

# Creation et traitement d images avec Pil

April 10, 2020

## 0.1 Création et traitement d'images BitMap avec Python et Pil

Attention les fichiers images comme "buse.jpg" doit être présent dans le même dossier que ce présent fichier et tous les autres.

Note :les images créées s'affichent dans le logiciel configuré par défaut pour afficher les images. Il faut penser à le fermer pour pouvoir poursuivre l'exécution des programmes.

On pourra aller voir : <https://he-arc.github.io/livre-python/pillow/index.html>

## 0.2 Charger, Afficher, Sauvegarder une image, récupérer et changer la couleur d'un point

```
In [3]: from PIL import Image
        from random import *
        # monImage1=Image.open("buse.jpg") #Charger une image
        lg=100 #largeur de l'image
        monImage1=Image.new('RGB', (lg,lg))

        for k in range(100000):
            x=randint(0,monImage1.width-1)
            y=randint(0,monImage1.height-1)
            r,v,b = monImage1.getpixel((x, y)) #r,v,b prennent les valeurs des composante
            r,v,b=r+randint(0,10),v+randint(0,10),b+randint(0,10)
            monImage1.putpixel( (x, y), (r,v,b))#On fixe la couleur du pixel (x,y) de Image

        monImage1.save("Image Aléatoire.png") #On enregistre cette image
        monImage1.show() #On affiche une image composée de Image1 et Image2
```

## 0.3 Copier Coller

```
In [6]: from PIL import Image
        from random import *
        # monImage1=Image.open("buse.jpg") #Charger une image
        lg=10
        monImage1=Image.new('RGB', (lg,lg))

        for k in range(lg):
            monImage1.putpixel( (k, k), (255,0,0))#On fixe la couleur du pixel (x,y) de Image
            monImage1.putpixel( (lg-1-k, k), (0,255,0))#On fixe la couleur du pixel (x,y) d
```

```

monImage1.putpixel( (0, k),(0,0,255))#On fixe la couleur du pixel (x,y) de Image1
monImage1.putpixel( (k, 0),(0,0,255))#On fixe la couleur du pixel (x,y) de Image1

monImage1.save("Quadrillage1.png") #On enregistre cette image

n=100
monImage2=Image.new("RGB", (n*lg,n*lg))
for i in range(n):
    for j in range(n):
        monImage2.paste(monImage1, (i*lg, j*lg) )
monImage2.show() #On affiche une image composée de Image1 et Image2
monImage2.save("Quadrillage2.png") #On enregistre cette image

```

## 0.4 Dessiner et écrire

In [10]: `from PIL import Image, ImageDraw`

```

lg = 160
WHITE = (255, 255, 255, 0)
GREEN = (0, 255, 0, 0)
BLUE = (0, 0, 255, 0)
BLACK = (0, 0, 0, 0)

image = Image.new('RGBA', (width, height), WHITE)

draw = ImageDraw.Draw(image) # Obtention du contexte graphique
draw.line((0,lg//2, lg,lg//2), fill=GREEN)
draw.line((lg//2,0, lg//2,lg), fill=GREEN)

draw.ellipse((lg//4,lg//4, 3*lg//4,3*lg//4), outline=BLUE) #Le rectangle dans lequel

draw.text((20, 20), "Cercle trigonométrique", fill=BLACK)

image.show()
image.save("cercleTrigo.png", "png")

```

In [ ]: