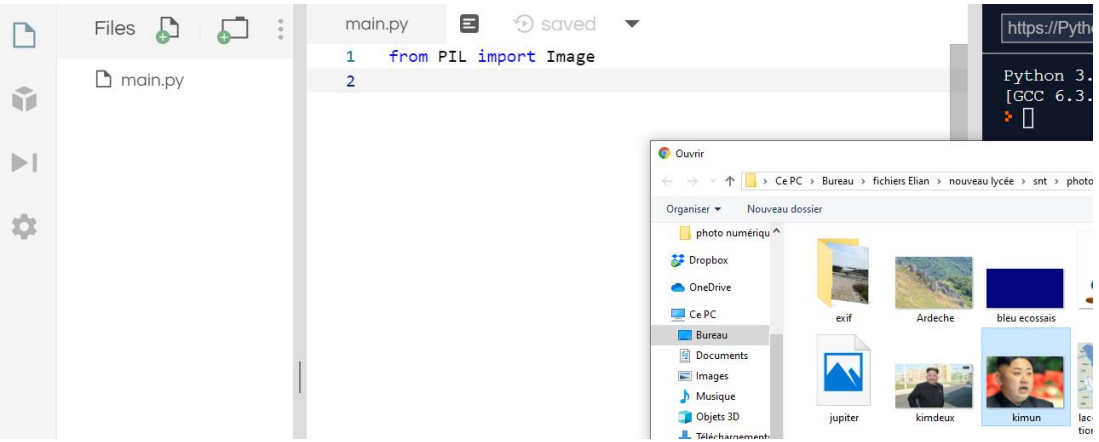
ETAPE 1

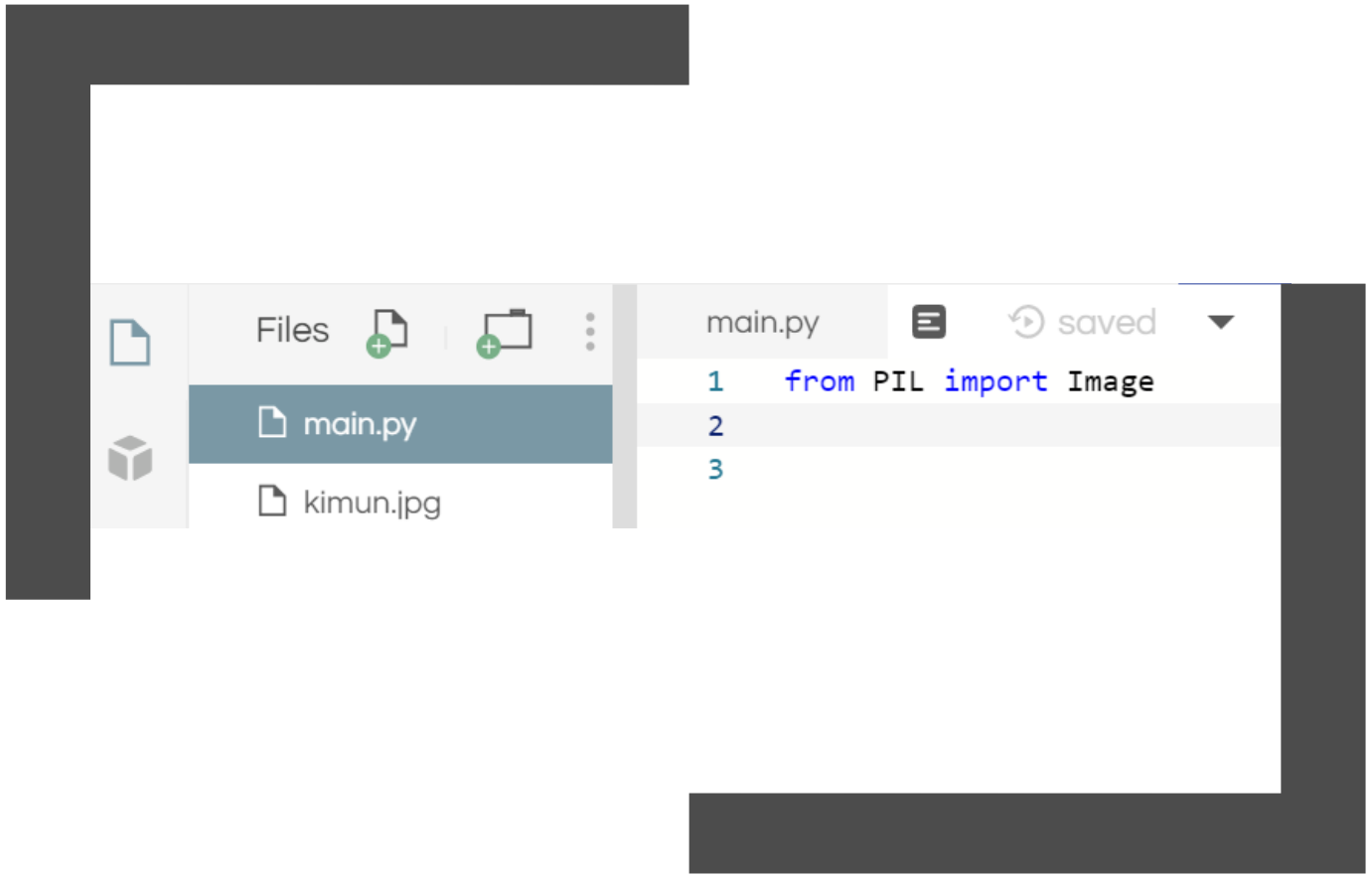
Tout d’abord il faut transférer l’image à ouvrir sur le serveur de repl.it

Cliquer sur Upload file



Sélectionner le fichier et cliquer sur ouvrir, par exemple dingokim.jpg

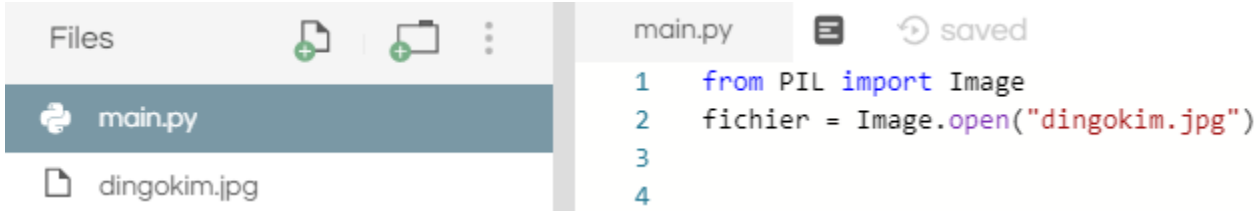




Dans l’onglet file est créé le fichier image

ETAPE 2 Ouvrir une image

Pour cela, dans l'éditeur de programme taper Image.open(nom du fichier entre guillemets)



Le fait d'avoir ouvert le fichier n'ouvre pas l'image, cela permet seulement d'effectuer des modification, ou afficher des paramètres de ce fichier.

