

Module turtle pour introduire python

Niveau : *

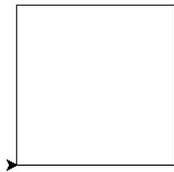
Python est un langage de programmation dont la syntaxe est particulièrement simple, il est très puissant et très utilisé (le créateur de ce langage travaillait chez Google et maintenant chez dropbox. La Nasa l'utilise aussi, pour ne citer qu'eux !). Pour vous familiariser avec le langage Python, essayez ces quelques commandes directement dans l'interpréteur python :

```

13 >>> from turtle import *
>>> forward(120)
>>> left(90)
>>> color("red")
>>> forward(80)
>>> reset() Analyser chacune de ces lignes et expliquer ce que fait
chacune des commandes.
    
```

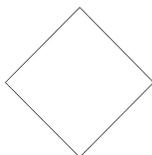
14 Carré

Tracer un carré de 100 pixels de côté.



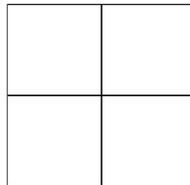
16 Carré ?

Tracer la figure ci-dessous. Le carré fait 100 pixels de côté.



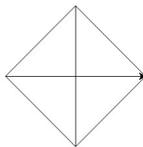
15 Carrés

Tracer la figure ci-dessous. Chaque carré fait 100 pixels de côté.



17 Carré ? (2)

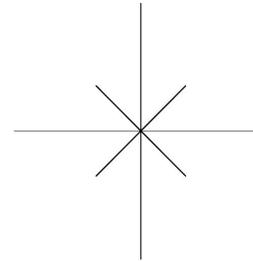
Tracer la figure ci-dessous. Le carré fait 100 pixels de côté.



Niveau : **

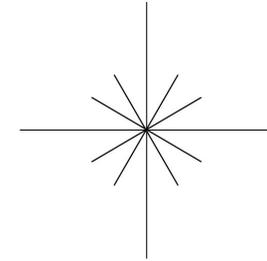
18 Cristal

Tracer le cristal ci-dessous. Les grands segments mesurent 300 pixels, les petits 150.



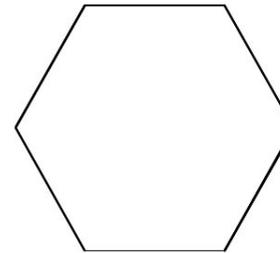
19 Cristal (2)

Tracer le cristal ci-dessous. Les grands segments mesurent 300 pixels, les petits 150.



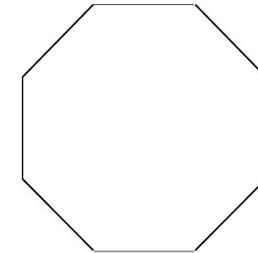
20 Hexagone

Tracer l'hexagone ci-dessous. Les 6 côtés mesurent 100 pixels.



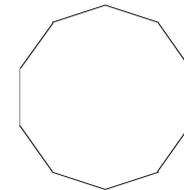
21 Octogone

Tracer le pentagone ci-dessous. Les 8 côtés mesurent 100 pixels.



22 (Double) décagone

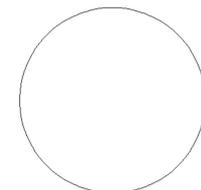
1. Tracer le décagone ci-dessous.



2. Tracer un double décagone

23 Cercle

Tracer le cercle ci-dessous.



$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

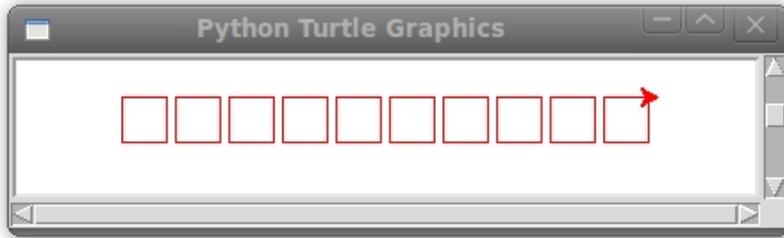
$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

