

Le robot dessinateur

Il existe un module en Python appelé Turtle qui permet, comme dans Scratch, de faire avancer une tortue avec des ordres qu'on lui donne. C'est un robot dessinateur.

Voici un exemple de code :

```
from turtle import *

forward(100)    # avancer des 100 unités graphiques
left(90)       # pivoter de 90° vers la gauche
forward(100)
right(90)      # pivoter de 90° vers la droite
forward(100)

# retour au départ
penup()        # relever le stylo
goto(0,0)      # aller à la position (0,0)
setheading(0)  # prendre la direction (0), vers la droite
```

La flèche indique la position initiale du robot. Au départ elle est à la position (0,0), et son orientation est vers la droite (0°).

A faire : tracer sur une feuille quadrillée le résultat obtenu avec le code donné. On prendra 2 cm pour 100 unités graphiques.

A savoir : on peut utiliser un site web dédié au module Turtle en Python comme : <http://fe.fil.univ-lille1.fr/apl/2018/turtle.html>

Exercice 1

1. Écrire un algorithme qui permet d'obtenir la figure 1.
2. Écrire un algorithme qui permet d'obtenir la figure 2.
3. Proposer la version la plus COURTE possible pour réaliser la figure 2.

Figure 1

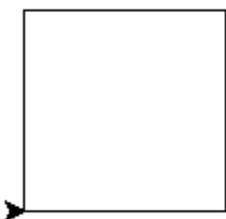
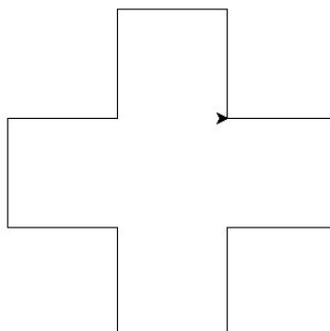


Figure 2



Dans cette fiche, nous allons utiliser comme figure de base le carré, et c'est pour cela que nous allons utiliser une fonction qui dessine un carré d'une longueur choisie.

A savoir : pour définir une fonction en Python, on utilise la commande « def », ensuite on donne le nom de la fonction et après entre parenthèses on donne les paramètres.

```
from turtle import *

def carre_g(long):
    """
    fonction qui dessine un carré
    en tournant à gauche
    """
    forward(long)
    left(90)
    forward(long)
    left(90)
    forward(long)
    left(90)
    forward(long)
    left(90)

carre_g(10)
# carre_g(20)
# carre_g(30)
```

→ Lire attentivement le code ci-contre.

→ Entourer en rouge le code qui correspond au tracé du carré

→ Que fait la fonction « carre_g » ?

→ Une fois le code exécuté, on obtient la figure ci-dessous :



→ Compléter la figure pour visualiser ce qu'on obtient si on enlève le # aux deux dernières lignes.

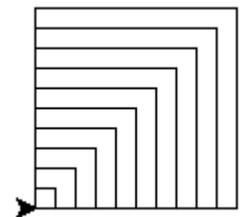
Exercice 1

Écrire une fonction « carre_d » qui prend comme paramètre la longueur de son côté et qui tourne à droite.

Exercice 2

Proposer et tester un algorithme qui utilise la fonction « carre_g » et qui donne la figure ci-contre.

Remarque : le plus petit carré a pour côté 10 « pixels ».



Exercice 3

Analyser chaque figure et proposer un algorithme qui permet de les tracer en utilisant les fonctions « carre_g » et « carre_d ».

