Activité 1 : Les figures dynamiques

Dans cette activité, nous allons réaliser de figures dynamiques en utilisant le langage de programmation Python et particulièrement sa bibliothèque turtle. Ce module permet de déplacer un point dans un espace 2D (ce point est souvent vu comme une tortue), et ainsi de réaliser des dessins. Les commandes de bases de la bibliothèque Turtle sont les suivantes :

up() : lève le crayon
down() : baisse le crayon
forward(n) : avance de n
left(d) : tourne vers la gauche de d degrés
right(d) : tourne vers la droite de d degrés
goto(x,y) : se déplace vers le point de coordonnées (x,y)
circle(r) : dessine un cercle de rayon r
width(e) : définit l'épaisseur du trait
speed("texte") : définit la vitesse d'exécution
write("texte") : écrit le texte
color("couleur") : définit la couleur du trait
bgcolor("couleur") : définit la couleur de fond
reset() : efface tout
done() : arrête le dessin

Dans les instructions color() et bgcolor() précédentes, les couleurs suivantes peuvent être choisies : blue, red, black, green, . . .
Pour la vitesse, on peut choisir (du plus rapide au plus lent) : slowest, slow, normal, fast et fastest.
Le début de chaque programme doit faire appel à la bibliothèque de fonctions Turtle en tapant la directive from turtle import *

1. Prise en main du logiciel

➔ **Créer** un fichier nommé first.py et contenant le code ci-contre.
➔ **Exécuter** ce programme et **analyser** le résultat afin d'identifier le rôle de chaque instruction

```
from turtle import *
longueur = 120
angle = 90
reset()
forward(longueur)
left(angle)
color('red')
forward(longueur-40)
done()
```

longueur et angle sont des variables. Elle permettent de définir en début de programme les caractéristiques de la figure souhaitée.
La correction est disponible ici

➔ **Modifier** ce programme afin de tester les fonctions suivantes : right, circle, width et color.

Ma première figure

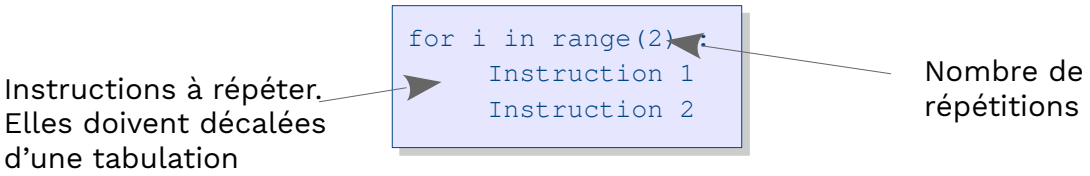
✗ **Dessiner** un carré rouge de côté 100 et un autre jaune deux fois plus petit et centré sur le premier
La correction est disponible ici

✗ **Dessiner** un carré rouge de côté 100 et un cercle jaune de rayon 80 centré sur le carré
La correction est disponible ici

2.Booster ses programmes

Les répétitions

Nous nous rendons compte que le programme précédent comporte plusieurs fois les mêmes instructions. Il est possible de simplifier le code en répétant plusieurs fois le groupe d'instruction à l'aide d'une boucle for. La syntaxe de de cette structure répétitive est la suivante :



En utilisant cette boucle répétitive, le code pour dessiner un carré de 50 de côté devient :

```
from turtle import *
reset()
for i in range(4) :
      forward(50)
      Left(90)
done()
```

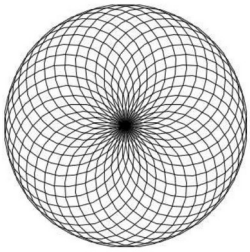
✗ **Dessiner** un triangle équilatéral jaune de 70 de côté en utilisant une boucle for pour répéter 3 fois le dessin d'un côté du triangle

La correction est disponible ici

✗ **Réaliser** le dessin de la figure suivante constituée de dix carrés de côté 20 avec un espace de 5 entre chacun d'eux.

La correction est disponible ici

✗ **Ecrire** un programme réalisant le dessin ci-contre.

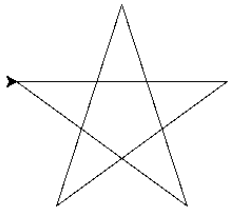


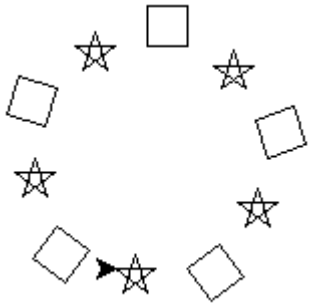
Pour obtenir ce dessin, on peut observer qu'il est constitué de cercles de même rayon (80 dans notre cas), avec un décalage de 10 degrés entre deux cercles successifs (soit 36 cercles au total).