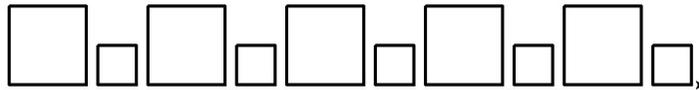
$$\mathbb{P} \left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2 \right) \approx 1 - \alpha$$

Niveau : ***

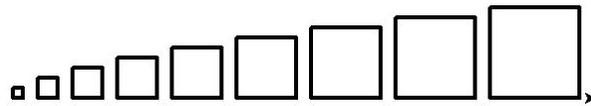
24 Réaliser le dessin ci-dessous à l'aide du module turtle :



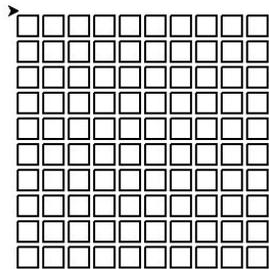
25 Réaliser le dessin ci-dessous à l'aide du module turtle :



26 Réaliser le dessin ci-dessous à l'aide du module turtle :



27 Réaliser le dessin ci-dessous à l'aide du module turtle :

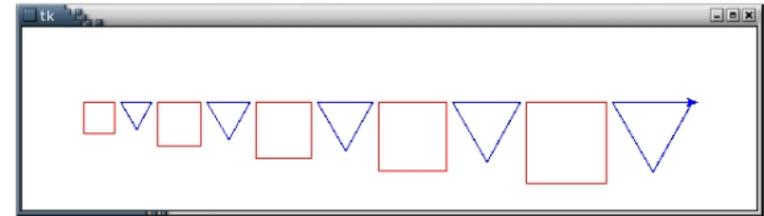


28 Modifier le code du programme précédent pour que le carré se trouvant en ligne 4 colonne 8 soit rouge

Niveau : ****

29 Ecrire un programme qui dessine un carré de coté 100 "plein"

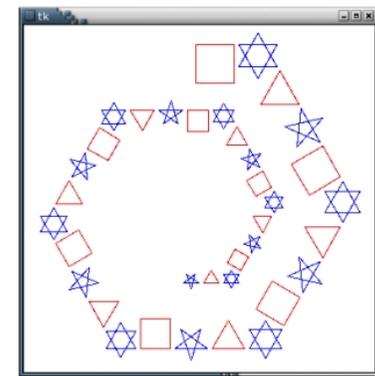
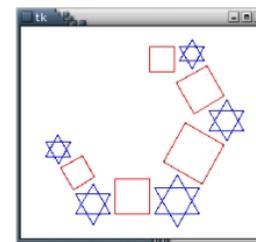
30 Réaliser le dessin suivant :



31 Dessiner la figure suivante :



32 Créer un script qui dessinera une série de ces étoiles :



$$\mathbb{E}(\phi(x)) = \int \phi(x) q_{\mathbb{B}}(x)$$

$$\mathbb{B}(A) = \sum_{i \in I} \mathbb{B}^i(A) \mathbb{B}^i(B^i)$$

$$\binom{x}{u} b_x (1-b)^{u-x} e^{-x_1} q_x = I \int_{-\infty}^{+\infty} \frac{\sqrt{2\pi}}{I}$$

$$\mathbb{I}_b \left(\sum_{j=1}^I \frac{W^{bj}}{(W^{bj} - M^{bj})^5} \right) \leq X_j^I - I^\alpha \leq I - \alpha$$

Mise en place de l'activité

Objectifs

Cette activité est la première que je propose en Python au mois de septembre depuis quelques années.

Elle a comme vocation de présenter la syntaxe python, la programmation séquentielle et textuelle.

Elle incite l'élève à demander de manière implicite la syntaxe de la boucle itérative et de l'instruction conditionnelle et montre l'intérêt de l'affectation. Elle permet également une diversification en fonction des demandes des élèves : possibilité de présenter la syntaxe d'une fonction (sans paramètres) ainsi que la syntaxe d'une fonction avec paramètres pour ceux qui m'en font la demande.

niveau

Je l'utilise pour tout niveau pour des élèves n'ayant jamais rencontrés Python.

Il me faut en général 2H pour arriver à faire émerger les notions.

