

# Introduction au langage Python

11 octobre 2024

Les exercices marqués par (\*) sont à faire à la maison.

## 1 Dessin d'images

Dans cette partie, on va utiliser le module `turtle` pour faire des dessins (un module est un ensemble de fonctions pré-écrites). La documentation complète de `turtle` est disponible [ici](#). Commencer avec le fichier `tortue.py` fourni. Il donne un squelette des fonctions à écrire dans le TP. Les fonctions qu'il contient ne font, pour la plupart rien, car elles ne contiennent que l'instruction Python `pass`, qui ne fait rien.

Sauvegardez ce fichier sur votre compte et ouvrez-le sous **Python** en lançant **Thonny** depuis le menu Applications / Développement en haut à gauche de l'écran. Sous **Thonny**, tapez Control-O pour ouvrir le fichier `tortue.py` sauvé. C'est sur ce fichier que l'on travaillera.

On peut ensuite déplacer une « tortue », qui laisse une trace de crayon lors de ses déplacements. On peut ainsi créer des figures variées. Les fonctions les plus utiles pour dessiner avec cette tortue sont les suivantes :

<code>reset()</code>	Efface l'écran, repositionne la tortue au centre avec sa vitesse par défaut.
<code>goto(x,y)</code>	Va au point de coordonnées (x, y).
<code>forward(d)</code>	Avance de la distance d, suivant la direction actuelle de la tortue.
<code>up()</code>	Lève le crayon, pour qu'un déplacement de la tortue ne laisse pas de trace.
<code>down()</code>	Abaisse le crayon, pour qu'un déplacement de la tortue laisse une trace.
<code>pencolor(r,g,b)</code>	Change la couleur du crayon : r (rouge), g (vert) et b (bleu) sont entre 0 et 1.
<code>setheading(angle)</code>	Positionne la tête de la tortue selon l'angle indiqué (0 correspond à l'Est).
<code>left(angle)</code>	Tourne à gauche de l'angle indiqué.
<code>right(angle)</code>	Tourne à droite de l'angle indiqué.
<code>width(e)</code>	Change l'épaisseur du crayon à e.
<code>speed(vitesse)</code>	Ajuste la vitesse de la tortue, par exemple 'fastest' ou 'normal'.

Dans le squelette de programme fourni dans le fichier `tortue.py`, les fonctions demandées sont vides : il ne reste plus qu'à les écrire...

### Exercice 1

Dessins avec turtle

1. Préliminaire : écrire une suite d'instructions pour tracer un carré dont les côtés ont taille 100.
2. Écrire une fonction `carre_100()` qui dessine un carré dont les côtés ont taille 100.

3. Écrire une fonction `carre(taille)` qui dessine un carré dont les côtés ont la taille donnée en paramètre.
4. Écrire une fonction `polygone(taille, cotes)` qui dessine un polygone régulier dont la taille et le nombre de côtés sont donnés en paramètres.
5. Tester la fonction `polygone` en écrivant une fonction `triangle(taille)`, et... en dessinant un *cercle* grâce à la fonction `polygone`.
6. Écrire une fonction `rosace(taille, cote)` qui dessine une rosace en dessinant plusieurs polygones dont la taille et le côté sont donnés en paramètres. Le premier côté du  $k$ -ième polygone dessiné sera incliné d'un angle  $10k$ , et on fera varier cet angle de 0 à 350 degrés, par pas de 10 degrés. On dessine donc au total 36 polygones, inclinés de 10 degrés en 10 degrés.
7. (\*) Écrire une fonction `rosace_coloree(taille, cote)` qui fait la même chose que la fonction précédente, mais qui en plus change progressivement la couleur des polygones dessinés. On part d'une première couleur, par exemple rouge (noté 1,0,0 dans le codage r,g,b) pour arriver à une couleur de fin, par exemple bleu (noté 0,0,1). À chaque nouveau polygone, on modifie progressivement et uniformément la couleur. Par exemple, `rosace_coloree(100,5)` donne ce dessin :

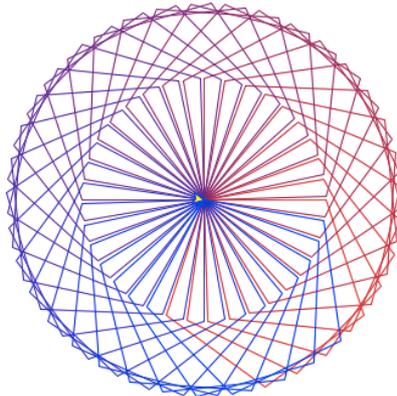


FIGURE 1 – Une rosace colorée créée par juxtaposition de 36 pentagones.

8. (\*) Écrire une fonction `spirale(n, d, a)` qui trace une spirale selon le motif suivant :
  - on fait successivement des segments de longueur  $d$ ,  $2d$ ,  $3d$ ,  $4d$ , etc. jusqu'à  $n \cdot d$ .
  - entre 2 tels segments, la tortue tourne de l'angle  $a$ .
 Tester la fonction avec plusieurs valeurs des paramètres, par exemple  $d=2$  et  $a=90$ , puis  $a=89$ .

## 2 Deux exercices pour vous entraîner en langage Python

### Exercice 2

(\*) Récupérez le fichier `start.py`. Ouvrez-le avec Thonny et exécutez-le pas à pas (avec F5). Expliquez les résultats obtenus.

### Exercice 3

(\*) Récupérez le fichier `erreurs.py`. Ce fichier contient un certain nombre d'erreurs très courantes. Corrigez-les!