

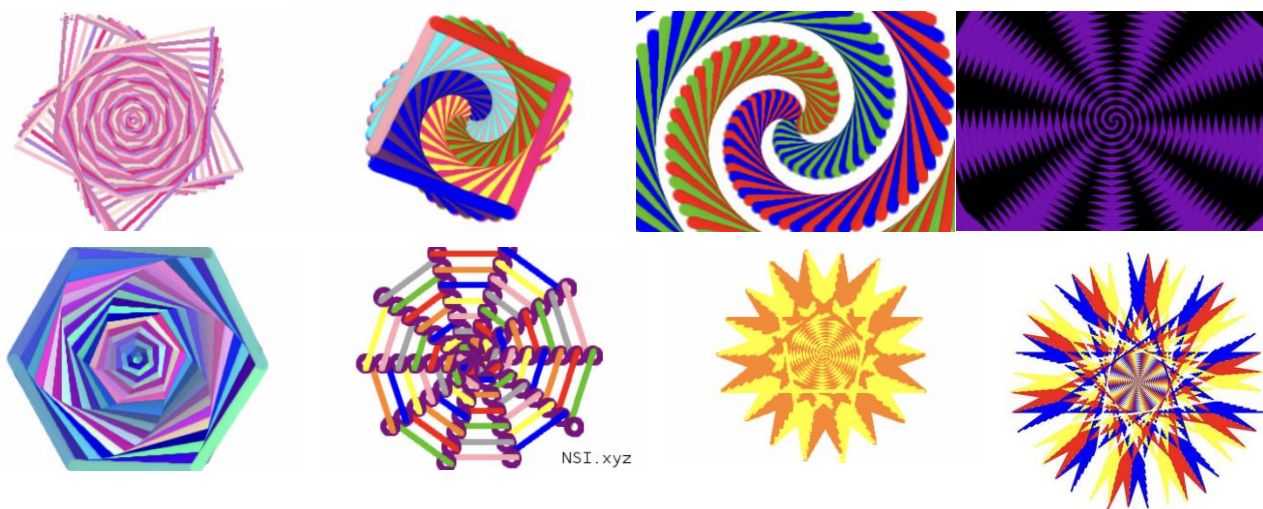
Projet : Le module Turtle

Objectifs :

- Découvrir la méthodologie de projet
- Travailler en équipe : collaboration et répartition des tâches pour travailler individuellement
- Mettre en application les premières séquences (algorithme et programmation Python)

But du projet :

Faire des jolies dessins avec le module Turtle de Python tout en approfondissant l'étude des fonctions et en ~~mettre plein la vue aux~~ faire aussi bien (voir mieux) que les élèves d'un autre lycée dont voici quelques réalisations :



Cahier des charges :

Vous allez donc écrire un programme Python :

- En utilisant des fonctions et le module Turtle
- Qui donnera **au moins un dessin différent** en forme et couleur des autres groupes

Quelques exercices pour vous familiariser avec le module Turtle.

Les instructions de ce langage font se déplacer une tortue munie d'un crayon à la surface d'une feuille virtuelle. On peut alors observer l'exécution du programme à travers les mouvements de la tortue et le tracé qu'elle laisse derrière elle.

Vous pouvez bien sûr vous reporter à la documentation officielle python

: <https://docs.python.org/3/library/turtle.html>

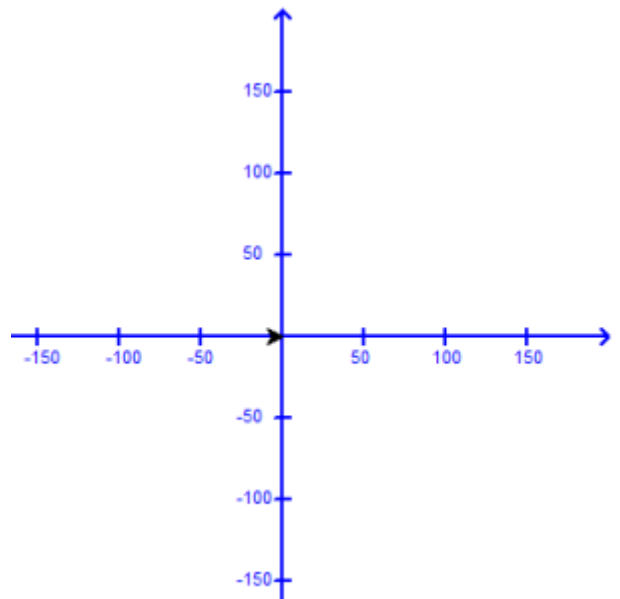


Repère de Turtle

Le module graphique **turtle** permet de piloter un « crayon » permettant de tracer dynamiquement des figures géométriques.

Les dessins sont réalisés dans un repère orthonormé virtuel centré sur la fenêtre d'affichage. L'unité des axes est le pixel. Le repère n'est pas visible à l'écran.

La forme par défaut du crayon de tracé est une flèche « orientée », placée au départ à l'origine du repère. Le crayon est situé à la pointe, la flèche montre le sens du tracé en cours ou à venir.



Les instructions de base

Les instructions de Turtle comprennent en premier lieu des moyens d'orienter et déplacer la tortue dans le plan cartésien.

