

Activité Python : dessiner avec le module graphique Turtle

Présentation

Turtle est un des **modules graphiques** disponible pour le langage Python. Il est inspiré de la programmation Logo et permet de déplacer une tortue sur l'écran. Il est alors possible d'effectuer très simplement des **dessins constitués de lignes** en déplaçant une **tortue** dans le plan.

Pour cette activité, tu vas utiliser :

- soit **EduPython**,
- soit le site Web **Trinket** : <https://trinket.io/>

Découvrir l'utilisation de Trinket

The screenshot shows the Trinket online Python IDE interface. The code editor contains a file named 'main.py' with the following code:

```
1 print("Bonjour")
2
```

The output console on the right displays 'Bonjour'.

Annotations with red arrows point to various UI elements:

- Utilise ce menu pour passer au mode plein écran**: Points to the hamburger menu icon in the top left.
- nom de ton script**: Points to the file name 'main.py' in the editor header.
- Choisir le mode console**: Points to the dropdown arrow in the top toolbar.
- Exécuter ton script**: Points to the play button in the top toolbar.
- Partager ou télécharger ton script**: Points to the share and download icons in the top toolbar.

Other visible elements include 'Draft Saved', a 'Remix' button, and the text 'Powered by trinket'.

Utiliser Turtle

Pour pouvoir utiliser **Turtle** dans un programme Python, il faut **importer le module Turtle**. Au début de ton programme Python ajoute la ligne suivante :

```
from turtle import *
```

Les fonctions disponibles

Début du programme

Fonction	Description
<code>reset()</code>	Efface l'écran, recentre la tortue et remet les variables à zéro

Les 4 commandes de base

Fonction	Description
<code>forward(n)</code>	Avance de n pixels
<code>backward(n)</code>	Reculé de n pixels

Fonction	Description
left(r)	Pivote vers la gauche de r degrés
right(r)	Pivote vers la droite de r degrés

Pour modifier les traits dessinés

Fonction	Description
width(n)	Définit la largeur du trait
color("blue")	Définit la couleur du dessin, ici bleu
bgcolor("red")	Définit la couleur du fond

Autres fonctions bien utiles

Fonction	Description
up()	Lève le crayon pour avancer sans dessiner
down()	Baisse le crayon pour dessiner
goto(x,y)	Positionne le crayon en (x,y), (0,0) est le centre de l'écran
circle(r,a)	Dessine un cercle de rayon r sur un arc d'angle a
speed(vitesse)	Définit la vitesse de déplacement ("slowest", "slow", "normal", "fast", "fastest")
write("texte")	Ecrit le texte là où se trouve le curseur
pos()	Renvoie un tuple (x,y) de la position de la tortue

Pour en savoir plus :

- https://fr.wikibooks.org/wiki/Programmation_Python/Turtle

Exemple de dessin d'un carré

```
<html>
```

```
</html>
```

Activité de découverte à faire : dessiner une fleur

Crée un programme python qui dessine une fleur dont chaque pétale est un losange :

Dans cet exemple :

- chaque **ligne** a une longueur de **60 pixels**,
- les **angles** utilisés sont soit de **30 degrés** soit de **150 degrés**.

```
<html>
```

```
</html>
```

Le projet à réaliser

Tu vas écrire un programme Python qui réaliser une **oeuvre graphique interactive** basée sur la **répétition** de **motifs colorés**.

Quelques pistes à explorer ...

Pour faire **varier** les motifs, tu peux changer leur **couleur**, leur **dimension** en mettant ces informations en paramètre dans des **variables**.

Pour placer tes motifs à différents endroits :

- utilise la fonction **random()** pour définir les **positions x et y**.

Pour choisir tes motifs tu peux aussi demander à l'utilisateur :

- les motifs à dessiner,
- leur nombre, leur couleur.

Autres exemples de motif

Indice : l'instruction **circle(rayon, angle)** trace un arc de cercle de rayon et d'angle donné à partir de la position actuelle de la tortue. L'instruction **write(v)** affiche le contenu de la variable **v**.

Indice : le premier carré ABCD est un carré de côté 100 pixels ; Chaque nouveau carré est construit en déplaçant chaque sommet de 10 pixels.

Utilisations des fonctions

L'utilisation de **fonction** facilite la création de plusieurs fleurs. Pour cela, tu peux créer et utiliser une fonction **dessine_fleur()** qui dessine une fleur avec pour paramètre :

- la **position** de la fleur,
- le **nombre** de pétales,
- la **longueur** de chaque ligne de la fleur ;

Les activités ...

[Je reviens à la liste des activités.](#)

From:

/ - **Les cours du BTS SIO**

Permanent link:

[/doku.php/icn/facultatif/a_python_turtle?rev=1568836747](#)

Last update: **2019/09/18 21:59**