Pour ouvrir l'image, il suffit de taper dans la colonne Files.

Si on tape print(fichier)

On obtient alors

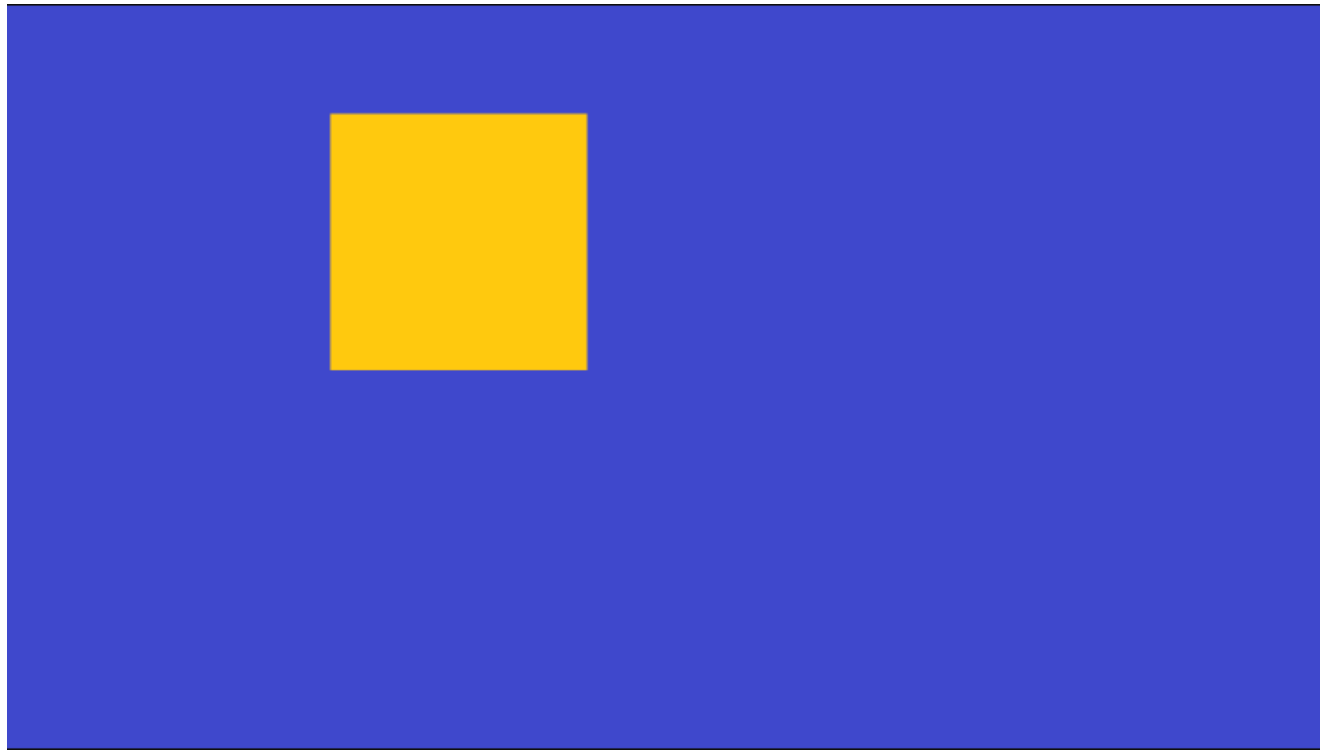
```
1   from PIL import Image
2   fichier = Image.open("dingokim.jpg")
3   print fichier
4
```

```
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=5
00x500 at 0x7F86AFBEA090>
>
```

Ce qui donne le mode de lecture de l’image « RGB », puis le nombre de pixels qui est de 500 par 500 cela veut dire qu’il y a 500 colonnes et autant de lignes.

EXERCICE 1 : retrouver le code couleur

1/ télécharger le fichier image, le fichier se nomme «mpsun.png»



On tape print(fichier.getpixel((1000,1000))

L'ordinateur répond après traitement (63,72,204)

2/ Expliquer ce que veut dire concrètement le triplet (63,72,204)

3/ Trouver le code couleur de la partie orange

Exercice 2 : boucle pour

On a tapé les commandes suivante, et on a obtenu en retour l'image reproduite dans la page suivante.

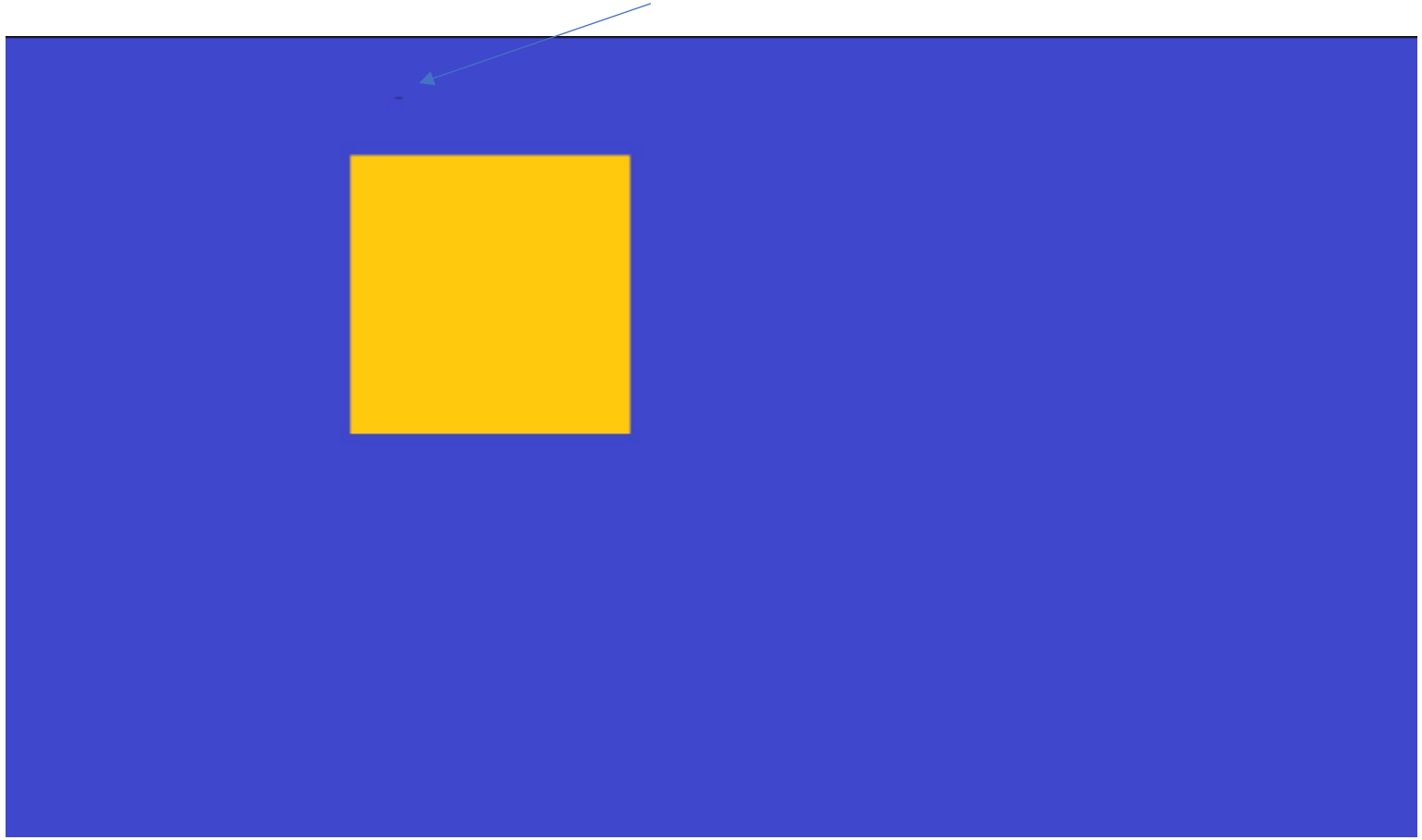
Sous la copie d'écran vous avez le code, que vous pourriez copier et coller dans repl.it (non demandé)

