



# TP N°3.0 - Python *Dessin avec turtle*

Thibault Napoléon  
thibault.napoleon@isen-ouest.yncrea.fr

Notions de syntaxe abordées :

- ✓ Configuration de la fenêtre
- ✓ Dessin
- ✓ Couleurs
- ✓ Pointeurs
- ✓ Vitesse
- ✓ Remplissage
- ✓ Fin avec pause

## 1 Prérequis

Pour pouvoir utiliser le module *turtle* avec *Python* il est nécessaire d'installer la bibliothèque d'affichage graphique nommée *Tk*. Pour cela sous linux :

```
sudo apt-get install python3-tk
```

## 2 Dessin avec *turtle*

### Configuration de la fenêtre

Le module s'utilise simplement en l'important et en appelant différentes fonctions précédées du mot-clé *turtle*. En particulier on peut définir la taille de la fenêtre avec *setup(...)*. Définir le fond avec *bgpic(...)* ou encore mettre un titre avec *title(...)*.

```
1 import turtle
2
3 turtle.setup(900, 900)
4 turtle.bgpic('Fond.png')
5 turtle.title('Titre de la fenêtre')
```

### Dessins

Pour dessiner, tout se passe comme avec un crayon. Il est possible de le lever, de le poser et de dessiner... Deux stratégies s'offrent à vous :

- Déplacer votre crayon à une position précise.
- Déplacer votre crayon en lui donnant une série de directions à suivre.

Les instructions à utiliser pour la première solution sont :

```
1 turtle.down()
2 turtle.goto(10, 20) # x, y
3 turtle.up()
```

Les instructions à utiliser pour la seconde solution sont :

```
1 turtle.down()
2 turtle.forward(100) # 100 pixels.
3 turtle.left(90) # 90 degrés.
4 turtle.backward(50) # 50 pixels.
5 turtle.right(45) # 45 degrés.
6 turtle.up()
```

## Couleurs

Lorsque vous dessiner vous pouvez changer la couleur du tracé avec :

```
1 turtle.pencolor('red');
```

## Pointeurs

Différents types de pointeurs existent :

- *blank* : Rien
- *classic* : Curseur classique
- *arrow* : Flèche
- *circle* : Cercle
- *square* : Carré
- *triangle* : Triangle
- *turtle* : Tortue

```
1 turtle.shape('turtle')
```

## Vitesse

Vous pouvez changer la vitesse du tracé :

- *fastest*  $\Leftrightarrow$  0
- *slowest*  $\Leftrightarrow$  1
- *slow*  $\Leftrightarrow$  3
- *normal*  $\Leftrightarrow$  6
- *fast*  $\Leftrightarrow$  10

```
1 turtle.speed('slowest')
```

## Remplissage

Parfois il peut être nécessaire de remplir l'intérieur d'une forme géométrique que l'on vient de tracer. Pour cela on utilise la syntaxe suivante :

```
1 # Dessine un carré vert.
2 turtle.fillcolor('green')
3 turtle.begin_fill()
4 turtle.down()
5 for i in range(0n 4):
6     turtle.forward(10)
7     turtle.left(90)
8 turtle.up()
9 turtle.end_fill()
```

### Fin avec pause

Finalement, pour laisser votre dessin affiché tant que l'utilisateur n'a pas fermé la fenêtre ou cliqué à l'intérieur, vous pouvez utiliser :

```
1 turtle.exitonclick()
```