

Turtle....

Présentation

Le langage contient également de très nombreuses bibliothèques, qui apportent des collections d'instructions plus spécialisées permettant d'effectuer de nouvelles tâches.

Commençons ici par la bibliothèque turtle, qui permet de reproduire les fonctionnalités de base du langage de programmation éducatif Logo. Les instructions de ce langage font se déplacer une tortue munie d'un crayon à la surface d'une feuille virtuelle. On peut alors observer l'exécution du programme à travers les mouvements de la tortue et le tracé qu'elle laisse derrière elle.

Les instructions de turtle comprennent en premier lieu des moyens d'orienter et déplacer la tortue dans le plan cartésien à deux dimensions (en utilisant le repère standard des mathématiques, avec abscisses sur un axe horizontal croissant vers la droite et ordonnées avec axe vertical croissant vers le haut)

Exemples :

<code>goto(x, y)</code>	aller au point de coordonnées (x, y)
<code>forward(d)</code>	avancer de la distance d
<code>backward(d)</code>	reculer de la distance d
<code>left(a)</code>	pivoter à gauche de l'angle a
<code>right(a)</code>	pivoter à droite de l'angle a
<code>circle(r, a)</code>	tracer un arc de cercle d'angle a et de rayon r
<code>dot(r)</code>	tracer un point de rayon r

La tortue commence au point de coordonnées (0, 0), situé au centre de l'écran, et est orientée par l'axe des abscisses. L'axe des ordonnées est orienté vers le haut. Les coordonnées et distances sont mesurées en pixels et les angles en degrés. Les arcs de cercles sont parcourus dans le sens trigonométrique si le rayon est positif, et sens horaire si le rayon est négatif.

Testez le programme suivant :

```
from turtle import *

forward(60)
left(120)
forward(60)
right(90)
circle(60, 300)
right(90)
forward(60)
goto(0, 0)
```

S'ajoutent à cette base une série d'instructions permettant de modifier les dessins produits par chacun des déplacements.

p()	relever le crayon (et interrompre le dessin)
down()	redescendre le crayon (et reprendre le dessin)
width(e)	fixer à e l'épaisseur du trait
color(c)	sélectionner la couleur c pour les traits
begin_fill()	activer le mode remplissage
end_fill()	désactiver le mode remplissage
fillcolor(c)	sélectionner la couleur c pour le remplissage

Par défaut les tracés sont faits en noir avec une épaisseur d'un pixel. Les couleurs peuvent être désignées par leur nom, sous la forme de chaînes de caractères, avec notamment : "black", "white", "grey", "pink", "purple", "blue", "green", "yellow", "orange", "red", "brown". Pour plus de souplesse on peut également fournir un triplet de nombres compris entre 0 et 1 indiquant les niveaux respectifs de rouge, vert et bleu composant la couleur : on obtient du rouge avec color(1, 0, 0), un gris foncé avec color(0.3, 0.3, 0.3), une certaine teinte de violet avec color(0.5, 0, 0.6), etc. Toute l'aire contenue à l'intérieur de la trajectoire de la tortue pendant une période de temps où le mode remplissage est activé prend la couleur choisie pour le remplissage (noir par défaut).

Testez le programme suivants :

```
from turtle import *

# Rectangle épais
width(6)
color(0.2, 0.2, 0.2)
goto(60, 0)
goto(60, 110)
goto(0, 110)
goto(0, 0)

# Déplacement
up()
goto(5, 5)
down()

# Sablier gris clair
width(1)
fillcolor("grey")
begin_fill()
goto(55, 5)
goto(5, 105)
goto(55, 105)
goto(5, 5)
end_fill()
```