

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

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

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

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

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

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

Table des matières

Thématique web : algorithme de moteur de recherche	1
Activité : (débranchée) Comprendre l'algorithme du moteur de recherche de GOOGLE	1
mise en place de l'activité	5
Thématique la photographie numérique : algorithme de retouche d'images	6
Activité : module turtle pour introduire python (3H en mathématiques)	6
Mise en place de l'activité	8
Correction Python	10
Activité : Comprendre ce qu'est une image	12
Activité : Retouche d'images	13
Correction Python	14
Mise en place de l'activité	17

Comprendre l'algorithme d'un moteur de recherche

Fichiers attachés : algo_google.py

Matériel :

- Des dés à 2,3,4,5 et 6 faces pour chacun des groupes
- Des dés à 100 faces pour chacun des groupes

comment un moteur de recherche classe les sites ?

Pour hiérarchiser la pertinence de site web sur un domaine, plusieurs solutions assez naturelles existent :

- On peut faire appel à une sommité (personne ou groupe de personnes) du domaine pour classer les pages. Le classement serait certes très pertinent pour ce domaine mais il faudrait solliciter un grand nombre de spécialistes de différents domaines pour obtenir un algorithme pouvant traiter des requêtes dans tous les domaines possibles et imaginables.

Il se pose aussi la difficulté de gérer le très grand nombre de pages évoquant une requête (plusieurs millions au minimum) ainsi que l'actualisation fréquente de ce classement.

- Une deuxième solution serait de demander aux internautes eux-mêmes de voter, pour chaque domaine, et de choisir le classement sur la base de ce vote, considérant, que compte-tenu du grand nombre de votants, le classement obtenu serait pertinent.

Ces deux modèles, pourtant intéressants et logiques (ils sont mis en œuvre par wikipédia), font intervenir l'humain et trouvent leur limite dans le trop grand nombre de pages à gérer qui sont de plus évolutives. Ce qui conduit à privilégier des protocoles entièrement automatisables sous forme d'algorithme.

Dans cet optique, Google a cherché un modèle de hiérarchisation qui soit exploitable dans tous les domaines, utilisable pour tous les mots clés, adaptable à un très grand nombre de données, même évolutives, tout en étant automatisable et suffisamment efficace.

C'est en répondant à ce cahier des charges que ce nouveau venu a réussi l'exploit, en quelques mois et malgré l'émergence de Bing ou encore Qwant, à obtenir le quasi-monopole de la recherche thématique sur le web.

L'idée à la base du modèle de Larry Page et Sergey Brin, fondateurs de Google, revient à attribuer à chaque page un nombre positif entre 0 et

1, appelé score (en anglais "PageRank") de la page, qui caractérisera la pertinence de cette page. Ils proposent alors de déterminer ce score à partir des deux règles suivantes :

- R1** : Le score attribué à une page doit être d'autant plus élevé que celle-ci est référencée dans une page faisant autorité (dont le score est élevé).
- R2** : Le score attribué à une page doit être d'autant moins élevé que celle-ci est référencée dans une page contenant un grand nombre de références.

Leur idée :

utiliser un surfeur aléatoire.

Principe du surfeur aléatoire

Après avoir fait la liste (sans classement) de tous les sites traitant la requête, le surfeur aléatoire en choisit au hasard un. Puis il s'intéresse aux liens hypertexte du site sur lequel il se trouve vers les autres sites qu'il a listé. Il en choisit alors un au hasard et répète cette opération sans s'arrêter en comptant pour chacun des sites combien de fois il l'a visité. Les sites sont alors affichés dans l'ordre décroissant de leur nombre de visites.

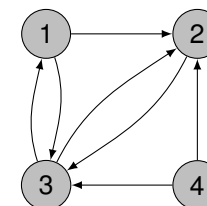
Ainsi pour un certain mot clé rentré, il s'intéresse aux sites qui évoquent ce mot-clé mais également aux liens hyper-texte qui permettent de passer d'un site à l'autre.