Déroulement de l'activité

Après une présentation succincte de python (moins de deux minutes) je les laisse aller à leur rythme.

- Les exercices 2 et 3 leur permet de rentrer dans le logiciel et de prendre en main python. Ils sont tapés directement, instruction par instruction, dans l'interpréteur. A ce stade, je ne leur montre pas encore le bloc note.
- L'exercice 5 me permet d'agacer la plupart des élèves qui se voient obligés de "resetter" car ils ont tendance à trouver la longueur de la diagonale par tâtonnement avant d'arriver à mobiliser Pythagore. C'est l'occasion de leur présenter le bloc-note et la programmation textuelle ainsi que l'intérêt d'écrire son programme dans un fichier texte avant d'exécuter. On peut observer chez certains élèves des stratégies de contournement consistant à changer la couleur en "white" pour effacer. Cela les empêche d'avoir à resetter. C'est également l'occasion de revenir sur l'exercice 4 et de leur montrer qu'il est identique au 2 en rajoutant une seule instruction au début.
- Les exercices 6 et 7 me servent à faire "attendre" les élèves performants.

- Je passe directement sur l'exercice 8 après l'exercice 5 lorsque la totalité des élèves à réussi le 5
- Les exercices 8 à 11 imposent aux élèves une recherche de l'angle. Certains trouvent directement la bonne formule mais la majorité procède par tâtonnement et s'agace de devoir modifier toutes leurs lignes de code. J'en profite pour leur présenter l'affectation (angle = 37...).
- L'exercice 11 voit les élèves faire un polygône à 360 côtés en général. Certains me demandent si il n'y a pas moyen de faire "répéter". Les autres s'agaçant de devoir faire un copier coller me permet de leur présenter alors la syntaxe de la boucle itérative. Certains "petits malins" trouve l'instruction circle. Je leur demande alors de voir si il n'en existe pas une pour faire un carré. Après quelques essais infructueux, ils se convainquent que ce n'est pas le cas.
- L'exercice 12 me permet de leur montrer la syntaxe d'une procédure/fonction (sans paramètre) afin de créer une telle fonction par nous même. Je présente la possibilité de mettre des arguments aux élèves qui m'en font la demande.
- Les exercices 13 et 14 permettent de montrer l'intérêt d'une fonction à paramètre. C'est l'occasion de revenir sur les exercices demandant de tracer des polygônes et de montrer que le carré, le décagone etc...
Peuvent s'hériter de la fonction suivante :

```
def polygone(n,cote) :
    # n represente ne nombre de cote
    # cote est la longueur du cote
    angle = 360/n
    for k in range(n) :
        forward(cote)
        left(angle)

def carre(cote) :
    polygone(4, cote)

def decagone(cote) :
    polygone(10, cote)
```

- L'exercice 14 permet également d'introduire l'utilisation du k dans la boucle de répétition.
- L'exercice 15 introduit la double boucle pour
- L'exercice 16 permet d'introduire la syntaxe d'une instruction conditionnelle.
- Les exercices suivants me permettent de donner des exercices un

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{X-\mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2\right) \approx 1 - \alpha$$

peu défi à certains élèves performants.

$$\begin{aligned}
 \mathbb{I}_b(A) &= \sum_{i \in I} \mathbb{I}_{B^i}(A) \mathbb{I}_b(B^i) \\
 \binom{\kappa}{u} b_{\kappa}(1-b)_{u-\kappa} &= \frac{\sqrt{\sum_{\lambda} \dots}}{\Gamma} \int_{+\infty}^{-\infty} e^{-x_2} q x = \Gamma \frac{a}{x-a} \rightarrow \chi(0, \Gamma) \\
 \mathbb{E}(\phi(X)) &= \int \phi(x) q_{\mathbb{I}_b}(x) \\
 \mathbb{I}_b &\left(\sum_{j=1}^{\lambda} \frac{M^{b^j}}{(M^{b^j} - M^{b^l})^5} \right) \leq \chi_{\Gamma}^{1-\Gamma^{\alpha}} \leq \Gamma - \alpha
 \end{aligned}$$

Correction Python

Correction exercice 2

```
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
```

Correction exercice 4

```
left(45)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
```

