

4

Mémo Python

Le module turtle



Le module turtle simule le déplacement d'une tortue virtuelle, dont la trace permet de créer des figures.

Les contrôles

Importation / Syntaxe

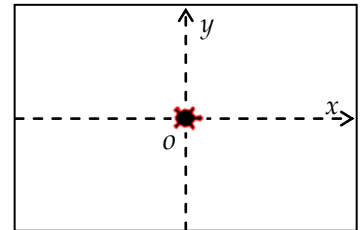
En début de programme : `from turtle import*`

Contrôles standards / Description

`nom_tortue = Turtle()` Création de l'objet tortue permettant la gestion des dessins (tracé).

`nom_toile = Screen()` Création de l'objet zone de dessin (toile) avec son système d'axes. (O au centre du rectangle).

Remarque : Si aucune zone de dessin n'est déclarée, une zone de dessin sera affichée par défaut avec au départ la tortue au centre dirigée vers la droite.



Les méthodes

Les **méthodes** sont les actions qui vont être réalisées sur l'objet créé à l'aide d'un contrôle.

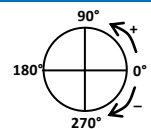
Syntaxe

On applique une méthode à un objet en écrivant selon la syntaxe ci-contre.

`nom_objet.nom_methode(attribut1, attribut2, ...)`
attribut1 : obligatoire
attribut2 : optionnel

Méthodes spécifiques au contrôle Turtle :

| Méthodes / Description | |
|---|---|
| <code>nom_tortue.goto(abscisse, ordonnée)</code> | Déplace la tortue (saut) aux coordonnées indiquées. Selon si le stylo est baissé ou levé un trait est tracé ou non. |
| <code>nom_tortue.right(angle)</code> <code>nom_tortue.left(angle)</code> | Tourne la tortue à droite ou à gauche de l'angle indiqué. |
| <code>nom_tortue.setheading(angle)</code> | Orienté la tortue selon un angle donné. |
| <code>nom_tortue.forward(distance)</code> | Avance la tortue du nombre de points indiqué (distance). |
| <code>nom_tortue.backward(distance)</code> | Reculé la tortue du nombre de points indiqué (distance). |
| <code>nom_tortue.degrees()</code> <code>nom_tortue.radians()</code> | Basculé les valeurs d'angles en degrés ou radians. Par défaut les mesures d'angles sont en degrés. |
| <code>nom_tortue.distance(x, y)</code> | Retourne la distance séparant la tortue du point de coordonnées (x, y) |
| <code>nom_tortue.towards(x, y)</code> | Retourne l'angle que doit avoir la tortue pour aller vers le point de coordonnées (x, y) |



| | |
|--|---|
| <code>nom_tortue.penup()</code> | Lève le crayon (stylo, pinceau) de la tortue qui ne laisse plus de trace lors de ses déplacements. |
| <code>nom_tortue.pendown()</code> | Baisse le crayon (stylo, pinceau) de la tortue et laisse une trace (trait) lors des déplacements de la tortue. |
| <code>nom_tortue.pencolor(couleur)</code> | Définie, ou retourne, la couleur du crayon (trait) de la tortue. On peut saisir "red", "green", "blue", "yellow", "orange", "cyan", ... |
| <code>nom_tortue.pensize(épaisseur)</code> | Définie ou retourne l'épaisseur du trait en pixels. |
| <code>nom_tortue.speed(valeur)</code> | Modifie ou retourne la vitesse de déplacement de la tortue (de 0 à 10). 0 est instantané. |
| <code>nom_tortue.clear()</code> | Efface tous les dessins effectués par la tortue spécifiée. |
| <code>nom_tortue.circle(rayon, partie, côte)</code> | Selon le sens du tracé, la rayon est positif (antihoraire) ou négatif (horaire). partie : nombre compris entre -360° et 360° si mesures en degrés. côte : nombre de côtés si on souhaite un polygone régulier. |
| <code>nom_tortue.dot(diamètre, couleur)</code> | Dessine un point dont on peut préciser le diamètre (en pixels) et la couleur (idem pencolor) |
| <code>nom_tortue.home()</code> | Déplace la tortue au point d'origine et dans l'axe par défaut. |
| <code>nom_tortue.begin_fill()</code> <code>nom_tortue.end_fill()</code> <code>nom_tortue.fillcolor(couleur)</code> | Début une figure qu'il faudra colorier ultérieurement. Termine une figure qu'il faut colorier. Couleur de remplissage de la figure (idem pencolor). |
| <code>nom_tortue.hideturtle()</code> <code>nom_tortue.showturtle()</code> | Cache la tortue. Montre la tortue. |
| <code>nom_tortue.xcor()</code> <code>nom_tortue.ycor()</code> | Retourne l'abscisse de la tortue. Retourne l'ordonnée de la tortue. |
| <code>nom_tortue.position()</code> | Retourne la position (x,y) de la tortue. |
| <code>nom_tortue.stamp()</code> | Imprime une pointe de flèche là où se trouve la tortue. |
| <code>nom_tortue.write("text", attribut1, ...)</code> | La tortue affiche le texte indiqué dans "text" depuis la position où elle se trouve. <u>Attributs :</u> move=True ou move =False : La souris bouge (ou pas) en écrivant le texte. font=('Arial', 12, 'normal') (ou 'bold' ou 'italic') : Le texte est écrit en police Arial, taille 12 et normal align='Left' (ou 'Center' ou 'Right') : Positionne le texte. |

