

Introduction à Python — Dessiner avec la tortue

Durée : 1h

1 La tortue de Python

Pour activer la tortue de Python, nous allons écrire la ligne suivante tout en haut de notre fichier python; veillez à la garder tout le long de l'exercice :

```
from turtle import *
```

Le but de l'activité sera de diriger le curseur (A.K.A. la tortue) pour tracer les formes demandées.

À titre indicatif, voici les instructions que nous utiliserons :

Pour aller tout droit de 20 pixels :

```
forward(20)
```

Pour tourner sur la gauche de 45° :

```
left(45)
```

2 Triangles, carrés et plus si affinités

Voici un programme Python qui trace deux traits de 100 pixels

EXEMPLE 1

```
forward(50)
left(120)
forward(50)
```

EXERCICE 1

Copiez collez ce programme dans votre éditeur python et exécutez le pour vérifier qu'il trace bien deux traits.
N'oubliez pas la première ligne (import) qui doit être en haut de chaque programme.

EXERCICE 2

Complétez ce programme pour qu'il affiche un triangle.

EXERCICE 3

Modifiez ce programme Python pour qu'il dessine un carré (polygone régulier à 4 côtés).

EXERCICE 4

Une fois que vous arrivez à tracer un carré à l'écran, adaptez le programme pour dessiner un octogone (polygone régulier à 8 côtés).

3 Répétons des instructions

EXEMPLE 2

Vous aurez remarqué que pour tracer des polygones, on copie colle les instructions `forward` et `left`. Pour éviter cette pratique inefficace, nous allons utiliser une boucle pour répéter des instructions. Par exemple, le programme suivant trace un triangle en répétant trois fois les instructions `forward` et `left`.

```
for compteur in range(3):
    forward(100)
    left(120)
```

Pour que des actions soient répétées, il faut qu'elles soient sous la ligne « for » et qu'elles soient précédées de 4 espaces.

EXERCICE 5

Recopiez ce programme et exécutez le. Vérifiez qu'il trace bien un triangle.

EXERCICE 6

Modifiez ce programme pour qu'il trace un carré, puis un hexagone. Quelles modifications faites vous?

4 Des variables

Nous allons essayer de modifier le programme pour qu'il soit capable de tracer n'importe quel polygone. Nous aurons besoin d'une variable, contenant le nombre de côtés du polygone. Le reste du programme tracera ce polygone.

EXEMPLE 3

Ce programme python trace un triangle équilatéral.

```
nb_cotes = 3
for compteur in range(nb_cotes):
    forward(100)
    left(360/nb_cotes)
```

EXERCICE 7

Recopiez ce programme et vérifiez qu'il trace bien un triangle.

EXERCICE 8

Modifiez la valeur stockée dans la variable `nb_cotes` (vous pouvez par exemple mettre 4, 6, 8 ou 180). Que se passe t'il?

5 Écrire un sous-programme

Pour tracer plusieurs polygone, il nous faudrait copier-coller notre programme. Le résultat serait difficile à lire et donc à comprendre... Nous allons créer une fonction (c'est un sous-programme) capable de tracer n'importe quel polygone. Nous pourrons alors exécuter ce sous-programme dès que besoin.

EXEMPLE 4

```
def polygone(nb_cotes):  
    for compteur in range(nb_cotes):  
        forward(100)  
        left(360/nb_cotes)  
polygone(3)  
polygone(5)
```

EXERCICE 9

Recopiez ce programme et vérifiez qu'il trace bien deux polygones. Vous pouvez maintenant en une seule ligne, tracer le polygone de votre choix.

6 Damier (Bonus)

On peut colorier nos polygones à l'aide des fonctions « `begin_fill` » et « `end_fill` ». Voici l'exemple d'un triangle que l'on remplit :

```
begin_fill()  
polygone(4)  
end_fill()
```

EXERCICE 10

Écrivez un programme qui trace un damier de 6 cases par 6 cases. Les couleurs des cases doivent alterner blanc et noir.

Faites vérifier le programme par le professeur

Indices :

- Écrivez une fonction `carre()` traçant un carré et `carreN()` traçant un carré noir. Vous pourrez ainsi construire le damier à l'aide de ces deux fonctions.
- Écrivez une fonction `ligne()` qui trace une ligne de damier.
- Essayez ensuite d'exécuter la fonction `ligne()` pour tracer le damier.