Correction exercices polygones

Les élèves partent en général d'une syntaxe comme celle ci pour l'hexagone

```
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
forward(100)
left(60)
```

Mon objectif est d'arriver à les faire arriver à quelque chose comme cela (double décagone ici) :

```
angle = 360/20
longueur = 50
for k in range(20) :
    forward(longueur)
    left(angle)
```

Correction exercice 12

Dans un premier temps on obtient généralement ça :

```
color("red")
for k in range(10) :
    forward(100)
    left(90)
    forward(100)
    left(90)
    forward(100)
    left(90)
    forward(100)
    left(90)
    forward(100)
    left(90)
    up()
    forward(15+100)
    down()
```

Après presentation des procédures, on obtient ça :

```
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
```

```
color("red")
for k in range(10) :
    carre()
    up()
    forward(15+100)
    down()
```

Voir :

```
def carre() :
    for k in range(4) :
        forward(100)
        left(90)

color("red")
for k in range(10) :
    carre()
    up()
    forward(15+100)
    down()
```

Correction exercice 13

```
def carre(cote) :
```

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = 1$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\frac{(N\hat{p}_j - Np_j)^2}{Np_j} \leq \chi_{J-1, \alpha}^2 \approx 1 - \alpha$$

$$\mathbb{I}_b(A) = \sum_{i \in I} \mathbb{I}_b(A) \mathbb{I}_b(B^i)$$

$$\binom{k}{u} b_k (1-b)_{u-k}$$

$$\mathbb{E}(\phi(X)) = \int \phi(x) q_{\mathbb{I}_b}(x)$$

$$\int_{-\infty}^{+\infty} \frac{\sqrt{\Sigma}}{\Gamma} e^{-x/\Sigma} q_X = \Gamma \frac{a}{X-a} \rightarrow \chi(0, \Gamma)$$

$$\mathbb{I}_b \left(\sum_{j=1}^{\Gamma} \frac{M^{bj}}{(M^{bj} - M^{b1})^{\Sigma}} \right) \leq \chi_{\Sigma}^{\Gamma - \Gamma \alpha}$$

```

for k in range(4) :
    forward(cote)
    left(90)

for k in range(5) :
    carre(100)
    up()
    forward(100+15)
    down()
    carre(50)
    up()
    forward(50+15)
    down()
    
```

Correction exercice 14

```

for k in range(1,10) :
    carre(10*k)
    up()
    forward(10*k+15)
    down()
    
```

Correction exercice 15

```

for i in range(10) :
    for j in range(10) :
        carre(20)
        up()
        forward(20+5)
        down()

        up()
        backward(250)
        right(90)
        forward(20+5)
        left(90)
        down()
    
```

Correction exercice 16

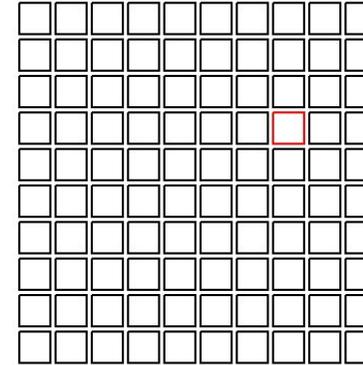
```

for i in range(10) :
    for j in range(10) :
        if i == 3 and j == 7 :
            color("red")
        else :
            color("black")
        carre(30)
        up()
        forward(30+5)
        down()

    up()
    backward(350)
    
```

```

right(90)
forward(30+5)
left(90)
down()
    
```



Images numériques

Prendre en main la bibliothèque PIL

1 Dans l'interpréteur d'Edupython

- Ouvrir un nouveau fichier python avec **Edupython** que vous appellerez `depart.py`. **Sauvegarder** le dans votre dossier personnel, puis **exécuter** le.
- Dans l'interpréteur d'Edupython taper :


```
>>> from PIL import Image
>>> im = Image.new("RGB", (600,300), "grey")
>>> type(im)
>>> im.save("trace.png", "PNG")
```
- A quoi sert la fonction `Image.new` et comment fonctionne-t-elle ? On n'hésitera pas à regarder ce qu'il se passe dans le dossier contenant le fichier `depart.py`.
- A quoi sert la méthode `.save` et comment fonctionne-t-elle ?

