

Le module turtle est un ensemble d'outils permettant de dessiner à l'aide d'instructions simples.

1. Exemple

Taper le programme ci-dessous dans l'interpréteur de commande de IDLE, cela permettra de suivre ce qui se passe au fur et à mesure:

```
from turtle import * # il faut importer le module turtle et ses fonctions
forward(120)
left(90)
color('red')
forward(80)
```

2. Fonctions turtle

Les principales fonctions du module turtle sont :

Mouvement de la Tortue:

- forward(d): Avancer d'une distance d (en pixels)
- backward(d): Reculer d'une distance d (en pixels)
- goto(x,y): Positionner la tortue au point de coordonnées (x; y)
- color(couleur): Couleur peut être une chaîne prédéfinie ('red', 'blue', 'green', etc.)
- left(a): Fait pivoter la tortue d'un angle a degrés vers la gauche
- right(a): Fait pivoter la tortue d'un angle de a degrés vers la droite
- circle(r) : trace un cercle de rayon r, le point de départ de la tortue appartient au cercle (attention il n'est pas centré sur la position de la tortue)
- circle(r,s) : trace un arc de cercle correspondant à s degrés
- dot(d,c): dessine un disque de diamètre d et de couleur c là où est la tortue
- setheading(a): où a est en degrés – permet de fixer un cap absolu à la tortue

Contrôle du stylo :

- up() : Releve le crayon (pour pouvoir avancer sans dessiner)
- down(): Abaisse le crayon (pour pouvoir recommencer à dessiner)
- width(épaisseur) : Choisir l'épaisseur du trace (en pixels)
- reset() : nettoie la fenêtre de dessin, réinitialise la tortue ; elle est située alors au centre de l'écran de dessin tournée vers la droite.
- color(c) : la couleur par défaut est le noir, on peut la changer en mettant une couleur prédéfinie c: 'red', 'green', 'blue', 'yellow', . . .
- fillcolor(c) : Remplit une figure fermée à l'aide de la couleur demandée c.
- les balises begin_fill() et end_fill() permettent de commencer et de terminer le remplissage d'une figure géométrique.
- write(texte) texte doit être une chaîne de caractères délimitée avec des " ou des '

<https://docs.python.org/3.2/library/turtle.html>

Exercices d'entraînement au module Turtle

Exercice 1

Tapez le programme suivant dans Wing IDLE, enregistrez-le et exécutez-le :

```
from turtle import *
a=0
while a<12:
    a=a+1
    forward(150)
    left(150)

exitonclick()
```

Exercice 2

- (1) Écrire la fonction `triangle1(a)` qui dessine un triangle équilatéral dont les côtés sont de longueur `a` et qui a la pointe vers le haut.
- (2) Écrire la fonction `triangle2(a)` qui dessine un triangle équilatéral dont les côtés sont de longueur `a` et qui a la pointe vers le bas.
- (3) Écrire la fonction `triangle3(a,angle)` qui dessine un triangle équilatéral dont les côtés sont de longueur `a` et d'une orientation bien déterminées.

Exercice 3

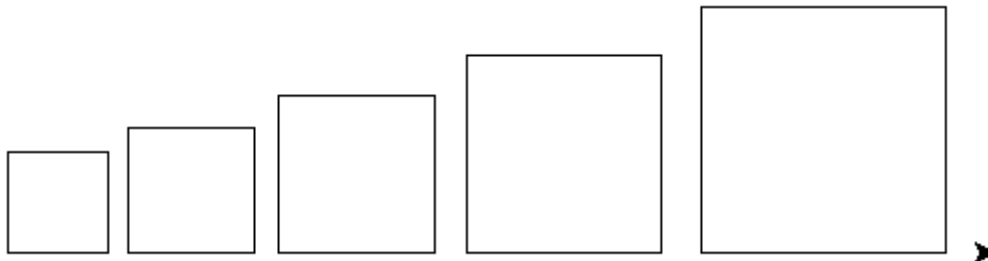
- (1) Écrire la fonction `carre(a)` qui trace un carré de côté `a`. Il est préférable que la tortue termine son dessin là où elle a démarré et avec la même orientation.
- (2) En déduire la fonction `ligneDeCarres(a,n)` qui trace `n` carrés sur une ligne chaque carré étant de côté `a` (on utilisera la fonction `carre`).



- (3) Écrire la fonction `carresCroissants(a,n)` qui trace une ligne de carrés, le premier carré étant de côté `a`, le suivant de taille 1,25 fois la taille du carré qui le précède ; les carrés seront espacés la première fois de `a/4` puis cette distance sera multipliée aussi par 1,25 à chaque fois.

Remarque :

On utilisera la fonction `carre` mais pas `ligneDeCarres`.



Exercice 4

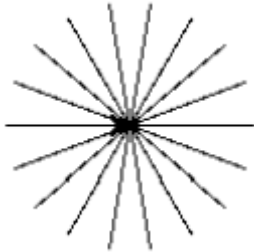
Réaliser le dessin suivant, où le contour de chaque figure est en gras de couleur violette et l'intérieur de chaque figure est de couleur orange.

Le dessin ci-dessous correspond à l'appel de la fonction `figuresPleines(4,20)`, le premier paramètre est le nombre de répétitions et le deuxième correspond la longueur du côté du triangle et du carré.



Exercice 5

(1) Écrire la fonction `rayons(n,d)` qui trace n rayons de longueur d , de même origine et régulièrement espacés ; pour $n=18$ et $d=60$, on obtient le dessin :



(2) Écrire la fonction `polygone(a,n)` qui trace un polygone régulier à n cotés, chaque coté étant de longueur a .

(3) Écrire la fonction `etoile(a,n)` qui trace une étoile avec n sommets, les côtés étant de longueur a .

Pour $n=5$, on obtient l'une des étoiles de la question (4) ci-dessous.

(4) En déduire une fonction pour réaliser le dessin ci-dessous : attention s'il y a un nombre pair d'étoiles, les deux du centre doivent avoir la même taille . . .