1 Et concrètement : un premier exemple

Matériel : un dé à six face

Pour illustrer comment un algorithme de calcul peut être mis en place à partir de ces règles, nous allons prendre l'exemple du classement de quatre pages.

Le problème de l'attribution du score peut être représenté par un graphe orienté : les quatre pages sont représentées par les quatre sommets d'un graphe dont les arêtes orientées représentent les références (liens) pouvant exister entre ces différentes pages.

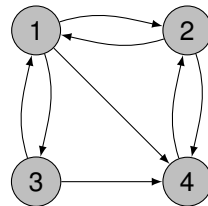


Dans ce graphe, la flèche allant de 1 vers 2 signifie que la page 1 référence la page 2 et l'absence de flèche de 2 vers 4 signifie que la page 2 ne référence pas la page 4.

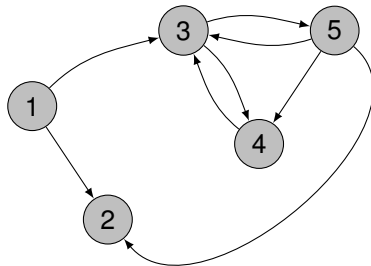
1. Choisissez un site parmi les 4 qui sera votre point de départ pour tout l'exercice.
2. Comment avec un dé pouvez-vous simuler un déplacement aléatoire de notre surfeur ?
3. Simuler pendant un certain temps le surfeur aléatoire en n'oubliant pas de noter le nombre de fois où il est passé par site.
4. Proposer un classement de ces 4 pages.
5. Comparer votre classement avec les autres groupes.
6. Est ce intéressant de comparer vos effectifs avec ceux des autres groupes ? Que peut-on faire pour remédier à ce problème ? Que remarque-t-on alors ?

2 Un deuxième graphe

Estimer le PageRank des sites représentés par le graphe ci-dessous. Comparer vos scores avec les autres groupes. On effectuera 50 surfs aléatoires.



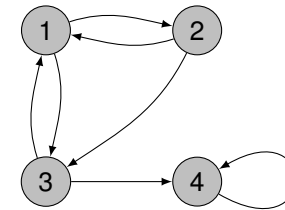
3 Un premier problème La technique du surfeur aléatoire ne marche pas pour le graphe suivant :



1. Pourquoi ?
2. Proposer une solution pour pallier à ce problème.
3. Faites alors une proposition de classement pour ce graphe après avoir calculé le PageRank.
4. Comparer vos résultats avec les autres groupes.

4 Puits

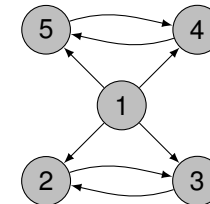
1. Calculer les PageRank du graphe suivant. Comparer le avec les autres groupes.



2. Quel problème moral est soulevé par ce graphe ?
3. Avez vous une solution à proposer pour pallier à ce problème ?

5 Poche web

1. Estimer les PageRank des sites suivants. Comparer avec les autres groupes.



2. Quel problème est soulevé par ce graphe ?
3. Avez vous une solution à proposer pour pallier à ce problème ?

Heureusement il y a les maths !

Un théorème complexe d'algèbre linéaire qui peut s'adapter à notre cas : le théorème de Peyron-Froebenius nous dit que si de chaque sommet part un lien vers chacun des autres sommets alors les fréquences des positions au cours de notre surf aléatoire vont toujours converger vers la même valeur et cela indépendamment de la position de départ.

Idée Brillante de L. Page et S. Brin

Préalable : Si une page ne comporte aucun lien vers l'extérieur, comme le sommet 2 du graphe 3, on crée artificiellement un lien de la page vers toutes les autres pages.

Détail de l'idée :

- À chaque étape, on continue la promenade aléatoire précédente avec une probabilité de 0.85 ;
- et avec probabilité de 0.15, on fait un saut aléatoire (vers n'importe quelle page) avec une probabilité $\frac{1}{n}$ de tomber sur une page donnée, où n est le nombre de pages.