2 Toujours dans l'interpréteur d'Edupython

- Dans l'interpréteur python taper :


```
>>> im.putpixel( (10,10), (0,0,0) )
>>> im.save("trace.png", "PNG")
```
- A quoi sert la méthode `.putpixel` et comment fonctionne-t-elle ?
- Pourquoi `.save` est-il indispensable ?

Création d'images

3 Premier programme

- Dans le bloc note d'Edupython, écrire un programme qui créera une image toute jaune de taille 800 par 800 pixels. Cette image devra s'appeler `premiere_image.png`
- Modifier le programme pour que l'image contienne un pixel noir, un bleu, un rouge, un vert et un blanc.
- Modifier votre programme pour que votre image contienne un pixel de couleur "skyblue".
- Comparer le résultat si on avait décidé d'appeler notre image `premiere_image.jpg`
Expliquer pourquoi les résultats sont différents.

Pour les exercices de cette page, on partira d'une image carrée de taille 800 par 800 dont le fond est gris.

4 Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



5 Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



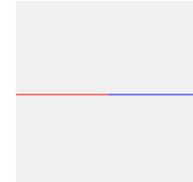
6 Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



7 Écrire un programme permettant d'obtenir une "ligne en pointillée" comme ci-dessous :



8 Écrire un programme permettant d'obtenir une "ligne bicolore" comme ci-dessous :



9 Dessiner les drapeaux suivants :



$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P} \left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi_{J-1, \alpha}^2 \right) \approx 1 - \alpha$$

Traitement d'images

Quelques commandes en plus

10 Fichiers attachés :

- fichier retouche.py 
- lena en couleur : jpg  png 
- lena en noir et blanc 
- papillon 
- lena en niveau de gris 10 par 10 
- canyon Noir et blanc 
- canyon couleurs 

1. Dans votre espace personnel, créer un dossier contenant toutes les images.
2. Ouvrir le fichier retouche.py avec **Edupython** et **enregistrer** le dans le même dossier que celui qui contient les quatre images.
3. Après avoir **exécuter** le fichier retouche.py, déterminer ce que fait la commande .open
4. Quel est le type de img ?

11 Dans cet exercice on va travailler sur l'image papillon.png qui est en noir et blanc.

Préalable : Modifier le programme retouche.py de manière à importer le papillon dans une ImageFile que vous appellerez pap.

Exécuter le programme retouche.py.

Grâce à l'interpréteur, déterminer ce que font les commandes suivantes :

1. >>> pap.format
2. >>> pap.size
3. >>> pap.show()
4. >>> pap.getpixel((3,7))
5. >>> pap.putpixel((3,7),128)
6. >>> pap2 = pap.copy()
7. >>> pap.mode

Opérations sur un pixel

12 Négatif d'une image en noir et blanc

1. Effectuer le négatif de l'image papillon.png
2. Effectuer le négatif de l'image lena1010.png.
3. Effectuer le négatif de l'image lena.png

13 Comprendre les images couleurs

1. importer lena.jpg dans une Imagefile que vous appellerez lena (en dehors d'une fonction).
2. Exécuter votre fichier.
3. Tester et comprendre les commandes suivantes dans l'interpréteur :


```
>>> valeurs = len.getpixel((3,7))
>>> valeurs
>>> valeurs[0] = 35
>>> valeurs
>>> (R,G,B) = len.getpixel((3,7))
>>> G
```
4. Comment fonctionne len.getpixel ?

14 Transformer une image couleur en noir et blanc

1. Traiter lena.jpg en remplaçant, pixel par pixel, chacune des valeurs des composantes des couleurs par la moyenne des trois.
2. Dans lena.pgm, toujours pixel par pixel, remplacer chaque composante des couleurs par la valeur commune :

$$0,21 \times R + 0,71 \times G + 0,07 \times B$$

3. Répéter la même opération mais en prenant cette fois ci :

$$0,299 \times R + 0,587 \times G + 0,114 \times B$$

15 Négatif d'une image en couleurs

Effectuer le négatif de l'image lena.jpg

16 miroir

1. Effectuer le miroir de l'image papillon.png
2. Effectuer le miroir de l'image lena.png
3. Effectuer le miroir de l'image lena.jpg