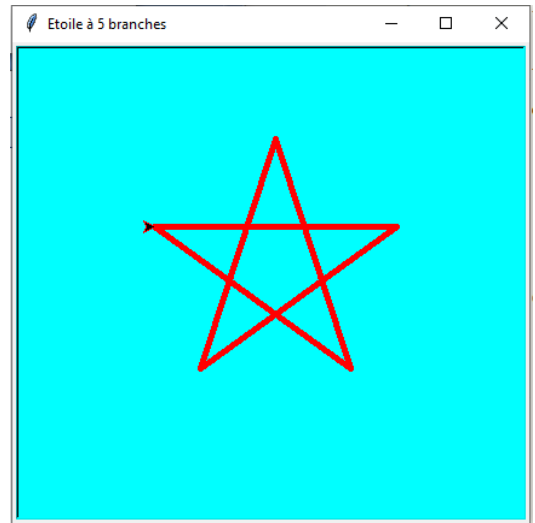
Méthodes spécifiques au contrôle Screen :

| Méthodes / Description | |
|--|--|
| <code>nom_toile.bgcolor(couleur)</code> | Définie, ou retourne, la couleur de fond de la toile. On peut saisir "red", "green", "blue", "yellow", "orange", "cyan", ... |
| <code>nom_toile.clear()</code> | Efface tous les dessins de la toile. |
| <code>nom_toile.exitonclick()</code> | Ferme la fenêtre graphique dès que l'utilisateur clique dessus. |
| <code>nom_toile.setup(largeur, hauteur)</code> | Définie la largeur et la hauteur de la toile qui s'affiche. |
| <code>nom_toile.setworldcoordinates(xb, yb, xh, yh)</code> | Modifie le système d'axes de la toile : (xb,yb), coin en bas à gauche. (xh,yh), coin en haut à droite. Par défaut, l'origine du repère est au centre de la toile. |
| <code>nom_toile.title("texte")</code> | Affiche un texte dans la barre de titre de la fenêtre graphique. |

Exemples

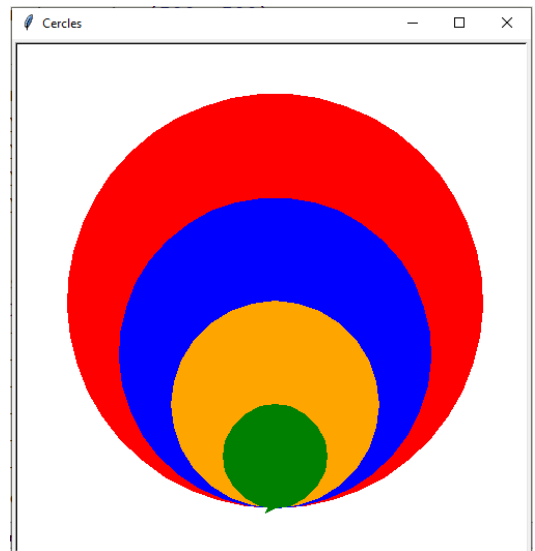
Etoile à 5 branches

```
1 from turtle import *
2
3 # Définition de la tortue appelée Star et de la toile
4 Star = Turtle()
5 Fenetre = Screen()
6 Fenetre.setup(400, 400)
7
8 # Choix du feutre et du fond
9 Star.pencolor("red")
10 Fenetre.bgcolor("cyan")
11 Fenetre.title("Etoile à 5 branches")
12 Star.pensize(5)
13 Star.penup()
14 Star.goto(-100, 50)
15 Star.pendown()
16
17 # Dessin de l'étoile
18 for i in range(5):
19     Star.forward(200)
20     Star.right(144)
```



Cercles colorés

```
1 from turtle import *
2
3 # Définition de la tortue appelée tommy et de la toile
4 tommy = Turtle()
5 Fenetre = Screen()
6 Fenetre.setup(500, 500)
7
8 # Définie le départ de la tortue et les couleurs
9 Fenetre.title("Cercles")
10 tommy.penup()
11 tommy.goto(0, -200)
12 tommy.pendown()
13 tommy.speed(10)
14 color = ["red", "blue", "orange", "green"]
15
16 # Dessin des 4 cercles remplis
17 c = 200
18 for i in range(4):
19     tommy.pencolor(color[i])
20     tommy.begin_fill()
21     tommy.fillcolor(color[i])
22     tommy.circle(c)
23     tommy.end_fill()
24     c = c-50
```



Rosace de cercles et carrés

```
1 from turtle import *
2
3 tommy = Turtle()
4 tommy.speed(11)
5 tommy.pensize(3)
6
7 for i in range(36):
8     for j in range(4):
9         tommy.pencolor('red')
10        tommy.forward(150)
11        tommy.right(90)
12    tommy.color('green')
13    tommy.circle(125)
14    tommy.left(10)
```