On peut démontrer mathématiquement que cette façon de procéder permet de faire systématiquement rentrer le graphe considéré dans le champ d'application du théorème de Peyron-Froebenius.

Cela garantit donc la convergence des fréquences vers des valeurs limites qui seront considérées comme étant les pages-Rank. Leur méthode résout au passage le cas des puits car elle empêche de se retrouver dans une poche du web sans pouvoir en sortir.

De plus, plutôt que de calculer les valeurs limites, calculs qui se révèlent dans la pratique très long, l'algorithme du PageRank simule comme l'on vient de le faire un surfeur aléatoire et prend les fréquences trouvées comme estimation des valeurs limites.

6 Modélisation de leur idée

Proposer un protocole pour modéliser leur idée avec un dé à 100 faces

7 Et en pratique

Matériel : Dés à 6 faces et à 100 faces

1. Grâce à la méthode des fondateurs de Google, proposer un classement des pages des quatrième et cinquième graphes.
2. Leur méthode aboutie modifie-t-elle le classement de pages pour lequel il n'y avait pas de problèmes ?

plop

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

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

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

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

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

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

Mise en place de l'activité

Exercice 1

Ce premier exercice a pour but de faire manipuler le surfeur aléatoire aux élèves au moyen d'un dé. Il permet de ré-expliquer la consigne éventuellement à des groupes qui l'aurait mal comprise.

En effet la convergence des fréquences étant assez rapides, avec une trentaine de saut, les différents groupes devraient avoir le même classement. Un groupe qui aurait un classement étonnant est vraisemblablement un groupe qui a mal compris le principe.

D'autre part, sans autres consignes, notamment sur le nombre de surf aléatoire à exécuter, la comparaison des effectifs n'a que peu de sens (même si l'ordre sera le même). L'idée est de faire émerger que la bonne quantité à comparer est la fréquence de visite de chaque site.

Exercice 2

Après avoir pris un temps de remédiation pour les groupes qui n'avaient pas tout à fait compris le principe du surf, cet exercice ou l'on invitera les différents à donner la fréquence plutôt que l'effectif, permet d'illustrer le principe de l'algorithme. Les fréquences sont très similaires, et cela indépendamment du site de départ.

Cet exercice permet aussi une deuxième remédiation pour le groupe qui n'a toujours pas compris le principe du surf aléatoire. Il aura en effet des valeurs de fréquence divergentes des autres groupes.

Exercice 3

Cet exercice présenter le premier écueil de l'algorithme naïf : que faire quand un site n'a aucun lien vers d'autres sites.

Assez spontanément les élèves ont tendance à rajouter des flèches vers les autres sites. Cette solution : le surf équiprobable, permet de solutionner cette difficulté.

Exercice 4 et 5

Ces deux exercices présentent le principal écueil de l'algorithme : le blocage dans un puit ou une poche du web.

Même si les élèves peuvent avoir de nombreuses idées pour s'en sortir, la solution choisie par Google ne peut être trouvée à cause de la trop grande difficulté mathématique : les chaînes de markov et le théorème de Perron-Frobenius se trouvant en dehors de leur connaissance.

C'est le moment de leur présenter la solution mise en place par google.

6 et 7

Ces exercices peuvent être à traiter en classe, si il reste du temps ou à la maison.

Il est également judicieux de faire simuler le surfeur aléatoire à l'aide du fichier algo_google.py

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

8 >>> from turtle import *

>>> forward(120)

>>> left(90)

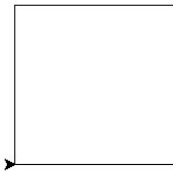
>>> color("red")

>>> forward(80)

>>> reset() Analyser chacune de ces lignes et expliquer ce que fait chacune des commandes.

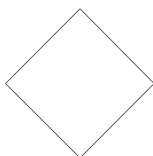
9 Carré

Tracer un carré de 100 pixels de côté.