17 Images non carrées

1. Effectuer le négatif de canyon.png et canyon.jpg
2. Effectuer le miroir de canyon.png et canyon.jpg

$$\mathbb{I}_b(A) = \sum_{i \in I} \mathbb{I}_b(A) \mathbb{I}_b(B^i)$$

$$\mathbb{I}_b(\phi(X)) = \int \phi(x) q_{\mathbb{I}_b} X(x)$$

$$\frac{\sqrt{\sum_{u,v} e^{-x_u \setminus v} q_{X^u} = I}}{I} \rightarrow \chi(0, I)$$

$$\mathbb{I}_b \left(\sum_{j=1}^J \frac{M^{b^j}}{(M^{b^j} - M^{b^j})^5} \right) \leq \chi^2 - I^\alpha$$

$$\int \dots \rightarrow I - \alpha$$

Correction Activités retouche images

Correction : 

Correction exercice 4

```
im = Image.new("RGB" , (800,800) , "yellow")
im.putpixel((1,1), (255,255,255))
im.putpixel((1,2), (255,0,0))
im.putpixel((1,3), (0,255,0))
im.putpixel((1,4), (0,0,255))
im.putpixel((1,5), (0,0,0))
im.putpixel((1,6), (135,206,235))
im.save("premiere_image.png", "png")
im.save("premiere_image.jpg", "jpeg")
```

Correction exercice 5

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(800):
    im.putpixel((k,400), (255,255,255))
im.save("image.png", "png")
```

Correction exercice 6

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(800):
    im.putpixel((k,k), (0,0,0))
im.save("image.png", "png")
```

Correction exercice 7

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(800):
    im.putpixel((799-k,k), (0,0,0))
im.save("image.png", "png")
```

Correction exercice 8

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(0,800,2):
    im.putpixel((k,400), (0,0,0))
im.save("image.png", "png")
```

Correction exercice 9

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(400):
    im.putpixel((k,400), (255,0,0))
for k in range(400,800):
    im.putpixel((k,400), (0,0,255))
im.save("image.png", "png")
```

```
im = Image.new("RGB" , (800,800) , "grey")
for k in range(800):
    if k<400:
        im.putpixel((k,400), (255,0,0))
    else:
        im.putpixel((k,400), (0,0,255))
im.save("image.png", "png")
```

Correction exercice 9

```
"""dimension et couleur : source wikipedia"""
im = Image.new("RGB" , (900,450) , "grey")
for l in range(450):
    for c in range(900):
        if c<300:
            im.putpixel((c,l), (5,20,64))
        elif c<600:
            im.putpixel((c,l), (255,255,255))
        else:
            im.putpixel((c,l), (236,25,32))
im.save("image.png", "png")
```

Correction exercice 12

```
"""renvoie le negatif du papillon"""
pap=Image.open("papillon.png")
for l in range(10):
    for c in range(10):
        valeur = pap.getpixel((c,l))
        if valeur == 0:
            pap.putpixel((c,l),255)
        else:
            pap.putpixel((c,l),0)
pap.save("image.png", "png")
```

```
"""renvoie le negatif de lena1010.png"""
lena=Image.open("lena1010.png")
for l in range(10):
    for c in range(10):
```

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\binom{n}{k} p^k (1-p)^{n-k}$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\frac{\bar{X} - \mu}{\sigma} \rightarrow \mathcal{N}(0,1)$$

$$\mathbb{P} \left(\frac{\sum_{j=1}^J (N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi^2_{J-1, \alpha} \right) \simeq 1 - \alpha$$

$$\mathbb{E}(\phi(X)) = \int \phi(x) q_{\mathbb{B}}(x) dx$$

$$\mathbb{B}(A) = \sum_{i \in I} \mathbb{B}^{(i)}(A) \mathbb{B}^{(i)}(B)$$

$$\int_{-\infty}^{+\infty} \frac{\sqrt{2\pi}}{I} e^{-x^2/2} dx = I$$

$$\frac{M^{(b)}}{(M^{(a)} - M^{(b)})^5} \leq X^2 - I^\alpha$$

$$\sum_{i=1}^I \left(\sum_{j=1}^I \frac{M^{(b)}}{(M^{(a)} - M^{(b)})^5} \right) \leq I - \alpha$$

