

Projet 2 : Python et Turtle

1 Modalités

1. Ce projet est un **travail individuel** ;
2. Vous devrez écrire un **programme en Python** :
 - le code devra tenir compte des **bonnes pratiques** de programmation attendues pour ce langage.
(voir : https://python.sdv.univ-paris-diderot.fr/15_bonnes_pratiques/)
 - le code devra être **commenté** ;
3. Le fichier à rendre sera nommé **Projet2_nom.py** et devra être envoyé par mail, au plus tard le **mercredi 11 Décembre** à minuit, à l'adresse **nathalie.bessonnet@ac-orleans-tours.fr**.
4. Vous pouvez à tout moment **demandeur de l'aide** en posant vos questions en classe ou en m'envoyant un mail.

2 Sujet

On va importer le module **Turtle** qui permet d'utiliser une **interface graphique** avec Python.

1. Découverte de Turtle :

- Recopiez le code et observez ce qui est tracé dans la fenêtre.

```
0 #On importe le module Turtle
1 import turtle
2
3 #Voici les commandes de base
4 turtle.fillcolor("green") #Couleur de remplissage
5 turtle.begin_fill()      #Commencer a remplir
6
7 turtle.forward(60)       #Avancer de 60
8 turtle.left(120)         #Tourner a gauche d'un angle = 120
9                           #degrés
9 turtle.forward(60)
10 turtle.right(100)        #Tourner a droite
11 turtle.circle(60,320)    #Tracer un arc de cercle : rayon=60
12                           #angle=320 degrés
12 turtle.right(100)
13 turtle.forward(60)
14 turtle.left(120)
15 turtle.goto(0,0)        #Se déplacer jusqu'au point de
16                           #coordonnées (0;0)
16
17 turtle.end_fill()        #Fin du remplissage
18
19 turtle.exitonclick()     #Fermer la fenetre si on clique dessus
```

- Voici quelques commandes complémentaires :

Commande	Description
<code>title("texte")</code>	placer un titre sur la fenêtre graphique
<code>setup(largeur, hauteur, posx, posy)</code>	taille et position de la fenêtre
<code>speed("slowest")</code>	vitesse du tracé : entre 1 et 10 ou "slowest", "normal", "fast", "fastest"
<code>width(L)</code>	largeur du trait égale à L
<code>color("bleu")</code> <code>color((0.5,1,0.2))</code>	couleur du trait, en anglais, ou code RGB (Red, Green, Blue) avec des valeurs entre 0 et 1
<code>bgcolor()</code>	couleur de fond pour la fenêtre
<code>up()</code>	lever le crayon
<code>down()</code>	poser le crayon
<code>dot(r)</code>	tracer un point de rayon r
<code>reset()</code>	effacer le dessin
<code>ht()</code>	cache la tortue (le curseur)

2. Créez une fonction **carre()** qui dessine un carré, en demandant les coordonnées du coin inférieur gauche, et la longueur de son côté. Vous devrez utiliser une boucle (for ou while).

Ajoutez une couleur pour remplir le carré, et une couleur pour le trait (différente).

3. Créez une fonction **etoile()** qui trace une étoile, régulière ou non, avec autant de branches que vous voulez. La taille et les coordonnées du point de départ sont données en paramètres. Choisissez-lui une couleur, et utilisez une boucle dans le code.

4. Créez une fonction **liste_alea()**, qui prend en paramètre un entier positif n et retourne une liste constituée au hasard de 0 et de 1 et ayant pour longueur n .

5. Écrivez un programme utilisant les trois fonctions précédentes et permettant d'obtenir un dessin comportant des étoiles et des carrés de taille aléatoire, placées au hasard dans la fenêtre.

On demandera à l'utilisateur le nombre d'objets qu'il faut dessiner ;

On utilisera la fonction **liste_alea()** pour obtenir une liste constituée de 0 et de 1 : en lisant cette liste, on choisira si l'on doit tracer un carré (lorsque l'élément est égal à 0) ou si l'on doit tracer une étoile (lorsque l'élément est égal à 1).