main.py saved

```
1   from PIL import Image
2   fichier=Image.open('mpsun.png')
3   fichier.putpixel((536,81),(0,0,0))
4   fichier.putpixel((537,81),(0,0,0))
5   fichier.putpixel((538,81),(0,0,0))
6   fichier.putpixel((539,81),(0,0,0))
7   fichier.putpixel((540,81),(0,0,0))
8   fichier.putpixel((541,81),(0,0,0))
9   fichier.putpixel((542,81),(0,0,0))
10  fichier.putpixel((543,81),(0,0,0))
11  fichier.putpixel((544,81),(0,0,0))
12  fichier.save('pixel.jpg')
```

```
from PIL import Image
fichier= Image.open("mpsun.png")
fichier.putpixel((536,81),(0,0,0))
fichier.putpixel((537,81),(0,0,0))
fichier.putpixel((538,81),(0,0,0))
fichier.putpixel((539,81),(0,0,0))
fichier.putpixel((540,81),(0,0,0))
fichier.putpixel((541,81),(0,0,0))
fichier.putpixel((542,81),(0,0,0))
fichier.putpixel((543,81),(0,0,0))
fichier.putpixel((544,81),(0,0,0))
fichier.save("pixel.jpg")
```

On obtient , ce petit trait



Expliquer comment fonctionne la commande putpixel, et la commande save

On veut colorier toute la ligne 100 en rouge

1/Quelles sont les « coordonnées »de pixels à colorier?

2/ compléter l'algorithme ci contre,

pour colorier cette ligne 100 en rouge, et pour sauver la nouvelle image sous le nom de « mpsrouge.jpg »

```
from PIL import Image
fichier=Image.open('mpsun.png')
for k in range(    ):
  fichier.putpixel(    ,    )
fichier.save(    )
```

Copier et coller les lignes suivantes sur repl.it pour vérifier votre proposition

```
from PIL import Image
fichier= Image.open("mpsun.png")
for i in range(....) :
fichier.putpixel(..... , .........)
fichier.save(.....)
```

A VOUS de jouer

1) Toujours sur l'image « mpsun », colorier en vert tous les pixels se trouvant sur la 654ème colonne

2) Colorier en vert tous les pixels se trouvant entre la 654ème colonne et la 700ème

Exercice 3 : boucle tant que

On a colorié en noir une ligne sur l’image ci-dessous, on souhaite déterminer le numéro de la ligne que l’on a colorié

Vous trouverez cette image dans votre répertoire sous le nom :
« mpsnoire.png »

Après avoir chargé l’image, copier, coller et compléter le programme suivant pour qu’il réponde à l’objectif

```
from PIL import Image
fichier= Image.open("mpsnoire.png")
k=0
while fichier.getpixel(...) ... (...) :
k=...
print(...)
```

Compléter l’algorithme ci-dessus pour qu’il trouve le numéro de ligne cherché.

Pour tester si une égalité est vrai il faut taper ……==……

Pour tester si une égalité n’est pas vrai il faut taper …….!=…….

EXERCICE 4 : création d’image

1.2 Créer une nouvelle image

On veut créer un le drapeau français dans une image 900 par 900

Si on tape

```
10 nouvelleimage=Image.new()
11
```

```
Arguments
new(mode, size, color=0)
```

La commande Image.new va donc créer un nouveau fichier image python. Pour cela il faut remplir trois parties, mode,size, color

mode : attention c’est un texte on sait que le mode est le mode RGB

size: attention c’est un couple

color: il faut mettre 0

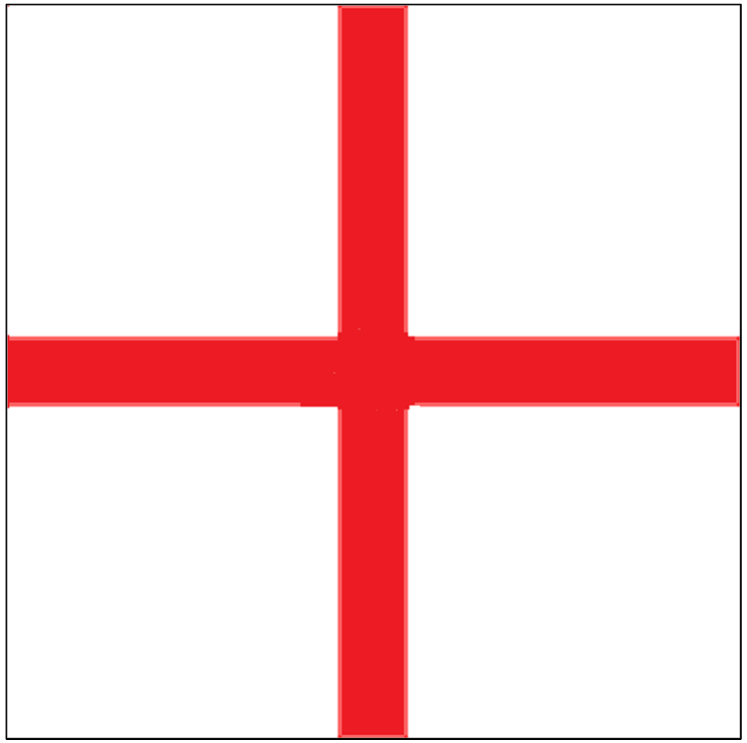
Proposer un script permettant de fabriquer le drapeau français, pour cela compléter l'algorithme suivant

```
1   from PIL import Image
2   from numpy import *
3   fichier=Image.new('RGB',(900,900),0)
4   for k in range(300):
5     for j in range(900):
6       fichier.putpixel(      ,          )
7   for k in arange(301,600,1):
8     for j in range(   ):
9       fichier.putpixel(      ,          )
10  for k in arange(          ):
11    for j in range(   ):
12      fichier.putpixel(      ,          )
13  fichier.save(          )
```