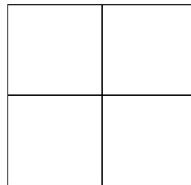
11 Carré ?

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



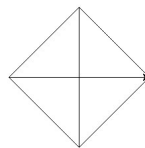
10 Carrés

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



12 Carré ? (2)

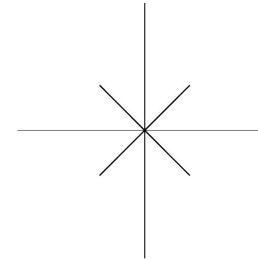
Tracer la figure ci-dessous.
Le carré fait 100 pixels de côté.



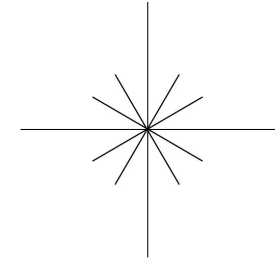
Niveau : **

13 Cristal

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

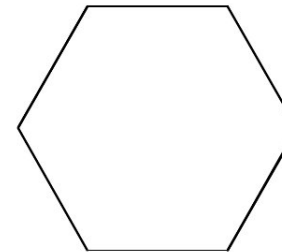


14 Cristal (2) Tracer le cristal ci-dessous. Les grands segments mesurent 300 pixels, les petits 150.



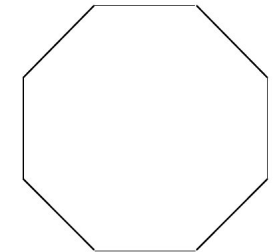
15 Hexagone

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



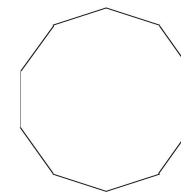
16 Octogone

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



17 (Double) décagone

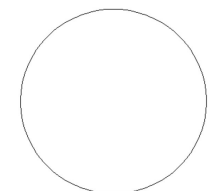
1. Tracer le décagone ci-dessous.



2. Tracer un double décagone

18 Cercle

Tracer le cercle ci-dessous.



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

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

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

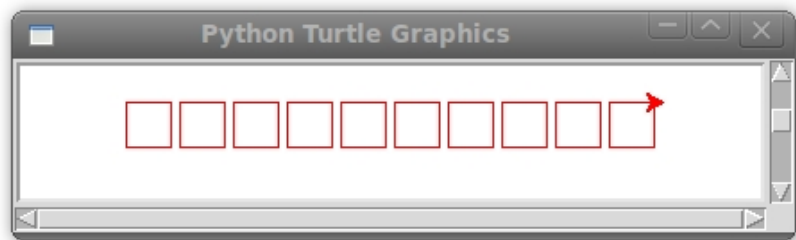
$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} dx = 1$$

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

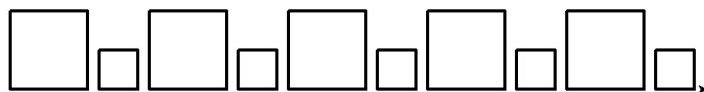
$$\mathbb{P} \left(\sum_{j=1}^J \frac{(N_{\hat{p}_j} - N_{p_j})^2}{N_{p_j}} \leq \chi^2_{J-1, \alpha} \right) \simeq 1 - \alpha$$

Niveau : ***

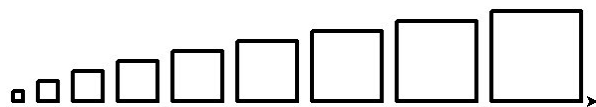
19 Réaliser le dessin ci-dessous à l'aide du module turtle :



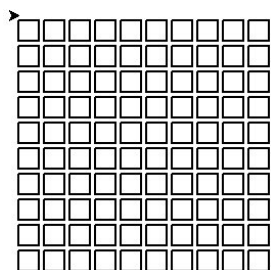
20 Réaliser le dessin ci-dessous à l'aide du module turtle :



