

La tortue de Python

En cas de difficultés, pensez à relire la leçon avant de demander de l'aide au professeur.

I. DES SEGMENTS ET DES ANGLES

. EXERCICE 1

a. Écrire l'algorithme correspondant à ce code.

```
from turtle import *
for i in range(3):
    forward(20)
    left(120)
exitonclick()
```

Importer le module turtle pour utiliser la tortue

.....
.....
.....

Fermer la fenêtre lorsque l'on clique dessus

b. Recopier et exécuter le code ci-dessus. Quel affichage obtient-on ?

c. Modifier les lignes 2 et 4 pour obtenir un carré.

d. Ajouter des instructions d'entrée/sortie pour faire choisir la taille du carré à l'utilisateur. Modifier le paramètre de la fonction « **forward** » (ligne 3 ci-dessus) pour que le carré ait la taille choisie.



Les lignes surlignées seront à conserver pour chacun des exercices de ce TP.

. EXERCICE 2

a. Écrire l'algorithme correspondant à ce code.

```
for i in range (5):
    forward (200)
    right (360 * 2/5)
```

.....
.....
.....

b. Recopier et exécuter le code ci-dessus. Quel affichage obtient-on ?

c. Modifier le code pour que l'utilisateur puisse choisir la taille de la figure.

. EXERCICE 3

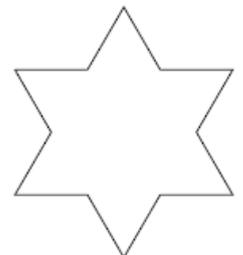
Reproduire l'un des dessins suivants avec Turtle



. EXERCICE 4

a. Écrire un programme Python dessinant une étoile à 6 branches, comme ceci :

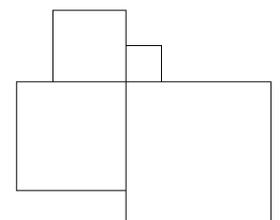
b. Ajouter des instructions d'entrée/sortie pour faire choisir « pleine » ou « vide » à l'utilisateur. Utiliser un test pour colorier l'étoile en bleu s'il a choisi « pleine ».



. EXERCICE 5 (FACULTATIF)

a. Écrire un programme Python dessinant la figure ci-contre.

b. Modifier le code pour que l'utilisateur puisse choisir de colorier la figure. En cas de coloriage, chacun des carrés devra être d'une couleur différente.



II. DES CERCLES, DES DEMI-CERCLES ET DES POLYGONES REGULIERS



Tapez dans la console l'instruction « **help(circle)** » et lisez la page qui s'affiche

. EXERCICE 6

a. Écrire l'algorithme correspondant à ce code.

```
rayon = 1
rayonMax = 100
while(rayon < rayonMax):
    circle(rayon, 180)
    rayon = rayon * 1.5
```

.....
.....
.....

- b. Recopier et exécuter le code ci-dessus. Quel affichage obtient-on ?
- c. Modifier le code pour que l'utilisateur puisse choisir, avant la boucle « **while** », la valeur de la variable « **rayonMax** » ainsi que le nombre décimal par lequel sera multiplié la variable « **rayon** » à chaque passage dans cette boucle « **while** ».

. EXERCICE 7 (FACULTATIF)

a. Écrire l'algorithme correspondant à ce code.

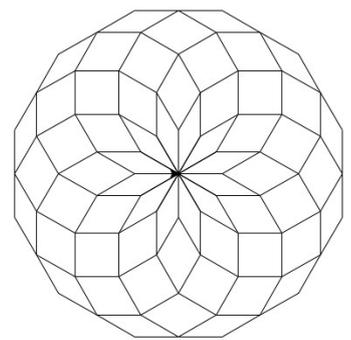
```
for x in range(24):
    circle(60,70)
    if x % 2 == 0:
        circle(10,70)
```

.....
.....
.....

- b. Recopier et exécuter le code ci-dessus. Quel affichage obtient-on ?
- c. Modifier le code pour que l'utilisateur puisse choisir, avant la boucle « **for** », la valeur qui sera passée comme deuxième paramètre lors des appels de la fonction « **circle** » (ligne 2 et 4 du code ci-dessus).
- d. Modifier le code pour que la boucle continue tant que la tortue n'est pas retournée à l'origine c'est-à-dire que « **distance(0,0) > 0.000001** ». Comment faire pour qu'il y ait au moins un passage dans cette boucle ?

. EXERCICE 8

- a. La fonction « **circle** » accepte un troisième paramètre, facultatif. A quoi sert ce paramètre ?
- b. Que fait l'instruction « **circle(100,360,6)** » ?
- c. Quelle instruction permet de tracer un polygone régulier à 12 côtés dont le cercle circonscrit a pour rayon 50 unités ?
- c. Écrire un programme Python dessinant la figure ci-contre.



. EXERCICE 9 (POUR LES PLUS RAPIDES)

Écrire un programme dessinant une spirale ayant l'allure suivante avec un nombre de tours définis par l'utilisateur. Cette spirale est constituée de demi-cercles dont l'épaisseur augmente de 1 à chaque demi-tour et le diamètre est le carré de cette valeur.



III. UN DÉFI : CRÉER UN JEU « CLIQUE TORTUE »

Une tortue, cachée au départ, devra apparaître régulièrement quelques secondes à un emplacement choisi **aléatoirement**.
Un clic sur la tortue rapportera un point.
Le joueur aura gagné lorsqu'il aura atteint 5 points.

Conseils : Lisez l'aide de « **shape** » et « **hideturtle** » du module **turtle** ainsi que de « **randInt** » du module **random** et « **time** » du module **time**.

```
## compteur de clic sur la tortue ##  
compteur = 0  
def marque(x,y):  
    global compteur  
    compteur += 1  
onclick(marque)  
#####
```