Le drapeau de la Hongrie est le suivant

Proposer un algorithme permettant de construire l’image de ce drapeau sous python

Même chose avec le drapeau de l’Angleterre



EXERCICE 5 :Equation de droits interessant pour tracer des lignes droites

Exercice 5 équations de droites

Lorsqu'une image est créée, sa taille par exemple (600,300) nous permet de définir un repère orthonormé,

Chaque pixel de l'image est défini par son numéro de colonne i et son numéro de ligne j, ainsi cela définira un point de coordonnées (i ; j)

On souhaite construire le drapeau de l'Ecosse

On donc l'origine du repère est le point suivant

L'axe des abscisses est situé donc ici

L'axe des ordonnées est situé donc ici

Pour ce drapeau, on va choisir une image (600,300)

Le point K a pour coordonnées (0;0)

Le point H a pour coordonnées (0;49)

Le point L a pour coordonnées (599,0)

Déterminer les coordonnées de tous les autres points en respectant la symétrie du drapeau

Déterminer les équations des droites (GJ) et (HI)

Déterminer les équations des droites (CE) et (DF)

Rappel pour les équations des droites

Si on a $A(x_A; y_A)$ et $B(x_B; y_B)$, si les abscisses de A et BV ne sont pas égales alors une équation de la droite (AB) est donnée par

$$y = ax + b$$

Avec

$$a = \frac{y_B - y_A}{x_B - x_A}$$

ET

$$b = y_A - a \times x_A$$

L’objectif est donc de compléter l’algorithme suivant

```
8  from math import *
9  from PIL import *
10 from PIL import Image
11 def a(A,B):
12     num=float(B[1]-A[1])
13     den=float(B[0]-A[0])
14     return num/den
15 def ______:
16     return ____________________
17 ecosse=Image.new( ______________ )
18 #droite (GJ)
19 ___=[___ ,___]
20 ___ =[____, ____]
21 #droite (HI)
22 ____________
23 ____________
24 #droite (DF)
25 ____________
26 ____________
27 #droite (CE)
28 ____________
29 ____________
30 for x in range(600):
31     for y in range(300):
32         ecosse.putpixel( ____________ )
33 for i in range(600):
34     for j in range(300):
35         if ____________ ___ ____________ :
36             ecosse.putpixel( ____________ )
37 for i in range(600):
38     for j in range(300):
39         if ____________ and ____________ :
40             ecosse.putpixel( ____________ )
   ecosse.show()
```

ETAPE 1 lignes 10; 11; 12 et 13

Construction d'une fonction permettant de calculer le coefficient directeur d'une droite donnée par deux points

On rappelle que

$$a = \frac{y_B - y_A}{x_B - x_A}$$

On va pour cela créer une fonction

Cette fonction comporte des erreurs, rectifiez les en sachant que

A[0] renvoie l'abscisse du point A

A[1] renvoie l'ordonnée du point A

La fonction float écrit le calcul sous forme décimale

```
8  from math import*
9  from PIL import *
10 def a(A,B):
11     n=float(B[1]-B[0])
12     d=float(A[1]-A[0])
13     return n/d
```

Etape 2 lignes 15 et 16:

Déterminer une fonction donnant l'ordonnée à l'origine d'une droite (AB) à partir des points A et B

On rappelle que

$$b = y_A - a \times x_A$$

On rappelle que le coefficient directeur

(a) de la tangente a été donné par la fonction précédente, et que l'abscisse de A est donnée par A[0], et que son ordonnée est donnée par A[1]

ETAPE 3 de la ligne 18 à 29 : rentrez les coordonnées des différents points

Par exemple pour définir le point K, il suffirait de rentrer

K=[0,0] penser aux crochets et à la virgule de séparation

ETAPE 4 lignes 30;31 et 32:

Colorier tout le drapeau de la couleur du bleu Ecosse

Pour cela on utilise une boucle « pour »

On rappelle que le bleu Ecosse a pour code couleur (5,5,125)

```
30 for x in range(600):
31     for y in range(300):
32         ecosse.putpixel(                )
```

ETAPE 5 lignes 33,34,35,36 : première bande blanche

Il suffit de colorier l'espace entre les deux droites (GJ) et (HI)

On rappelle que pour notre programme

La droite (GJ) a pour équation

$$y = a(G,J)x + b(G,J)$$

La droite (HI) a pour équation

$$y = a(H,I)x + b(H,I)$$

On se donne un point de coordonnées $(i;j)$ situé entre les droites d'équation remplir alors en utilisant le bon symbole parmi

$$=; \geq; \leq$$

$$\boldsymbol{a(G,J) \times i + b(G,J) \ldots j}$$

De la même manière déterminer une relation entre les réels

$a(H,I)$ **;** $b(H,I)$; i et j

Compléter alors les lignes 33;34;35;36

```
33 for i in range(600):
34     for j in range(300):
35         if                    and                    :
36             ecosse.putpixel(                    )
```

ETAPE 6 lignes 37;38;39 et 40 : colorier en blanc l'autre bande

On rappelle que l'espace à colorier est maintenant délimité par les droites (DF) et (CE) Il suffit pour cela d'adapter l'étape 5 à ces deux droites

```
37 for i in range(600):
38     for j in range(300):
39         if                    and                    :
40
```

Exercice 6 : calcul d’aires à partir d’images (facultatif pour votre culture)

calculer des aires

on se donne l'image ci –contre,
elle est dans le fichier image3

On souhaite déterminer l'aire de cette surface en km², en sachant que la largeur totale de cette image représente 100km en réalité