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



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

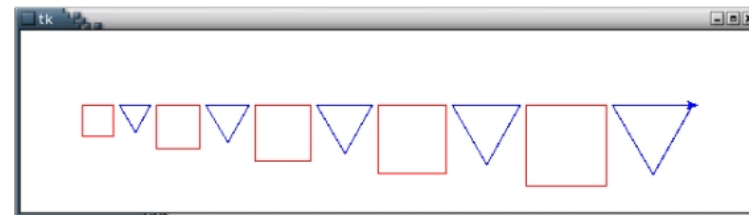


23 Modifier le code du programme précédent pour que le carré se trouvant en ligne 4 colonne 8 soit rouge

Niveau : ****

24 Ecrire un programme qui dessine un carré de côté 100 "plein"

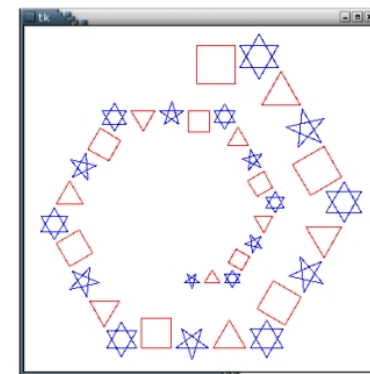
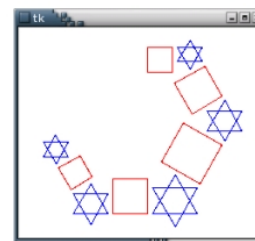
25 Réaliser le dessin suivant :



26 Dessiner la figure suivante :



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



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 polygone 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 polygones 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

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

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

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

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

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

peu défi à certains élèves performants.

Correction Python

Correction exercice 2

```
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)
```

```
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):
```

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

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

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

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

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

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

```
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 exercise 14

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

Correction exercise 15

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

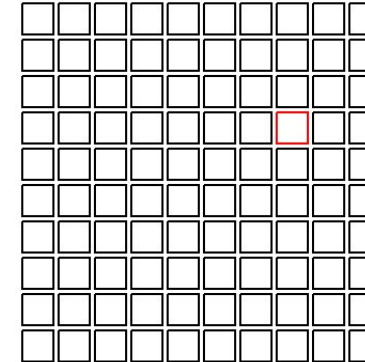
    up()
    backward(250)
    right(90)
    forward(20+5)
    left(90)
    down()
```

Correction exercise 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

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


```
>>> from PIL import Image
>>> im = Image.new("RGB", (600,300), "grey")
>>> type(im)
>>> im.save("trace.png", "PNG")
```
3. 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.
4. A quoi sert la méthode .save et comment fonctionne-t-elle ?

2 Toujours dans l'interpréteur d'Edupython

1. Dans l'interpréteur python taper :


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

Création d'images

3 Premier programme

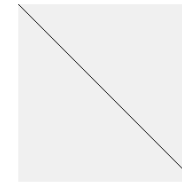
1. Dans le bloc note d'Edupython, écrire un programme qui créera une image toute jaune de taille 800 par 800 pixels. Cette image devra s'appeler premiere_image.png
2. Modifier le programme pour que l'image contienne un pixel noir, un bleu, un rouge, un vert et un blanc.
3. Modifier votre programme pour que votre image contienne un pixel de couleur "skyblue".
4. 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 :



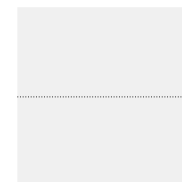
5 Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



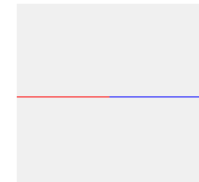
6 Tracer un segment de droite. Vous devez obtenir un résultat comme ci-dessous :



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 :



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

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):
```

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

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

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

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

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

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

```
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,haut) = img.size
    img2 = img.copy()
    for l in range(haut) :
        for c in range(large) :
```



```

        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")

```

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

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

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

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

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

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

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 perdus 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.