

Use the turtle in the Python

Pour les activités de dessin, on utilisera le module `turtle` de Python qui s'obtient grâce à l'instruction :

```
from turtle import *
```

Au départ, la tortue est au point de coordonnées (0, 0) (au centre de la fenêtre graphique) et elle est orientée à 0° (C'est-à-dire vers l'Est ou à droite).

J'ai utilisé les commandes :

□ <code>reset()</code>	Effacer le dessin
□ <code>goto(x , y)</code>	Aller au point de coordonnées (x, y)
□ <code>forward(distance)</code>	Avancer d'une distance exprimée en pixels
□ <code>backward(distance)</code>	Reculer d'une distance exprimée en pixels
□ <code>up()</code>	Relever le crayon (Pour ne plus dessiner)
□ <code>down()</code>	Abaisser le crayon (Pour dessiner)
□ <code>color(couleur)</code>	Définir la couleur du trait. <couleur> peut être égal à : 'red', 'blue', 'green', 'yellow'...
□ <code>left(angle)</code>	Tourner à gauche d'un angle exprimé en degrés.
□ <code>right(angle)</code>	Tourner à droite d'un angle exprimé en degrés.
□ <code>setheading(angle)</code>	Orienter la tortue à l'angle exprimé en degrés (le 0° étant à l'Est, le 90° étant au Nord)
□ Autres commandes	https://docs.python.org/fr/3/library/turtle.html

Exercice 1 Lien avec Scratch □

1. Exécuter les programmes ci-dessous. Donner avec précision le nom de la figure.

```
from turtle import *

color('blue')
forward(100)
left(90)
forward(200)
left(90)
forward(100)
left(90)
forward(200)
left(90)
done()
```

```
from turtle import *

color('green')
forward(80)
left(142)
forward(150)
left(38)
forward(80)
left(142)
forward(150)
left(38)
exitonclick()
```

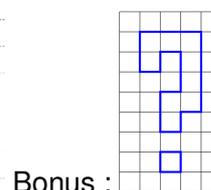
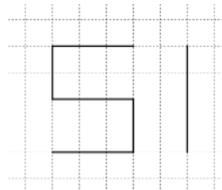
```
from turtle import *

color('red')
left(65)
forward(100)
right(135)
forward(40)
left(135)
forward(90)
done()
```

2. Tracer un rectangle de 150 pixels par 250 pixels.

```
from turtle import *
color('yellow')
begin_fill()                #Remplir
for i in range(36):
    forward(200)            #Respecter les espaces
    left(170)               # au début
end_fill()
done()                       #Autre manière de finir
```

Exercice 2 Reproduire les motifs suivant □



Exercice 3 Escalator

L'objectif de cet exercice est de tracer un escalier de 15 marches. À chaque marche, on monte de 40 pixels puis avance de 50 pixels. On définira le point de départ en bas à gauche de l'écran aux coordonnées (-200; -150).

1. Tracer une marche.

2. Utiliser un bloc d'instruction qui répète les instructions 15 fois.

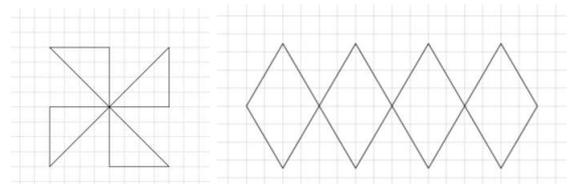
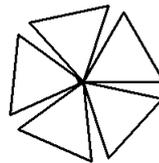
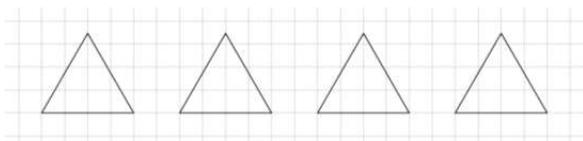
```
for i in range(15):      #Attention au ":"
    .....                #Attention indentation
```

Exercice 4 Polygones

1. Tracer un octogone régulier de côté 100 pixels.
2. Tracer un polygone à 120 côtés en utilisant les commandes `forward(6)` et `left(3)`.
3. Qu'est ce que l'on obtient ?
4. Augmenter la taille du stylo de 1 à chaque étape en utilisant la commande `width(taille)` et la variable `i`.

Exercice 5

1. Dessiner la première frise de quatre triangles équilatéraux :



2. Pour les motifs suivants, réutiliser et modifier le code précédent en utilisant une première fonction `triangle(cote)` qui prend en argument la longueur d'un côté et qui dessine un triangle équilatéral.

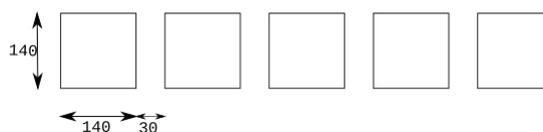
```
from turtle import *

def triangle(cote):
    forward(cote)      #Attention indentation
    .....

triangle(200)
```

Exercice 6

1. Définir une fonction carré qui prend en argument le côté du carré tel que `carré(cote)`.
2. En utilisant une boucle `for`, réaliser le motif ci-dessous.



3. Modifier votre programme pour que les côtés du carré soient augmentés de 15% à chaque étape.
4. Idem pour des triangles équilatéraux de côtés 140 pixels.