Premier travail

Récupérer le code couleur du bleu de ce lac

Deuxième travail

On va compter le nombre de pixels dont le code couleur vaut celui du lac. On pourra introduire une variable s qui chaque fois que le code couleur du pixel regardé vaut celui du lac, augmente d’une unité

Troisième travail

Déterminer la fraction des pixels dont le code couleur vaut celui de la couleur du lac

Dernier travail

A l’aide d’une règle de trois conclure

Exercice 8: calcul de l’évolution de l’aire du Lac Tchad (facultatif pour votre culture)

Sur l'image « lac Tchad » on a la représentation de son évolution

ON souhaite quantifier cette évolution

En sachant qu'il y a 4 cartes de même

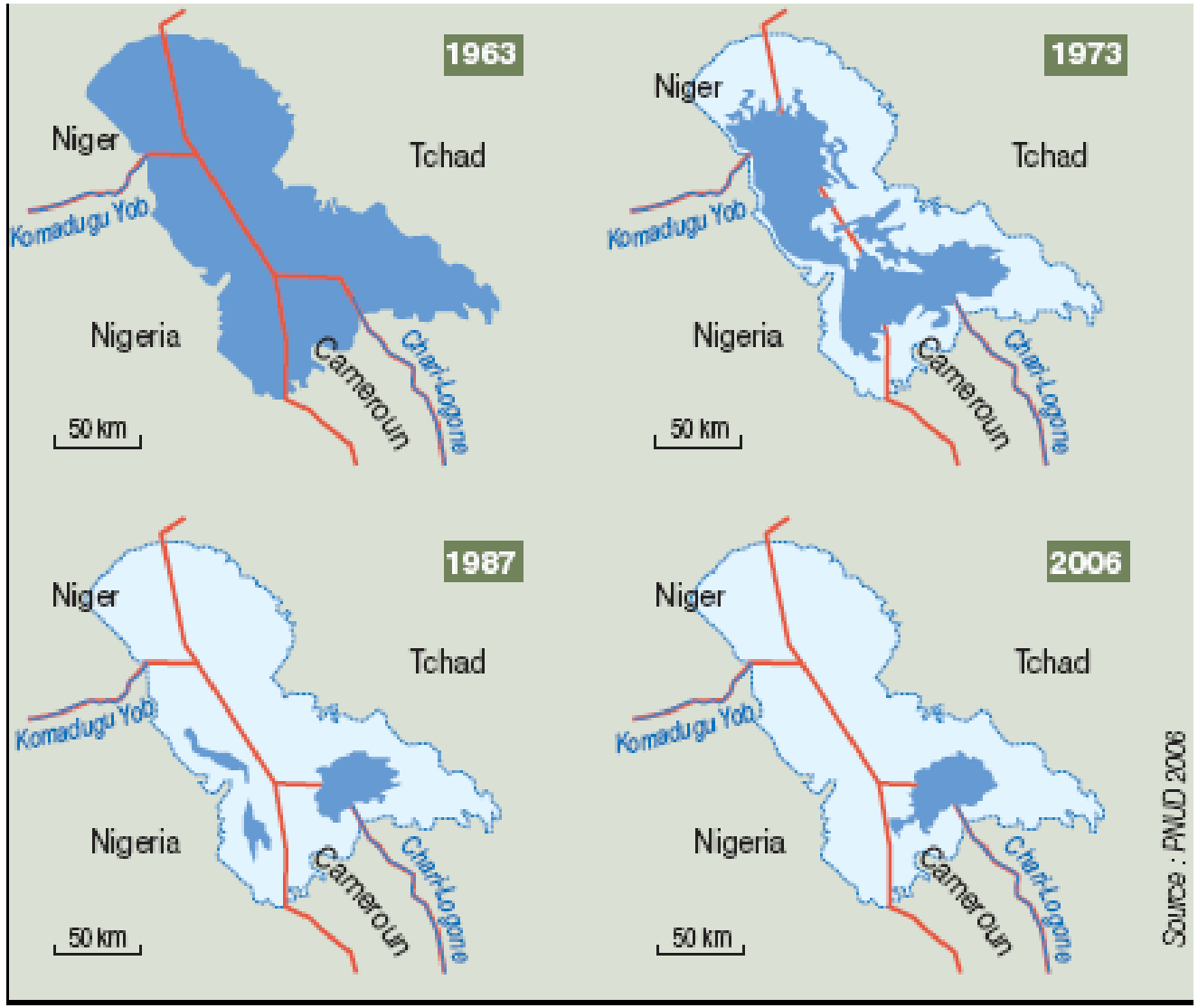
taille et de même échelle sur cette

image , on souhaite déterminer le

pourcentage de baisse entre deux

prises de vue.

Il y aura donc trois pourcentages à donner



Indications

1/récupérer toutes les informations de l’image

2/ bien regarder les pixels qui délimitent chaque photo

3/récupérer le code couleur du lac

4/calculer pour chaque partie l’aire occupée par le lac

Exercice 9 : filtrer suivant une couleur

Choisir une image de départ, filtrez là suivant du vert comme par exemple

Indication : pour chaque pixel de l'image de départ, on ne doit garder que la composante indiquant le niveau de vert

A vous de jouer: choisir une couleur et filtrer suivant cette couleur

Aide pour les codes de rouges, puis vert, puis bleus

Pour un pixel si on pose p=getpixel(32,64)

le fait de taper p[0] donnera le niveau de rouge pour le pixel (32,64)

le fait de taper p[1] donnera le niveau de vert pour le pixel (32,64)

le fait de taper p[2] donnera le niveau de bleu pour le pixel (32,64)

On se donne la photo suivante fleur.jpg, filtrez la selon la couleur de votre choix

Exercice 10 : superposer deux images

Superposer des images

Choisir deux images la première est dans le fichier « reine des neiges », et la deuxième est dans le fichier « trump » par exemple ces deux personnages (pas de politique!!) On veut superposer ces deux images et en créer une seule

On doit obtenir

Indications

D’abord ouvrir les fichiers contenant les images, ils se nomment « reines des neiges » et « trump »

Récupérer les tailles des deux images

Puis créer une nouvelle image que l’on nommera par exemple fin,e tenant compte des tailles précédentes

Utiliser les commandes getpixel pour se donner les codes couleur de chaque pixel des deux fichiers

Ensuite on peut utiliser la commande max, comme suit

```
fin.putpixel((k,j),(max(p[0],q[0]),max(p[1],q[1]),max(p[2],q[2])))
```