Cette figure est obtenue en dessinant 36 cercles décalés de 10° à leur origine. La correction est disponible ici

3.Pour aller plus loin

✗ **Écrire** un programme, qui trace les figures suivantes :





La correction est disponible ici et ici

4.Jouons avec les couleurs

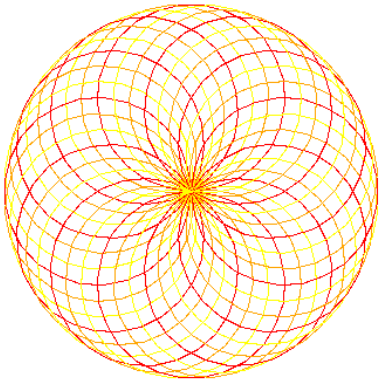
La fonction color() modifie la couleur du tracé en cours. En conditionnant l'appel de cette fonction à certaines conditions il est alors possible d'introduire des couleurs dans les figures.
Des conditions peuvent être implémentées dans le programme avec l'instruction if. Cette instruction exécute une suite d'instructions si l'expression associée est vraie :

Instructions à effectuer. Elles doivent décalées d'une tabulation

```
if expression :
    Instruction 1
    Instruction 2
```

Expression associée de test

✗ **Modifier** les deux programmes précédents afin d'obtenir les figures suivantes :



La correction est disponible ici et ici

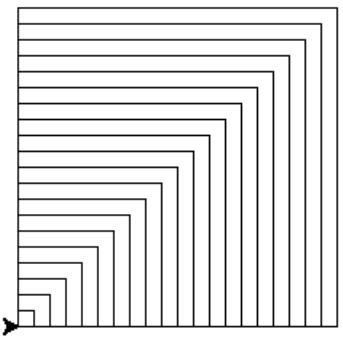
5.Les variables

Les variables sont l'un des concepts qui se retrouvent dans la totalité des langages de programmation. Le principal intérêt d'une variable réside dans le fait qu'il est possible de la faire évoluer durant l'exécution du programme. Par exemple le code ci-contre permet de dessiner deux carrés l'un dans l'autre.

```
from turtle import *
reset()
speed('fastest')
down()
lg = 80
for i in range(2) :
      for j in range(4):
            forward(lg)
            left(90)
      lg = lg + 200
```

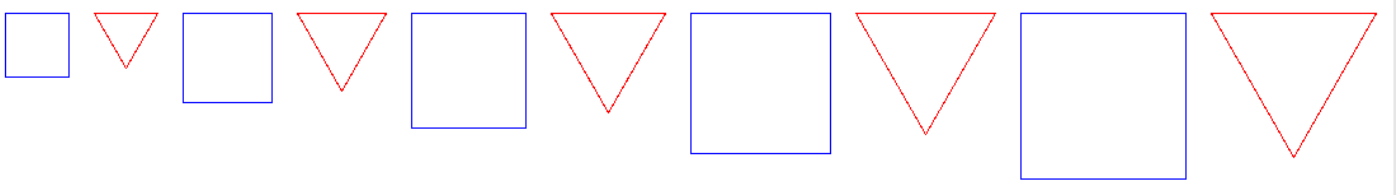
✗ **Analyser** ce code et **déterminer** la longueur des côtés des deux carrés

✗ **Modifier** le programme précédent afin d'obtenir la figure suivante. On peut noter qu'elle est constituée de 20 carrés dont le plus petit à des côtés de 10 et le dernier de 200



La correction est disponible ici

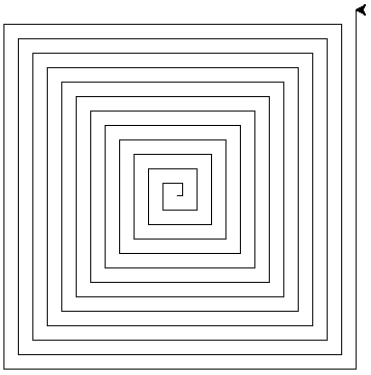
✗ **Écrire** un programme, qui trace un carré puis un triangle, qui grossissent au fur et à mesure.

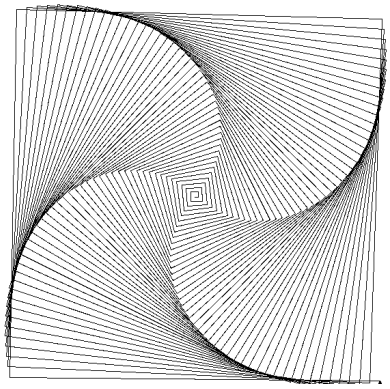


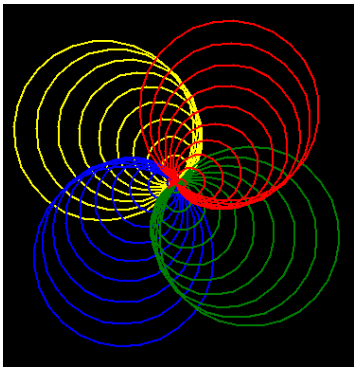
La correction est disponible ici

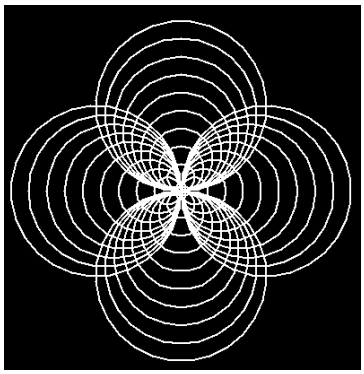
6. A vous de dessiner

✗ Voici des figures dont vous devez trouver le code :









Les corrections sont disponibles ici , ici , ici et ici