

Cours : Le module turtle

I) Les modules

Nous savons comment écrire une fonction. On peut être amené à utiliser dans plusieurs programmes les mêmes fonctions qui ne sont pas définies dans python à la base.

Par exemple en maths, j'aurais besoin de la fonction cos (mais peut-être aussi de sin, sqrt, exp, log, pi, ...) et je ne vais pas recopier ma fonction dans chaque programme qui en aura besoin.

Pour pallier ce problème, on peut créer une bibliothèque de fonctions (aussi appelée module) appelée math. Un programme nécessitant une fonction mathématique devra commencer par la ligne : `"from math import *"`

Les modules que vous devez connaître :

- math : contient la plupart des fonctions mathématiques usuelles (et pi)
- random : contient des fonctions permettant de générer des nombres aléatoires
- time : contient des fonctions permettant de gérer un chronomètre
- turtle : contient des fonctions permettant de tracer des figures

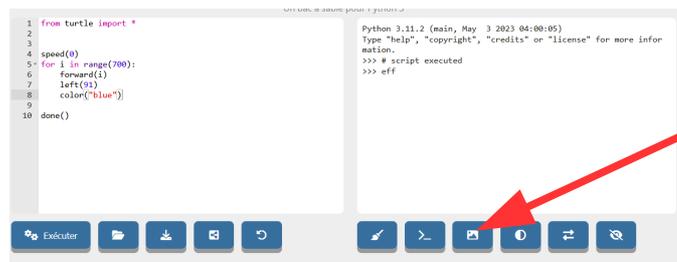
II) Le module Turtle

Le module Turtle permet le tracé de figure à l'aide de commande basique. Un curseur (la turtle) est placé en (0;0), au centre de l'écran, c'est votre crayon et vous le dirigez.

Les principales fonctions du module turtle sont :

- reset() On efface tout et on recommence
- goto(x,y) Aller à l'endroit de coordonnées x et y
- forward(distance) Avancer d'une distance donnée
- backward(distance) Reculer
- up() Relever le crayon (pour pouvoir avancer sans dessiner)
- down() Abaisser le crayon (pour pouvoir recommencer à dessiner)
- color(couleur) Couleur peut être une chaîne prédéfinie ('red', 'blue', etc.)
- left(angle) Tourner à gauche d'un angle donné (exprimé en degré)
- right(angle) Tourner à droite d'un angle donné (exprimé en degré)
- width(épaisseur) Choisir l'épaisseur du tracé
- fillcolor('blue') détermine la couleur de remplissage
- begin_fill() Remplir un contour fermé à l'aide de la couleur sélectionnée
- end_fill()
- write(texte) texte doit être une chaîne de caractères délimitée avec des " ou des '
- speed(0) Accélère au maximum le tracé

Tester par exemple le code suivant sur basthon.fr . Vous le recopiez, vous l'exécutez et vous cliquez sur l'image comme ci-dessous.



```
from turtle import *
speed(0)

for i in range(500):
    forward(i) #on avance de i, donc on avance de plus en plus
    left(91)   #on tourne à gauche de 90°

done()
```

Faites une capture d'écran et collez le dessin obtenu ici :

Effacez tout et testez maintenant ce code :

```
from turtle import *
speed(0)
color('green')
width(3)

for i in range(100):
    forward(100)
    left(50)

done()
```

Faites une capture d'écran et collez le dessin obtenu ici :

Effacez tout et testez ce code :

```
from turtle import *
speed(0)
fillcolor('blue')
width(3)

begin_fill()
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
forward(100)
left(90)
end_fill()
done()
```

Faites une capture d'écran et collez le dessin obtenu ici :

Effacez tout et testez ce code :

```
from turtle import *
speed(0)
fillcolor('blue')
width(3)

begin_fill()
for i in range(36):
    forward(300)
    left(150)

end_fill()
done()
```

Faites une capture d'écran et collez le dessin obtenu ici :

Effacez tout et testez ce dernier code :

```
from turtle import *
speed(0)
width(0)

def star(couleur):
    fillcolor(couleur)
    begin_fill()
    down()
    for i in range(5):
        forward(100)
        left(144)
    up()
    end_fill()

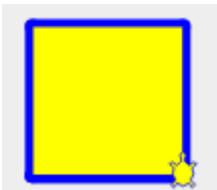
goto(-250,0)
star('blue')
forward(100)
star('red')
forward(100)
star('yellow')
forward(100)
star('green')
done()
```

Faites une capture d'écran et collez le dessin obtenu ici :

Problème 1 :

Tracer un carré jaune de côté 100 pixels avec une bordure bleue d'épaisseur 10 pixels.

Objectif



Programme

Problème 2 :

Ecrire un programme qui dessine 5 carrés de 50 pixels de côtés séparés de 50 pixels rempli de rose et à bordure noire (épaisseur 10 pixels).

Objectif

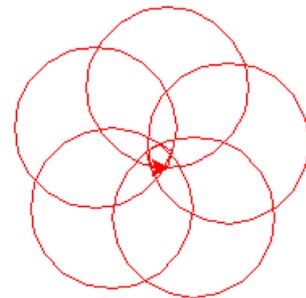
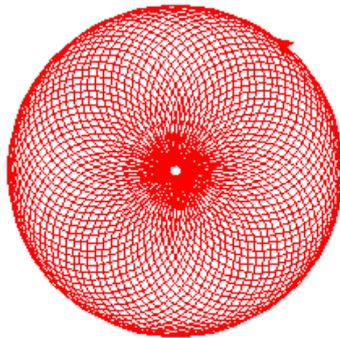
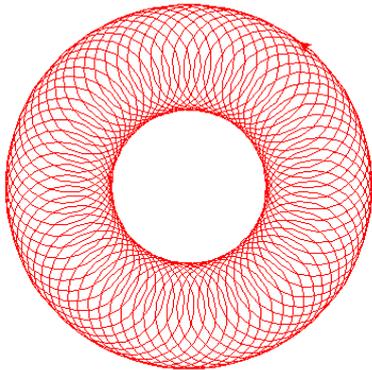
Programme



Problème 3 :

La fonction `circle(50)` trace un cercle de rayon 50 pixels.

En répétant les ordres : " tracer un cercle, tourner, avancer " essayer d'obtenir les dessins suivants :



Problème 4 :

Dessinez cette figure.
Il y a 100 segments de tracés, le premier segment a pour longueur 1,
le second a pour longueur 2, etc ...

Programme