Le pixel (k,j) de l’image fin sera colorié avec le code couleur rouge, le plus grand entre celui de « trump » et de « reine des neiges », car on a tapé (max(p[0],q[0]) , idem pour les codes du vert et du bleu

3 des fonctions en algorithmique

Exercice 11 : création de fonctions et contours d’images

On veut créer les contours d’une personne, d’un objet

Le contour d’une personne sur image peut être défini à partir d’une différence entre les codes couleur de deux pixels

En effet s’il y a un changement brusque entre deux pixels voisins, alors ce changement de couleur peut définir une partie du contour de la personne

Sur cette image les pixels voisins de ce pixel

seront pour ceux situés à gauche, très différents

de ceux situés à droite

Cette différence, permet de dire que ce pixel est

un pixel du contour de dingo

L’image de dingo précédente a pour taille (500,500)

```
8 from PIL import *
9 from PIL import Image
10 from numpy import*
11 dingo=Image.open('C:\\Python27\\Lib\\dingokim.jpg','r')
12 contour=Image.new('RGB',(500,500),0)
13 def c(k,j):
14     return dingo.getpixel((k,j))
```

Que permettent de réaliser les lignes 13 et 14?

Si on voulait le pixel (432,356) que faudrait il taper?

Pour un pixel (k,j) on définit les pixels voisins comme suit

nordouest (no)	nord(n)	nordest(ne)
ouest (o)	(k,j)	est (e)
sudoouest(so)	sud (s)	sudest(se)

Définir les différentes fonctions permettant de déterminer les codes couleurs des 8 voisins du pixel (k,j)

Pour cela remplir l'algorithme

```
8  from PIL import *
9  from PIL import Image
10 from numpy import*
11 dingo=Image.open('C:\\Python27\\Lib\\dingokim.jpg','r')
12 contour=Image.new('RGB',(500,500),0)
13 def c(k,j):
14     return dingo.getpixel((k,j))
15 def no(k,j):
16     return
17 def ne(k,j):
18     return
19 def so(k,j):
20     return
21
22
23
24
```

poursuivre pour définir les autres fonctions

A la suite copier les deux lignes suivantes

```
def situ(k,j):
    return [no(k,j),ne(k,j),so(k,j),se(k,j),n(k,j),s(k,j),o(k,j),e(k,j)]
```

Expliquer ce que permet de faire cette fonction.

Toujours à la suite copier les deux lignes suivantes

```
def R(k,j):
    A=[]
    for t in situ(k,j):
        A=A+[abs(t[0]-c(k,j)[0])]
    return max(A)
```

Que permet de faire cette fonction?

Toujours à la suite copier et compléter, pour créer

La fonction permettant de retourner la plus grande différence de vert entre les pixels voisins de (k,j)

ET

La fonction permettant de retourner la plus grande différence de vert entre les pixels voisins de (k,j)

```
38 def G(k,j):
39     A=[]
40     for t in situ(k,j):
41         [redacted]
42     return max( )
43 def B([redacted]):
44     A=[]
45     for t in [redacted]:
46         [redacted]
47     return [redacted]
```

Toujours à la suite, créer la fonction Z permettant de retourner pour (k,j) rentré , la plus grande valeur entre R(k,j), G(k,j) et B(k,j)

```
48 def Z(    ):
49     return
```

Enfin, compléter les lignes suivantes pour qu'à chaque pixel, (k,j) si on a Z(k,j) >50 alors le pixel de la nouvelle image nommée contour doit être colorié en noir. Sinon ce pixel doit être colorié en blanc

```
50 for k in                :
51     for j in                :
52         if                :
53             contour.putpixel((k,j),        )
54         else:
55
   contour.show()
```

Exercice 13 d’une image couleur à une image noir et blanc

On rappelle que le code couleur en mode RGB de chaque pixel est la donnée d'un triplet (….;….;….) où chaque composante est un entier entre 0 et 255

ON souhaite transformer en noir et blanc l'image suivante

un pixel dont le code couleur est du type (A,A,A) où A est un entier est de couleur grise.

Comment rendre gris un pixel dont le code couleur est $(124,13,49)$?

En déduire un algorithme permettant de transformer en noir et blanc cette photo

(makikata.jpg)

Exercice 14 : foncer ou éclaircir une photo en noir et blanc

On rappelle que plus le code RGB des pixels est proche du code (255,255,255), plus la couleur est proche du blanc

Et, plus les pixels le code RGB est proche du code (0,0,0), plus la couleur est proche du noir

L'objectif est de construire une fonction permettant de noircir, d'un certain pourcentage une photo

Un exemple

On se donne le pixel dont le code RGB est (150,150,150)

ON souhaite le noircir de 40%, c'est-à-dire diminuer de 40% l'écart qui le sépare du noir total. Ainsi on veut donc un pixel n'ayant plus que 60% de couleur initiale

ON multiplie donc 150 par $\frac{60}{100}$

Comment faire pour le noircir de 24,7%?

Construire alors une fonction permettant de noircir une photo en noir et blanc, de n’importe quel pourcentage

L’objectif est de construire une fonction permettant d’éclaircir, d’un certain pourcentage une photo

Un exemple

On se donne le pixel dont le code RGB est (150,150,150)

ON souhaite l’éclaircir de 40%, c’est-à-dire diminuer de 40% l’écart qui le sépare du blanc total.

Il y a $255 - 150 = 155$ d’écart avec le blanc total

On prend les 40% de cet écart ce qui donne 62

On ajoute alors 62 à 150, le pixel aura un code RGB de $(212,212,212)$

Comment faire pour éclaircir de 24,7%?

Construire alors une fonction permettant d’éclaircir une photo en noir et blanc, de n’importe quel pourcentage

Eclaircir de 13,54% la photo « makinoir.jpg »

Exercice 15 : niveau de gris et fonction de retouche

Une image en mode RGB, code chaque pixel suivant un niveau de rouge, un niveau de vert et un niveau de bleu. Chaque niveau est un entier entre 0 et 255, ainsi un rouge sera codé par (255,0,0)

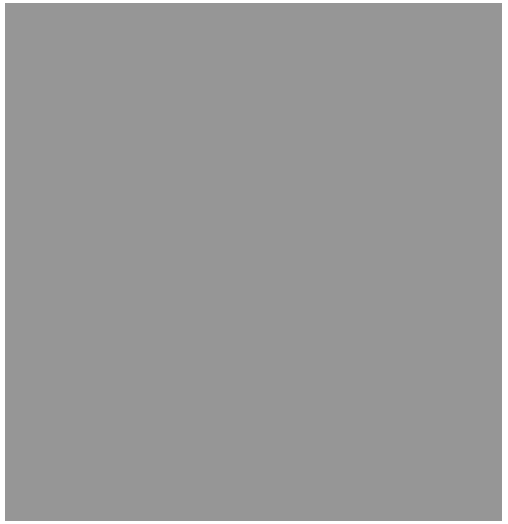
- 1/ Comment coder du blanc ?du noir ?
- 2/ Un pixel a un code couleur égal à (128,128,128) quelle sera sa couleur ?

