
Présentation du module Turtle

Turtle est un module graphique du langage de programmation Python. Il est inspiré de la programmation *Logo* et permet de déplacer une tortue sur l'écran.

Peu puissant, c'est cependant un outil très intéressant pour un usage pédagogique.

1. Structure minimale

Seul un nombre limité de fonctions est implémenté par défaut dans *Python*. Il est possible d'ajouter des fonctions en faisant appel à des *bibliothèques*, comme nous allons le faire avec la *bibliothèque Turtle*.

Il existe plusieurs façon d'appeler une bibliothèque en Python. La méthode ci-dessous, qui crée l'alias « *tu* » a le mérite d'être la plus polyvalente. Les fonctions de la bibliothèque *Turtle* seront ensuite appelées avec la syntaxe « *tu.fonction(arguments)* ».

Structure du programme :

```
1 import turtle as tu
2
3 # Bloc d'instructions
4
5 tu.mainloop()
```

La ligne finale « *tu.mainloop()* » évite que le programme tourne en boucle.

2. Quelques fonctions disponibles

2.1. Déplacement de la tortue

- *goto(x, y)* déplace la tortue au point de coordonnées (x, y) en pixels.

- *forward(x)* avance la tortue de x pixels.
- *backward(x)* recule la tortue de x pixels.
- *left(x)* tourne la tortue vers la gauche de x degrés.
- *right(x)* tourne la tortue vers la droite de x degrés.
- *circle(r, a)* trace un arc de cercle de rayon *r* pixels et d'un angle *a* degré. Si *r* est positif, l'arc est tracé dans le sens inverse des aiguilles d'une montre ; si *r* est négatif l'arc est tracé dans le sens des aiguilles d'une montre. Enfin, si aucune valeur n'est fournie pour *a*, le cercle complet est tracé.

2.2. Action sur la tortue

- *up()* lève le crayon : les déplacements ne traceront pas de trait.
- *down()* baisse le crayon : les déplacements traceront à nouveau des traits.
- *ht()* cache la tortue. Utile notamment en fin de programme.

2.3. Remplissage d'une zone

- *begin_fill()* débute le remplissage d'une zone, qui sera délimitée par les instructions suivantes.
- *end_fill()* termine le remplissage de la zone
- *fillcolor(couleur)* définit la *couleur* du remplissage. Elle peut être définie soit par des composantes RGB, soit par une chaîne de caractère (par exemple « "blue" »).

2.4. Paramètres du trait

- *width(x)* définit une largeur de x pixels pour le trait. Par défaut, le trait a une largeur de 1 pixel.
- *color(couleur)* définit la *couleur* du trait. Elle peut être définie soit par des composantes RGB, soit par une chaîne de caractère (par exemple « "blue" »).