```

    valeur = lena.getpixel((c,l))
    nouvelle_valeur = 255-valeur
    lena.putpixel((c,l), nouvelle_valeur)
    lena.save("image.png", "png")

```

```

"""renvoie le negatif de lena.png"""
lena=Image.open("lena.png")
for l in range(384):
    for c in range(384):
        valeur = lena.getpixel((c,l))
        nouvelle_valeur = 255-valeur
        lena.putpixel((c,l), nouvelle_valeur)
    lena.save("image.png", "png")

```

Correction exercice 14

```

"""transforme lena couleur en noir et blanc par moyenne des pixels"""
lena=Image.open("lena.jpg")
for l in range(384):
    for c in range(384):
        valeurs = lena.getpixel((c,l))
        nouveaux = (valeurs[0]+valeurs[1]+valeurs[2])/3
        lena.putpixel((c,l), (nouveaux,nouveaux,nouveaux))
    lena.save("image.png", "png")

```

```

"""transforme lena couleur en noir et blanc
0, 21R + 0, 71G + 0, 07B"""
lena=Image.open("lena.jpg")
for l in range(384):
    for c in range(384):
        valeurs = lena.getpixel((c,l))
        nouveaux = 0.21*valeurs[0]+0.71*valeurs[1]+0.07*valeurs[2]
        nouveaux = int(nouveaux)
        lena.putpixel((c,l), (nouveaux,nouveaux,nouveaux))
    lena.save("image.png", "png")

```

```

"""transforme lena couleur en noir et blanc
0, 299R + 0, 587G + 0, 114B"""
lena=Image.open("lena.jpg")
for l in range(384):
    for c in range(384):
        (R,G,B) = lena.getpixel((c,l))
        nouveaux = 0.21*R+0.71*G+0.07*B
        nouveaux = int(nouveaux)
        lena.putpixel((c,l), (nouveaux,nouveaux,nouveaux))
    lena.save("image.png", "png")

```

Correction exercice 15

```

"""Negatif de lena en couleurs """
lena=Image.open("lena.jpg")
for l in range(384):
    for c in range(384):
        (R,G,B) = lena.getpixel((c,l))
        nouveaux = (255-R,255-G,255-B)
        lena.putpixel((c,l), nouveaux)
    lena.save("image.png", "png")

```

Correction exercice 16

```

"""miroir du papillon"""
pap=Image.open("papillon.png")
pap2 = pap.copy()
for l in range(10):
    for c in range(10):
        valeur = pap.getpixel((c,l))
        pap2.putpixel((9-c,l), valeur)
    pap2.save("image.png", "png")

```

```

"""miroir lena noir et blanc"""
lena=Image.open("lena.png")
lena2 = lena.copy()
for l in range(384):
    for c in range(384):
        valeur = lena.getpixel((c,l))
        lena2.putpixel((383-c,l), valeur)
    lena2.save("image.png", "png")

```

```

"""miroir lena couleur"""
lena=Image.open("lena.jpg")
lena2 = lena.copy()
for l in range(384):
    for c in range(384):
        valeurs = lena.getpixel((c,l))
        lena2.putpixel((383-c,l), valeurs)
    lena2.save("image.png", "png")

```

Correction exercice 16

```

def negatif_NB(img):
    """fait le negatif d'une Imagefile (img) en noir et blanc"""
    (large,hauteur) = img.size
    img2 = img.copy()
    for l in range(hauteur):
        for c in range(large):

```

$$\mathbb{E}(\varphi(X)) = \int \varphi(x) d\mathbb{P}_X(x)$$

$$\binom{n}{k} p^k (1-p)^{n-k} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}_{B_i}(A) \mathbb{P}(B_i)$$

$$\mathbb{P}\left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi^2_{J-1, \alpha}\right) \approx 1 - \alpha$$

$\bar{X} - \frac{\mu}{\sigma} \rightarrow \mathcal{N}(0,1)$
 \uparrow
 \leftarrow