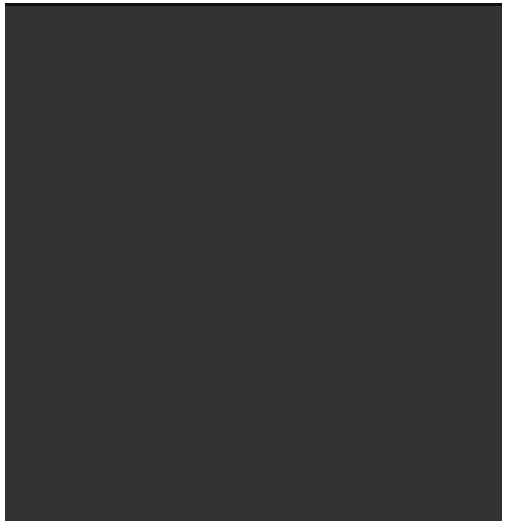
On code le niveau de gris sur une échelle de 0 à 1 (de noir à blanc)

- 3/Quel calcul effectuer sur le pixel (128,128,128) pour obtenir son niveau de gris ?

1/A partir des moyennes coefficientées, proposer une autre façon de mettre en noir et blanc une photo

On donne dans les fichiers « carrés1 »; « carrés2 »; « carrés3 » et « carrés4 » 4 images carrés, déterminer le niveau de gris de chacune de ces images





On souhaite construire une fonction permettant d'éclaircir ou de foncer une photographie en noir et blanc

Une fonction de retouche est une fonction f strictement croissante sur l'intervalle [0 ;1], telle que $f(0) = 0$ et $f(1) = 1$.

- 1/ La fonction définie sur $[0; 1]$ par $f(x) = x + 1$ est elle une fonction de retouche ?
- 2/ Prouver que les fonctions f, g et h définies par $f(x) = \sqrt{x}$;

$g(x) = x^2 et\ h(x) = x^3$ sont des fonctions de retouche.

- 3/ La fonction g éclaircit elle ou fonce t-elle l'image ?

On donne une image de départ, pour chaque image 1 et 2 on a appliqué une fonction de retouche sur l'image de départ

Déterminer pour chacune des images 1 et 2 la fonction de retouche qu'on a appliqué à l'image de départ 'makikatanetb.jpg'

Chaque fonction de retouche est parmi les fonctions $f, g\ et\ h$

Image de départ

image 1

image 2

- 1/Ci-dessous, on donne un algorithme, déterminer ce qu'il renvoie
- 2/adapter cet algorithme pour qu'il permette de retoucher l'image créé à l'aide d'une fonction de retouche de votre choix
- 3/choisir une image en noir et blanc et effectuer une retouche à l'aide d'une fonction de retouche

```
7  from PIL import *
8  from PIL import Image
9  from math import *
10 from numpy import*
11 carre=Image.new('RGB',(50,50),0)
12 x=input("niveau de gris entre 0 et 255")
13 print(x)
14 for k in range(50):
15     for j in range(50):
16         t=(x,x,x)
17         carre.putpixel((k,j),t)
18 carre.show()
```

Exercice 16 :utiliser l’aléatoire

Un premier exemple

On se donne l'algorithme suivant

```
from PIL import Image
from random import randint
fleur=Image.open('fleur.jpg')
fleur1=Image.new('RGB',(549,454),0)
for k in range(549):
    for j in range(454):
        (x,y,z)=fleur.getpixel((k,j))
        x=randint(0,x)
        y=randint(0,y)
        z=randint(0,z)
        fleur1.putpixel((k,j),(x,y,z))
fleur1.save('fleur1.jpg')
```

1/ Expliquer ce que fait en détails l’algorithme précédent

Dans on a foncé de manière aléatoire chaque pixel de la photo « fleur.jpg »

2/ proposer un algorithme permettant d’éclaircir aléatoirement chaque pixel de la photo

4/ compléter la fonction suivante afin qu'elle puisse changer l'ordre des composantes d'un triplet, puis appliquer cette fonction à chaque pixel de la photo de départ

```
def ech(L):
    L=[redacted](L)
    M=[]
    x=randint(0,2)
    M=[redacted]
    del [redacted]
    y=randint([redacted])
    M=[redacted]
    del [redacted]
    M=[redacted]
    M=[redacted](M)
    return [redacted]
```

```
> L=(12,564,78)
> L=list(L)
> print(L)
[12, 564, 78]
> M=[124,547,65]
> M=tuple(M)
> print(M)
(124, 547, 65)
```

Aide :

On peut modifier le type d’un objet, par exemple passer d’un tuple par exemple L=(12,564,78) à une liste L=[12,564,78]

Et vice versa, passer d’une liste M=[124,547,65] à un tuple M=(124,547,65)

Exercice 17 :les métadonnées d’une photo

Les métadonnées d’une photo

Y avez-vous pensé?

Lorsque vous publiez une photo, sur le web, vous laissez des informations. Lesquelles?

Un exemple: On a pris la photo suivante,

Pourriez vous savoir

quand et où elle a été prise?

avec quel appareil?

Comment savoir tout cela?

On peut traduire tout cela avec le code suivant, que vous trouverez dans le fichier exif.py

```python
from PIL.ExifTags import TAGS
from PIL import Image
from PIL.ExifTags import GPSTAGS
def get_exif(filename):
    image = Image.open(filename)
    image.verify()
    return image._getexif()
exif = get_exif('photo.jpg')
print(exif)
def get_labeled_exif(exif):
    labeled = {}
    for (key, val) in exif.items():
        labeled[TAGS.get(key)] = val
    return labeled
exif = get_exif('photo.jpg')
labeled = get_labeled_exif(exif)
print(labeled)
def get_geotagging(exif):
    if not exif:
        raise ValueError("No EXIF metadata found")
    geotagging = {}
    for (idx, tag) in TAGS.items():
        if tag == 'GPSInfo':
            if idx not in exif:
                raise ValueError("No EXIF geotagging found")
            for (key, val) in GPSTAGS.items():
                if key in exif[idx]:
                    geotagging[val] = exif[idx]
```

```python
exif = get_exif('photo.jpg')
geotags = get_geotagging(exif)
print(geotags)
def get_decimal_from_dms(dms, ref):
    degrees = dms[0][0] / dms[0][1]
    minutes = dms[1][0] / dms[1][1] / 60.0
    seconds = dms[2][0] / dms[2][1] / 3600.0
    if ref in ['S', 'W']:
        degrees = -degrees
        minutes = -minutes
        seconds = -seconds
    return round(degrees + minutes + seconds, 5)
def get_coordinates(geotags):
    lat = get_decimal_from_dms(geotags
    ['GPSLatitude'], geotags['GPSLatitudeRef'])
    lon = get_decimal_from_dms(geotags
    ['GPSLongitude'], geotags['GPSLongitudeRef'])
    return (lat,lon)
exif = get_exif('photo.jpg')
geotags = get_geotagging(exif)
print(get_coordinates(geotags))
```

