

Dessins avec le module "Tortue"

La "tortue" est un triangle mobile sur l'écran graphique, qui peut se déplacer grâce à des instructions *Python* comme :

- `forward(x)` pour que la tortue avance de x pixels
- `left(x)` pour que la tortue tourne de x degrés vers la gauche
- `right(x)` pour que la tortue tourne de x degrés vers la droite

Ces fonctions sont disponibles dans le module *turtle*. Il faut donc importer ce module en début de programme, par l'instruction suivante :

`from turtle import *`

I. Boucle

1) Essayer le script suivant :

```
from turtle import *
reset ( )
forward ( 100 )
left ( 90 )
forward ( 100 )
```

2) Compléter le script pour dessiner un carré de côté 100 pixels.

3) Dessiner un octogone (8 côtés) de côté 60 pixels.

4) Dessiner un polygone de 24 côtés de longueur 50 pixels.

II. Fonction

On veut dessiner plusieurs carrés imbriqués les uns dans les autres. Pour cela, on va **définir une fonction** qui dessine un carré de longueur donnée, et on va **l'appeler** plusieurs fois en spécifiant la longueur du côté voulue.

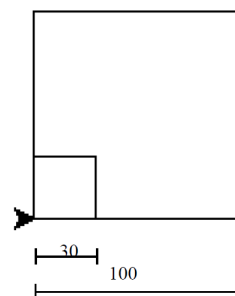
Voici comment déclarer cette fonction :

```
def ma_fonction_carre ( x ) :
    for n in range ( 4 ) :
        forward ( x )
        left ( 90 )
```

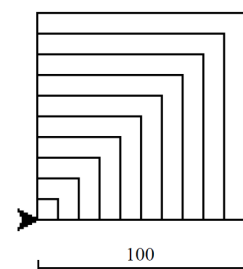
Et voici comment l'appeler (une première fois pour tracer un carré de côté 50 et une 2^e fois pour 60) :

```
ma_fonction_carre ( 50 )
ma_fonction_carre ( 60 )
```

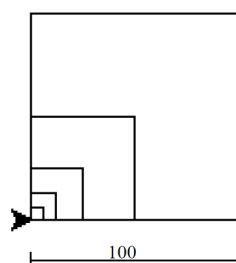
1) A l'aide de cette fonction, dessiner le motif suivant :



2) A l'aide d'une boucle et de cette fonction, dessiner le motif suivant :

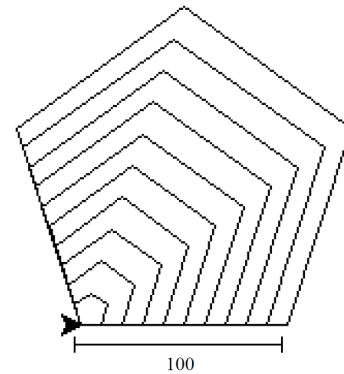


3) dessiner le motif suivant :



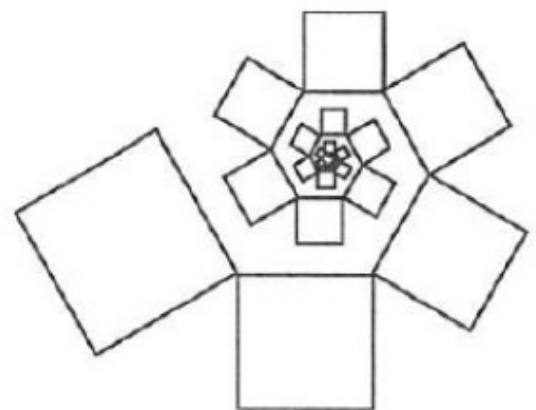
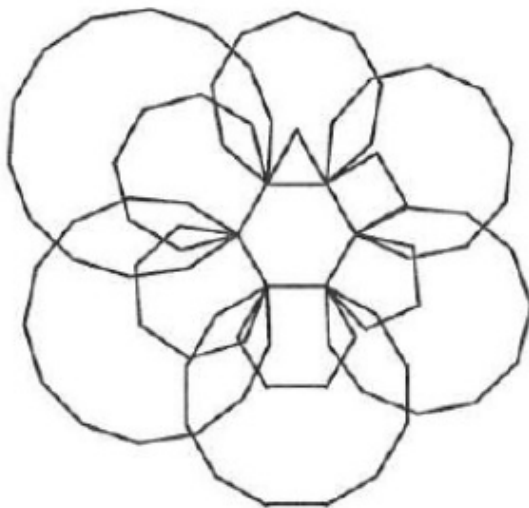
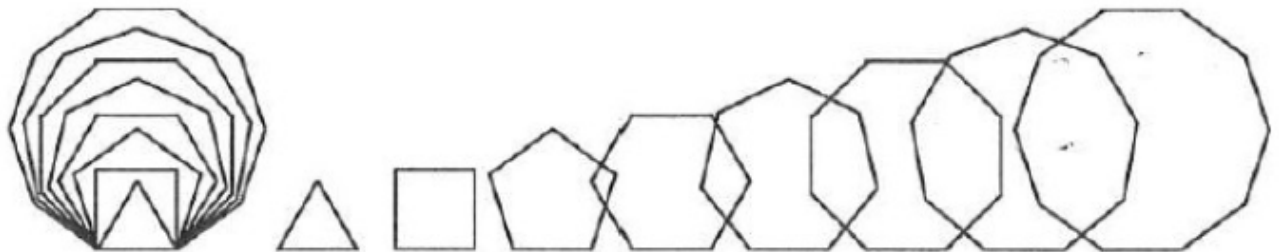
4) Créer une fonction *mon_pentagone(x)* qui dessine un pentagone de côté x donné

5) Utiliser cette fonction pour obtenir le motif suivant :



6) Créer une fonction *mon_polygone(n, l)* qui dessine un polygone régulier de n côtés de longueur l .

Prolongements : créer de nouvelles fonctions permettant de faire les figures ci-dessous :



augmentation de 20%
à chaque carré

Complément de fonctions disponibles avec le module **turtle** :

- `backward(x)` pour que la tortue recule de x pixels
- `reset()` pour effacer l'écran et mettre la tortue au point de coordonnées (0; 0)
- `home()` pour remettre la tortue au point de coordonnées (0; 0)
- `goto(x, y)` pour que la tortue aille au point de coordonnées (x ; y)

- `pendown()` pour que la tortue laisse une trace lors de son déplacement
- `penup()` pour que la tortue ne laisse pas de trace lors de son déplacement
- `position()` pour obtenir les coordonnées de la tortue