```

    valeur = img.getpixel((c,l))
    nouveau = 255-valeur
    img2.putpixel((c,l),nouveau)
    img2.save("image.png","png")
#img = Image.open("canyon.png")
#negatif_NB(img)

```

```

def negatif(img) :
    """fait le negatif d'une Imagefile (img) couleur"""
    (large,haut) = img.size
    img2 = img.copy()
    for l in range(haut) :
        for c in range(large) :
            (R,G,B) = img.getpixel((c,l))
            nouveaux = (255-R,255-G,255-B)
            img2.putpixel((c,l),nouveaux)
    img2.save("image.jpg","jpeg")
#img = Image.open("canyon.jpg")
#negatif(img)

```

```

def miroir_NB(img) :
    """fait le miroir d'une Imagefile en Noir et blanc"""
    (large,haut) = img.size
    img2 = img.copy()
    for l in range(haut) :
        for c in range(large) :
            valeur = img.getpixel((c,l))
            img2.putpixel((large-1-c,l),valeur)
    return img2
#img = Image.open("canyon.png")
#img2 = miroir_NB(img)
#img2.save("image.png","png")

```

```

def miroir(img) :
    """fait le miroir d'une Imagefile en couleurs"""
    (large,haut) = img.size
    img2 = img.copy()
    for l in range(haut) :
        for c in range(large) :
            valeurs = img.getpixel((c,l))
            img2.putpixel((large-1-c,l),valeurs)
    return img2
img = Image.open("canyon.jpg")
img2 = miroir(img)
img2.save("image.jpg","jpeg")

```

$$\mathbb{I}_b(A) = \sum_{i \in I} \mathbb{I}_b(A)_{\mathbb{I}_b(B^i)}$$

$$\mathbb{I}_b(\phi(X)) = \int \phi(x) q_{\mathbb{I}_b}(x)$$

$$\int_{-\infty}^{+\infty} e^{-x^2/2} dx = \sqrt{2\pi}$$

$$\mathbb{I}_b \left(\sum_{j=1}^J \frac{M^{(j)}}{M^{(j)} - M^{(b)}} \right) \leq X^{J-1} \alpha$$

$$\int_{-\infty}^{+\infty} \frac{1}{x} dx \rightarrow \gamma(0, \Gamma)$$

Exercice 1, 2 et 3

Ces exercices sont l'occasion de prendre en main le module PIL et de voir les premières instructions Image.new, .save et .putpixel.

Exercice 4, 5, 6, 7, 8 et 9

Ces exercices ont pour vocation à amener l'élève en autonomie à construire par lui même la double boucle pour permettant le traitement d'une image pixel par pixel.

Exercice 10 et 11

Présentation d'autres commandes utiles pour amener l'élève à construire le négatif d'une image :

- .format
- .size
- .show()
- .getpixel((c,l))
- .copy()
- .mode

Exercice 12

C'est l'exercice "cœur de séance" : construire le négatif d'une image. Je commence par une image 10 par 10 en noir et blanc sans niveau de gris. Des élèves vont le faire pixel par pixel sans utiliser getpixel mais en regardant l'images. Je les laisse faire.

La deuxième image est une image 10 par 10 en niveau de gris. Les élèves sont alors amenés à être plus fin pour récupérer la valeur (getpixel). Les élèves en autonomie créent une seule boucle pour au début pour faire les lignes.

Je termine enfin par une grosse image plus de 300 par 300 pour vraiment montrer l'intérêt de la double boucle pour. Volontairement les images sont carrés pour ne pas embêter les élèves à différencier hauteur et largeur. (voir exercice 17)

Exercice 13

L'exercice 13 me sert à introduire les images couleurs et le getpixel les manipulant de deux manière différentes.

Exercice 14 et 15

Exercice classique et basique de manipulation des images couleurs.

Exercice 16

La difficulté de cet exercice vient du fait que les élèves modifient leur image de départ. Ainsi arrivé à la moitié, ils ont perdu l'information de la partie

gauche. Ils est alors nécessaire de faire une copie de l'image de départ. Ce qui constitue une réelle difficulté pour eux, au même titre que travailler avec les indices.