En réponse on obtient

```
Python 3.7.4 (default, Jul  9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
{36864: b'0210', 37121: b'\x01\x02\x03\x00', 37377
: (298973, 10000), 36867: '2018:08:28 14:39:42', 3
6868: '2018:08:28 14:39:42', 37378: (227, 100), 37
379: (0, 1), 37380: (0, 10), 37383: 5, 37384: 1, 3
7385: 24, 37386: (3830, 1000), 40961: 1, 40962: 39
68, 40965: 8432, 41990: 0, 37520: '853035', 37521:
 '853035', 37522: '853035', 40963: 2976, 41996: 0,
 257: 2976, 258: (8, 8, 8), 41495: 2, 271: 'HUAWEI
', 41728: b'\x03', 33434: (529000, 1000000000), 28
2: (72, 1), 531: 1, 33437: (220, 100), 41729: b'\x
01', 283: (72, 1), 34850: 2, 34853: {0: b'\x02\x02
\x00\x00', 1: 'N', 2: ((47, 1), (37, 1), (29107360
, 1000000)), 3: 'W', 4: ((3, 1), (25, 1), (4297657
0, 1000000)), 5: b'\x00', 6: (4766, 100), 7: ((12,
 1), (39, 1), (42, 1)), 27: 'GPS', 29: '2018:08:28
'}, 41985: 1, 34855: 50, 296: 2, 41986: 0, 40960:
b'0100', 256: 3968, 41987: 0, 305: 'PRA-LX1 8.0.0.
370(C432)', 306: '2018:08:28 14:39:42', 41988: (10
0, 100), 41989: 26, 41991: 0, 41992: 0, 41993: 0,
270: 'dav', 41994: 0, 41995: b'ipp\x00', 272: 'PRA
-LX1', 274: 0, 34665: 288, 37500: b'Auto\x00'}
{'ExifVersion': b'0210', 'ComponentsConfiguration'
: b'\x01\x02\x03\x00', 'ShutterSpeedValue': (29897
3, 10000), 'DateTimeOriginal': '2018:08:28 14:39:4
2', 'DateTimeDigitized': '2018:08:28 14:39:42', 'A
pertureValue': (227, 100), 'BrightnessValue': (0,
1), 'ExposureBiasValue': (0, 10), 'MeteringMode':
5, 'LightSource': 1, 'Flash': 24, 'FocalLength': (
3830, 1000), 'ColorSpace': 1, 'ExifImageWidth': 39
68, 'ExifInteroperabilityOffset': 8432, 'SceneCapt
ureType': 0, 'SubsecTime': '853035', 'SubsecTimeOr
```

```
68, 'ExifInteroperabilityOffset': 8432, 'SceneCapt
ureType': 0, 'SubsecTime': '853035', 'SubsecTimeOr
iginal': '853035', 'SubsecTimeDigitized': '853035'
, 'ExifImageHeight': 2976, 'SubjectDistanceRange':
 0, 'ImageLength': 2976, 'BitsPerSample': (8, 8, 8
), 'SensingMethod': 2, 'Make': 'HUAWEI', 'FileSour
ce': b'\x03', 'ExposureTime': (529000, 1000000000)
, 'XResolution': (72, 1), 'YCbCrPositioning': 1, '
FNumber': (220, 100), 'SceneType': b'\x01', 'YReso
lution': (72, 1), 'ExposureProgram': 2, 'GPSInfo':
 {0: b'\x02\x02\x00\x00', 1: 'N', 2: ((47, 1), (37
, 1), (29107360, 1000000)), 3: 'W', 4: ((3, 1), (2
5, 1), (42976570, 1000000)), 5: b'\x00', 6: (4766,
 100), 7: ((12, 1), (39, 1), (42, 1)), 27: 'GPS',
29: '2018:08:28'}, 'CustomRendered': 1, 'ISOSpeedR
atings': 50, 'ResolutionUnit': 2, 'ExposureMode':
0, 'FlashPixVersion': b'0100', 'ImageWidth': 3968,
 'WhiteBalance': 0, 'Software': 'PRA-LX1 8.0.0.370
(C432)', 'DateTime': '2018:08:28 14:39:42', 'Digit
alZoomRatio': (100, 100), 'FocalLengthIn35mmFilm':
 26, 'GainControl': 0, 'Contrast': 0, 'Saturation'
: 0, 'ImageDescription': 'dav', 'Sharpness': 0, 'D
eviceSettingDescription': b'ipp\x00', 'Model': 'PR
A-LX1', 'Orientation': 0, 'ExifOffset': 288, 'Make
rNote': b'Auto\x00'}
{'GPSVersionID': b'\x02\x02\x00\x00', 'GPSLatitude
Ref': 'N', 'GPSLatitude': ((47, 1), (37, 1), (2910
7360, 1000000)), 'GPSLongitudeRef': 'W', 'GPSLongi
tude': ((3, 1), (25, 1), (42976570, 1000000)), 'GP
SAltitudeRef': b'\x00', 'GPSAltitude': (4766, 100)
, 'GPSTimeStamp': ((12, 1), (39, 1), (42, 1)), 'GP
SProcessingMethod': 'GPS', 'GPSDateStamp': '2018:0
8:28'}
```

```
(47.62475, -3.4286)
```

A partir de la réponse précédente,

1. déterminer où cette photo a été prise. Vous pourrez rentrer les coordonnées GPS dans géoportail,

2. Avec quel type d’appareil elle a été prise

3. Quand elle a été prise

Tester cette fonction avec l’une des photos d’un de vos camarades

Voici le lien qui vous éclairera un peu

http://sntflandres.free.fr/?page_id=1539

ET MAINTENANT SOYEZ CREATIFS POUR CONSTRUIRE VOS PROPRES IMAGES