

Filtre médian sur PYTHON

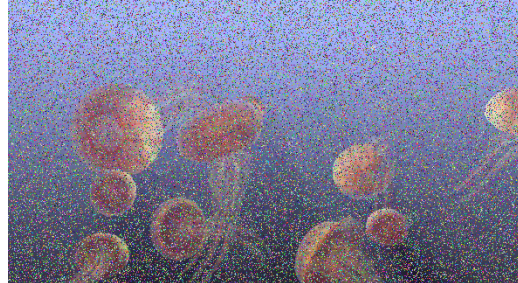
Exercice 1

Compléter le programme suivant, de telle façon qu'il prenne une image et la détériore en lui rajoutant du *bruit*, c'est-à-dire en insérant des pixels mal colorés — par exemple blancs.

AVANT



APRÈS



Structure du programme

```
from random import randint
from PIL import Image

image=Image.open("meduses.png")

longueur,largeur=image.size
image2=Image.new('RGB', (longueur,largeur))

for y in range(largeur):
    for x in range(longueur):
        z=randint(1,100)
        p=image.getpixel((x,y))
        if z<...:# On deteriere 10 pourcents des pixels
            rouge=.....
            vert=.....
            bleu=.....
        else: # les autres pixels sont laisses intacts:
            rouge=.....
            vert=.....
            bleu=.....
        image2.putpixel((x,y), (rouge,vert,bleu))

image2.save("image_bruitee.png")
image2.show()
```

Exercice 2

On considère le programme suivant :

Structure du programme

```
def mystere(suite_de_nombres):
    liste_r=[]
    for c in suite_de_nombres:
        liste_r.append(c)
    liste_r.sort()
    return liste_r[4]
```

1. Que renvoie ce programme si l'on appelle `mystere(6,8,4,3,9,10,20,7,11,24,0)` ?

2. Quel outil mathématique est renvoyé si l'on applique ce programme uniquement sur des suites de 9 nombres ?

Exercice 3

On considère une image image.png en mode RVB fournie ci-dessous :

155	103	169	121	161	116	161	120
117		122		125		133	
160	108	71	211	161	117	160	122
118		230		128		138	
166	111	164	118	161	122	157	125
117		123		135		140	

On considère par ailleurs le bout de code suivant :

Structure du programme

```
from PIL import Image

image=Image.open( "image.png")
longueur,largeur=image.size

rouge = [[0 for y in range(largeur)] for x in range(longueur)]
vert = [[0 for y in range(largeur)] for x in range(longueur)]
bleu = [[0 for y in range(largeur)] for x in range(longueur)]
```

1. Que valent les variables largeur et longueur à l'issue de ce code ?
2. Que valent les listes rouge, vert, bleu à cet instant ?

On considère ensuite le code suivant :

Structure du programme

```
image2=Image.new( 'RGB', (longueur,largeur))

for y in range(largeur):
    for x in range(longueur):
        p=image.getpixel((x,y))
        rouge[x][y] = p[0]
        vert[x][y] = p[1]
        bleu[x][y] = p[2]
```

3. Que valent les listes rouge, vert et bleu à l'issue de ce code ?

On considère le programme voisins ci-dessous :

Structure du programme

```
def voisins(couleur,ligne,colonne):
    liste = []
    c = couleur[ligne-1][colonne-1]
    liste.append(c)
    c = couleur[ligne][colonne-1]
    liste.append(c)
    c = couleur[ligne+1][colonne-1]
    liste.append(c)
    c = couleur[ligne-1][colonne]
    liste.append(c)
    c = couleur[ligne][colonne]
    liste.append(c)
    c = couleur[ligne+1][colonne]
    liste.append(c)
    c = couleur[ligne-1][colonne+1]
    liste.append(c)
    c = couleur[ligne][colonne+1]
    liste.append(c)
    c = couleur[ligne+1][colonne+1]
    liste.append(c)
    return liste
```

4. Quelle est la valeur de voisins(rouge,1,1) ?

Structure du programme

```
def mediane(liste):
    liste_r=[]
    for c in liste:
        liste_r.append(c)
    liste_r.sort()
    return liste_r[4]

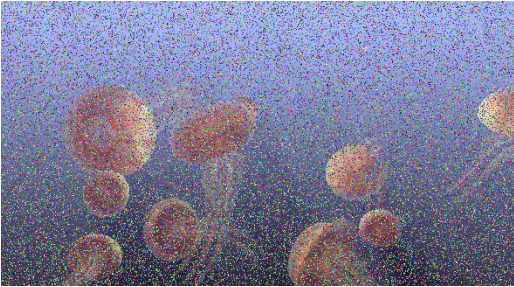
for y in range( 1,largeur-1):
    for x in range( 1,longueur-1):
        p=image.getpixel((x,y))
        Rouge = mediane(voisins(rouge,x,y))
        Vert = mediane(voisins(vert,x,y))
        Bleu = mediane(voisins(bleu,x,y))
```