Instruction	Description
goto(x, y)	Aller au point de coordonnées (x,y)
forward(d)	Avancer de la distance d
backward(d)	Reculer de la distance d
left(a)	Pivoter à gauche de l'angle a
right(a)	Pivoter à droite de l'angle a
circle(r, a)	Tracer un arc de cercle d'angle a et de rayon r
dot(r)	Tracer un point de rayon r
done()	Arrêter le dessin

La tortue commence au point de coordonnées (0,0), situé au centre de l'écran, et est orientée par l'axe des abscisses. L'axe des ordonnées est orienté vers le haut. Les coordonnées sont mesurées en pixels et les angles en degrés. Les arcs de cercles sont parcourus dans le sens trigonométrique si le rayon est positif, et sens horaire si le rayon est négatif.

Application n°1 : Qu'obtient-on ?

```
from turtle import *      # Importation du module
forward(60)               # la tortue avance de 60 pixels
left(120)                 # la tortue tourne sur elle-même de 120° à gauche
forward(60)               # la tortue avance de 60 pixels
right(90)                 # la tortue tourne de 90° à droite
circle(60, 300)           # trace un arc de rayon 60 et d'angle 300°
right(90)                 # la tortue tourne de 90° à droite
```

```
forward(60)      # la tortue avance de 60 pixels
goto(0, 0)       # revient à la position initiale
#done()          # pour pycharm pour que la fenêtre ne se ferme pas
exitonclick()    # ou pour que la fenêtre ne se ferme pas à la fin du dessin
```

Instructions permettant la modification

Les instructions permettant de modifier les dessins produits par chacun des déplacements.

Instruction	Description
up()	Relever le crayon (et interrompre le dessin)
down()	Redescendre le crayon (et reprendre le dessin)
width(e)	Fixer à e l'épaisseur du trait
color(c)	Sélectionner la couleur c pour les traits
begin_fill()	Activer le mode remplissage
end_fill()	Désactiver le mode remplissage
fillcolor(c)	Sélectionner la couleur c pour le remplissage
bgcolor(c)	Définir la couleur c de fond

Par défaut les tracés sont faits en noir avec une épaisseur d'un pixel. Les couleurs peuvent être désignées par leur nom, sous la forme de chaînes de caractères, avec notamment : « black », « white », « grey », « pink », « purple », « blue », « green », « yellow », « orange », « red », « brown ». Pour plus de précision, on peut également fournir un triplet de nombres compris entre 0 et 1 indiquant les niveaux respectifs de rouge, vert et bleu.

Par exemple color(1, 0, 0) correspond au rouge et color(0.3, 0.3, 0.3) correspond au gris foncé.

Application n°2 : Qu'obtient-on ?

```
from turtle import *
# rectangle épais
width(6)      # épaisseur du trait
color(1, 0, 0)  # R max, G = 0, B = 0,
goto(60, 0)
goto(60, 110)
goto(0, 110)
goto(0, 0)
# déplacement
up()
goto(5, 5)
down()
# sablier gris clair
width(1)
fillcolor("grey")
begin_fill()
goto(55, 5)
goto(5, 105)
goto(55, 105)
goto(5, 5)
end_fill()
done()
```

Application n°3 : Qu'obtient-on ?

```
from turtle import *
def spirale_carre(n, increment):
    speed(5)  # paramétrage de la vitesse de 1 lent à 10 rapide, 0 étant la vitesse la plus rapide
    shape("turtle")  # choix de la forme de la tortue
    pencolor("red")  # choix de la couleur du crayon
    pensize(4)  # épaisseur du crayon
```

```

up() # lever le crayon
goto(0, 0) # aller à la position (0,0)
setheading(0) # orientation de la tortue vers l'Est / 90 Nord / 180 Ouest / 270 Sud
down() # poser le crayon
angle = 90 # angle de rotation
cote = 0 # longueur du coté tracé
for k in range(4 * n):
    cote = cote + increment
    forward(cote) # avancer de cote pixels
    left(angle) # tourner de angle degrés

spirale_carre(6, 5) # appel de la fonction pour tracer une spirale carrée
exitonclick() # indispensable en cas d'exécution dans un IDE

```

Application n°4 : Qu'obtient-on ?

```

from turtle import *
def carre(cote):
    for i in range(4):
        forward(cote)
        left(90)
def frise (nbre, cote):
    speed(100)
    for i in range(nbre):
        carre(cote)
        left(5)
frise(10,100)
exitonclick()

```

Instructions modifiant l'action

Quelques instructions permettant de configurer l'action.

Instruction	Description
reset()	Tout effacer et recommencer à zéro
speed(s)	Définir la vitesse s de déplacement de la tortue
title(t)	Donner le titre t à la fenêtre de dessin
ht()	Ne montre plus la tortue (seulement le dessin)
write(s)	Ecrire le texte

Les vitesses possibles sont à choisir parmi les chaînes de caractères « slowest », « slow », « normal », « fast », « fastest » ou parmi les nombres entiers entre 1 (le plus lent) et 10 'le plus rapide. 0 correspond à un déplacement instantané.

A vous de jouer !

Vous avez désormais toutes les cartes en mains pour terminer. Bon codage !!

Travail à faire :

=> Un document numérique avec :

- * La répartition des tâches au sein du **GROUPE DE DEUX** (ou **TROIS**)
- * Une analyse du travail réalisé et des améliorations possibles

=>Un programme python avec :

- * Le code python global => Commentez quasiment chaque ligne du script pour expliquer ce qu'elle fait

PUIS Envoi du dossier compressé (avec les deux fichiers) au format zip ou rar sur ED