python et turtle

Activité 1

Nous allons découvrir comment réaliser un dessin en langage Python à l'aide du module turtle. Ce module est un peu comme une bibliothèque d'instructions utilisables pour dessiner. Nous découvrirons dans la présente activité les quelques commandes suivantes :

Commande	Effet
reset( )	On efface tout et on recommence
down( )	Abaisser le crayon (pour dessiner)
up( )	Relever le crayon (pour avancer sans dessiner)
forward(distance) ou fd( )	Avancer d'une distance donnée
backward(distance) ou bk( )	Reculer
left(angle) ou lt( )	Tourner à gauche d'un angle donné en degrés
right(angle) ou rt( )	Tourner à droite
color(couleur)	Définit la couleur du trait (la couleur peut être "red", "green", etc.)
goto(x, y)	Déplace la tortue jusqu'auau point de coordonnées (x, y)
done( ) ou mainloop( )	Attend que l'utilisateur ferme la fenêtre

Pour commencer à programmer en langage Python, ouvrons tout d'abord l'interface de développement intégrée IDLE. Nous allons écrire notre premier programme graphique en mode "Éditeur" en utilisant les fonctionnalités offertes par le module turtle. Le mode Éditeur est aussi appelé mode Script, un script étant un programme ou code exécuté par l'interpréteur Python, la machine capable de comprendre ou d'interpréter le script.
Une fois ouvert IDLE et activé le mode Éditeur, procéder à la sauvegarde immédiate du contenu vierge de la fenêtre affichée dans un fichier dénommé learning_turtle_26_01_24.py, puis saisir les instructions suivantes :

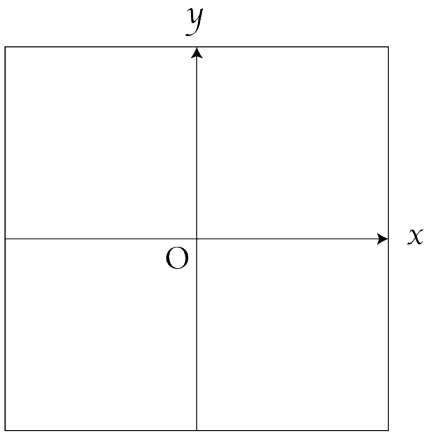
```
#Commentaires : Ce programme est mon premier programme Python avec turtle
#Date : Le vendredi 26 janvier 2024
#Auteur :

from turtle import *    # Il faut importer le module turtle pour pouvoir utiliser ses instructions.
reset( )                # La fonction reset( ) réinitialise la fenêtre de dessin utilisée.
down( )                 # La fonction down( ) abaisse le crayon pour pouvoir dessiner.
done( )                 # La fonction done( ) maintient ouverte la fenêtre contenant le dessin réalisé.
```

Une fois les instructions saisies, sauvegarder ("enregistrer") le programme, puis exécuter le script. Une fenêtre dans laquelle apparaît un curseur (la "tortue") centré s'affiche.

Le script ci-dessus commande la réinitialisation d'une fenêtre d'affichage dont les dimensions définies par défaut sont 950 pixels (largeur) × 800 pixels (hauteur), l'abaissement de la "tortue" (qui sera chargée de dessiner) et le maintien de l'ouverture de la fenêtre.

Pour que nous puissions dessiner un dessin, la fenêtre est dotée d'un repère orthogonal centré $(O; x, y)$ tel que ci-dessous.



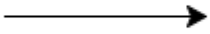
A l'ouverture de la fenêtre via l'instruction reset( ), la tortue se place au centre O du repère et pointe vers la droite.

Déplacement vers l'avant - L'instruction forward( ) ou fd( )

Saisir le script ci-dessous :

```
from turtle import *
reset( )
down( )                 # Cette instruction pose la tortue pour dessiner - turtle.up() relève la tortue.
forward(100)            # La fonction forward(100) commande un déplacement de 100 pixels vers l'avant.
done( )
```

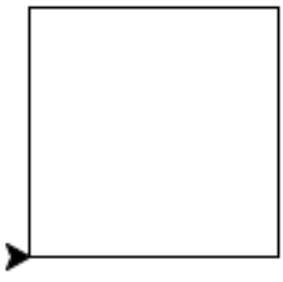
On obtient l'affichage :



Déplacement vers la gauche - L'instruction left( ) ou lt( )

En utilisant les commandes ci-avant, ainsi que l'instruction left( ) qui permet de faire tourner vers la gauche d'un angle choisi la tortue, on peut écrire le script ou programme du tracé d'un carré de côté 100 pixels, comme ci-dessous :

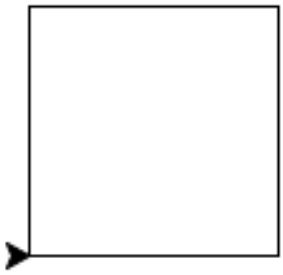
```
from turtle import *
reset( )
down( )
fd(100)
lt(90)
fd(100)
lt(90)
fd(100)
lt(90)
fd(100)
lt(90)
done( )
```



Optimisation - L'instruction for i in range( ):

L'instruction for i in range( ) permet de réitérer l'exécution d'un bloc d'instructions et de réduire le nombre d'instructions nécessaires pour tracer un carré, par exemple.

```
from turtle import *
reset( )
down( )
fd(100)
for i in range(4):
    fd(100)
    lt(90)
turtle.done( )
```



Nous observons que le code écrit est désormais plus concis.

