DESSINER EN PYTHON avec TURTLE

Le module Turtle permet d’accéder au codage en python sous une forme graphique très visible. On commande simplement un(plusieurs) crayon(s) qui trace(ent) des lignes.

L’exécution des scripts étant assez lente (et réglable), on peut en particulier suivre le déroulement progressif du programme que l’on vient d’écrire.

**L’objet de cette première activité est :**

* **découvrir (ou redécouvrir) la syntaxe du codage en Python**
* **découvrir (ou redécouvrir) quelques structures de code très classiques : notion de fonction et notion de boucles**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACTIVITE 1 - Rédiger un programme qui**

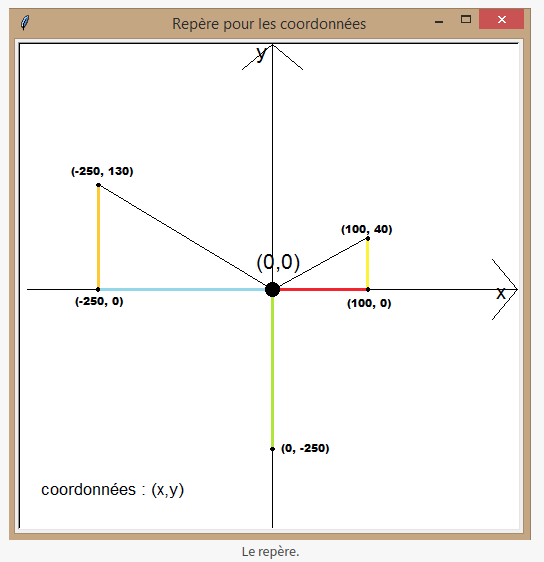
* ouvre une fenêtre de 800 x 800
* trace un carré de 100 x 100 parfaitement centré dans la fenêtre

Chaque ligne du programme sera commentée

INFORMATIONS pour l’activité :

* l’instruction **import turtle** doit figurer en début de programme (module à prendre en compte)
* crayon1 = turtle.Turtle() crée un objet “dessinant” qui sera piloté ensuite (autre nom possible), on peut créer plusieurs objets.
* les fonctions du tableau doc2 s’utilisent ensuite objet par objet  
  exemple : crayon1.fd(10) pour avancer de 10 le **crayon1**
* l’instruction **time.sleep(n)** stoppe toute activité durant n secondes

doc 1 : Système de coordonnées pour commander vos crayons (exemple)



Doc 2 : **Liste (non exhaustive) des fonctions du module Turtle**

|  |  |
| --- | --- |
| fd(n) | avance de n |
| bk(n) | recule de n |
| rt(n) | tourne à droite de n degrés |
| lt(n) | tourne à gauche de n degrés |
| clear() | efface l'écran |
| penup() | lève le stylo |
| pendown() | baisse le stylo |
| reset() | efface l'écran, remet la tortue au centre et réinitialise ses paramètres |
| showturtle() | montre la tortue |
| hideturtle() | cache la tortue |
| speed(n) | Change la vitesse de 1(lent) à 10 (rapide). La valeur spéciale 0 est la plus rapide. |
| tracer(n,d) |  |
| update() | Force l'affichage des graphismes en attente |
| bye() | Referme la fenêtre |
| setup(w,h) | Ouvre une fenêtre de taille wxh |

**ACTIVITE 2 - A partir du 1er programme, rédiger un nouveau programme qui**

* ouvre une fenêtre de 800 x 800
* trace 2 carrés de 100 x 100 assez éloignés pour qu’ils ne se “croisent” pas et de couleur différente

INFORMATIONS pour l’activité :

* crayon1.color(green) (ou red, yellow, blue…) est la méthode pour définir des couleurs

**ACTIVITE 3 - A partir du programme précédent, rédiger un nouveau programme qui**

* ouvre une fenêtre de 800 x 800
* trace 10 carrés de 50 x 50 assez éloignés pour qu’ils ne se “croisent” pas et de couleur différente

INFORMATIONS pour l’activité :

* Il faut que le programme fasse 10 fois la même chose mais en commençant à 10 endroits différents. Demander au professeur une astuce pour réduire le nombre de ligne du programme (notion de fonction)

**ACTIVITE 4 - A partir du programme précédent, rédiger un nouveau programme qui**

* ouvre une fenêtre de 1500 x 800
* trace 30 carrés de dimension croissante sur une ligne haute
* trace 30 hexagones de dimension décroissante sur une ligne basse

INFORMATIONS pour l’activité :

* une deuxième fonction doit être définie (hexagone)
* investiguer la notion de boucle