5. Que vaut la liste Rouge à l'issue de ce code ?

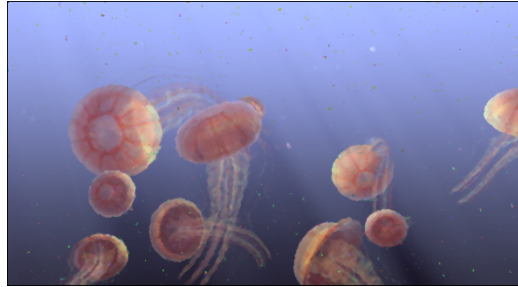
Exercice 4

A l'aide des exercices précédents, imaginer un programme qui applique le filtre médian sur une image

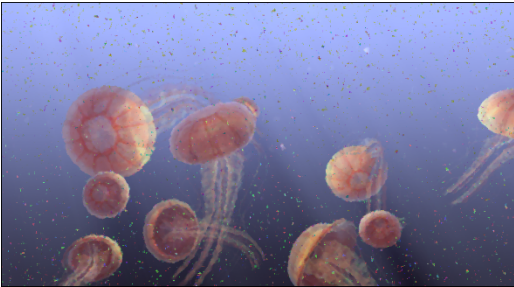
AVANT



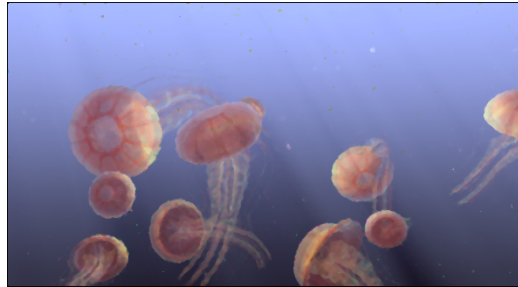
APRÈS 2 FILTRES MÉDIANS



APRÈS 1 FILTRE MÉDIAN



APRÈS 3 FILTRES MÉDIANS



Solutions

Exercice 1

Solution

```
from random import randint
from PIL import Image

image=Image.open("meduses.png")

longueur,largeur=image.size
image2=Image.new('RGB',(longueur,largeur))

for y in range(largeur):
    for x in range(longueur):
        z=randint(1,100)
        p=image.getpixel((x,y))
        if z<11:
            rouge=255
            vert=255
            bleu=255
        else:
            rouge=p[0]
            vert=p[1]
            bleu=p[2]
        image2.putpixel((x,y),(rouge,vert,bleu))

image2.save("image_bruitee.png")
image2.show()
```

Exercice 2

1. `mystere(6,8,4,3,9,10,20,7,11,24,0)` renvoie le 5^{ème} élément de la liste `[6,8,4,3,9,10,20,7,11,24,0]` une fois réordonnée dans l'ordre croissant : donc 8.
2. Le 5^{ème} élément d'une liste ordonnée de 9 nombres s'appelle la **médiane**

Exercice 3

1. `longueur=4` et `largeur=3`.
2. Ces trois listes sont identiques et valent `[[0,0,0],[0,0,0],[0,0,0],[0,0,0]]`
3. `voisins(rouge,1,1)=[155,169,161,160,71,161,166,164,161]`
4. Ensuite :
`rouge=[[155,160,166],[169,71,164],[161,161,161],[161,160,157]]`

Exercice 4

Ecrire un programme qui prenne une image bruitée, et colorie les pixels anormaux en utilisant la médiane des pixels voisins.

Solution

```
from random import randint
from PIL import Image

def mediane(liste):
    liste_r=[]
    for c in liste:
        liste_r.append(c)
    liste_r.sort()
    return liste_r[4]

def voisins(couleur,ligne,colonne):
    liste = []
    c = couleur[ligne-1][colonne-1]
    liste.append(c)
    c = couleur[ligne][colonne-1]
    liste.append(c)
    c = couleur[ligne+1][colonne-1]
    liste.append(c)
    c = couleur[ligne-1][colonne]
    liste.append(c)
    c = couleur[ligne][colonne]
    liste.append(c)
    c = couleur[ligne+1][colonne]
    liste.append(c)
    c = couleur[ligne-1][colonne+1]
    liste.append(c)
    c = couleur[ligne][colonne+1]
    liste.append(c)
    c = couleur[ligne+1][colonne+1]
    liste.append(c)
    return liste

image=Image.open( "image_bruitee.png")
longueur,largeur=image.size

rouge = [[0 for x in range( largeur)] for y in range(longueur)]
vert = [[0 for x in range( largeur)] for y in range(longueur)]
bleu = [[0 for x in range( largeur)] for y in range(longueur)]

image2=Image.new( 'RGB', (longueur,largeur))

for y in range(largeur):
    for x in range(longueur):
        p=image.getpixel((x,y))
        rouge[x][y] = p[0]
        vert[x][y] = p[1]
        bleu[x][y] = p[2]
for y in range( 1,largeur-1):
    for x in range( 1,longueur-1):
        p=image.getpixel((x,y))
        Rouge = mediane(voisins(rouge,x,y))
        Vert = mediane(voisins(vert,x,y))
        Bleu = mediane(voisins(bleu,x,y))
        image2.putpixel((x,y),(Rouge,Vert,Bleu))

image2.save( "image_reparee.png")
image2.show()
```