Fonction et tracé

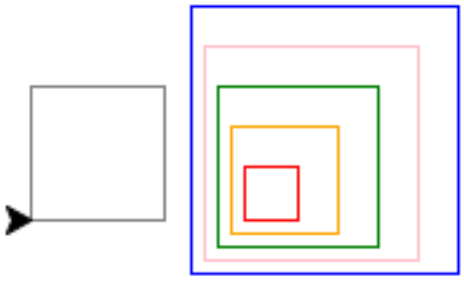
Le tracé d'un carré de dimension donnée peut être dévolu à une fonction que nous appelerons carre( ), comme ci-dessous.

```
from turtle import *
def carre( L):
   for i in range(4):
      fd(L)
      lt(90)
t.reset( )
t.down( )
carre(100 )
turtle.done( )
```

Exercice

1. Créer une fonction carre(L, couleur) qui affiche un carré de longueur L et de couleur la couleur entrée sous la forme "red" ou "blue", par exemple.
2. Créer une figure artistique uniquement à l'aide de carrés colorés.

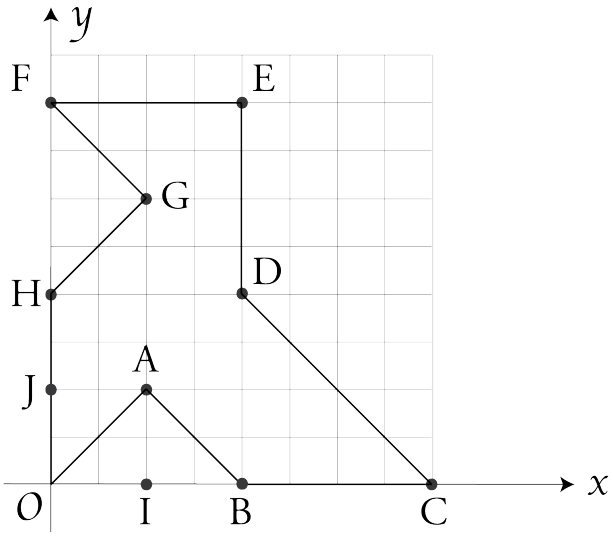
   Exemple :



Activité 2

Partie 1

On considère la cocotte ci-dessous pour laquelle OI = OJ = 50 :

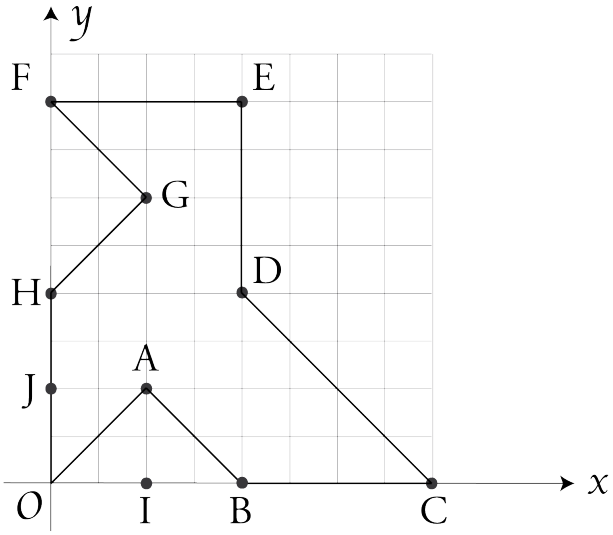


1. Exprimer les coordonnées des sommets A, B, C, D, E, F, G et H ci-dessus.

2. Écrire le code Python du tracé de la cocotte à l'aide de l'instruction goto( ) et enregistrer le code dans le fichier cocotte_50.py.

Partie 2

On considère la cocotte pour laquelle OI = OJ = L :



1. Exprimer les coordonnées des sommets A, B, C, D, E, F, G et H en fonction de la variable L.

2. Écrire en langage Python une fonction cocotte(L) qui réalise le tracé de la cocotte et appeler cette fonction pour tracer trois cocottes cocotte(50), cocotte(40) et cocotte(30) Enregistrer le code dans le fichier 3cocottes.py.

On voudrait réaliser une figure sur laquelle serait répété le motif de la cocotte.
Comment faire ?