



Unité 2 : Module Turtle et Python

Compétence 2 : Couleur et déplacement

Dans cette leçon, vous allez poursuivre la découverte du module Turtle en utilisant les options de dessin et de déplacement de la tortue du module Turtle.

Objectifs :

- Créer un triangle isocèle et le remplir de couleur.
- Colorier un disque.
- Assembler ces deux objectifs pour créer un dessin.

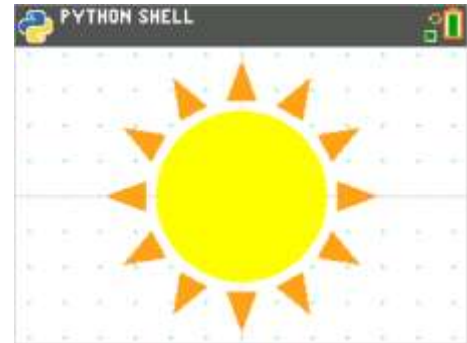
Comment réaliser la figure ci-contre à l'aide d'un script Python en utilisant le module Turtle ?

Cela semble relativement simple, mais qu'en est-il vraiment ?

On observe que la figure est constituée de :

- un disque de couleur jaune centré sur l'origine du repère.
- une série de triangles isocèles équirépartis autour du disque.

Avant de fournir en complément de cette leçon quelques éléments permettant de réaliser le soleil, réalisons séparément, chaque élément.



A : Tracé d'un triangle isocèle

- Commencer un nouveau script en le nommant TRIANGLE.
- Importer la librairie de fonctions mathématiques (utile pour les calculs de trigonométrie) et la librairie Turtle.
- Créer la variable c correspondant à la longueur de la base du triangle. ($c = 60$) correspond à 60 pixels.
- Créer la variable b exprimant en radian l'angle de 70° .
- Créer la variable h , hauteur du triangle isocèle.

```

ÉDITEUR : TRIANGLE
LIGNE DU SCRIPT 0001
from math import *
from turtle import *
t=Turtle()
c=60
b=pi*70/180
h=c/(2*cos(b))

```

- Appuyer sur la touche **f1 [Fns...]** puis ← afin d'accéder au menu **Modul**.
- Choisir **8:turtle...** et dans le sous menu **Settings**, choisir le menu **5:t.speed()** et donner le paramètre de votre choix entre parenthèses (de 0 à 10).
- A partir du sous menu **Pen** et **Draw** du module Turtle, copier les instructions **t.pencolor(r,v,b)**, **t.fillcolor(r,v,b)** permettant de choisir en code R, V, B (rouge, vert, bleu) la couleur du crayon ainsi que celle qui est utilisée pour remplir le triangle.

```

ÉDITEUR : TRIANGLE
LIGNE DU SCRIPT 0012
from turtle import *
t=Turtle()
c=60
b=pi*70/180
h=c/(2*cos(b))
t.speed(10)
t.pencolor(251,163,26)
t.fillcolor(251,163,26)
t.begin_fill()

t.end_fill()

```

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



- Copier également les instructions `t.begin_fill()` et `t.end_fill()` permettant d'établir les instants entre lesquels la méthode de remplissage débute puis se termine.

Conseil à l'enseignant : t.speed(0-10) – spécifie onze niveaux de vitesse de la tortue allant de 1 (lent) à 10 (rapide). La vitesse zéro (0) est le réglage le plus rapide. La valeur par défaut est 5, lorsque `t.speed()` n'est pas spécifié.

- Dans le menu **Move** du module Turtle, recopier les instructions `t.forward()` et `t.left()` afin de compléter la construction du triangle.
- Par souci d'esthétisme, masquer la tortue à l'aide de l'instruction `t.hideturtle()` située dans le menu **5 Settings**.
- Ne pas oublier l'instruction `t.done()` pour afficher le dessin à la fin du script.

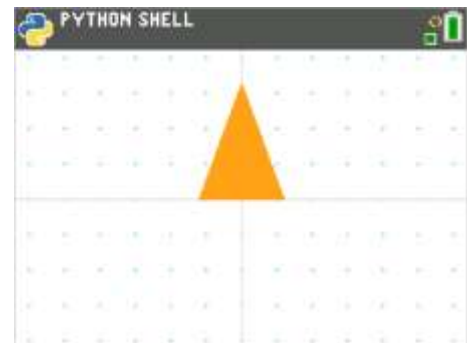
Remarque : Le tracé du triangle peut être simplifié. Il a été conçu ici de cette façon, afin d'assurer la compréhension du calcul de l'angle en se ramenant à 2 triangles rectangles et également sur le fait que le point de départ du tracé est l'origine du repère.

```

ÉDITEUR : TRIANGLE
LIGNE DU SCRIPT 0017
t.speed(10)
t.pencolor(251,163,26)
t.fillcolor(251,163,26)
t.begin_fill()
t.penup()
t.goto(50,25,)
t.setheading(-90)
t.pendown()
t.forward(c/2)
t.left(110)
t.forward(h)
t.left(140)
t.forward(h)
t.end_fill()
t.hideturtle()
t.done()

```

- Exécuter votre script et observer le tracé du triangle isocèle.
- Effectuer plusieurs tracés en modifiant les paramètres du script (valeur de *c*, couleurs du fond et du contour ...).
- Pour construire la couronne solaire, nous devons être capables de modifier l'orientation du triangle ainsi que sa position dans le repère.





a) Modification de la position

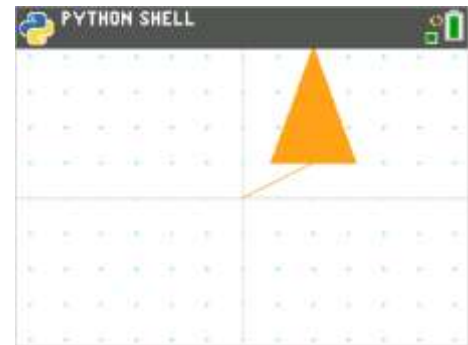
- Dans le menu **Move**, choisir l'instruction **t.goto(x,y)**. Cette instruction indique les coordonnées (x,y) du point vers lequel la tortue doit se déplacer.
- Bien entendu, la place de cette instruction s'impose. Dans notre exemple celle-ci sera avant le tracé du triangle.
- Mettons par exemple **t.goto(50,25)**.
- Exécuter à nouveau le script. Un changement de repère au point de coordonnées (50,25) a bien été effectué, avec l'inconvénient d'observer également le tracé d'un segment entre les points de coordonnées (0,0) et (50, 25).

Conseil à l'enseignant : Le code couleur (r,v,b) entré dans les instructions **t.pencolor()** et **t.fillcolor()** peut être fourni sous la forme d'un tuple (coul=(r,v,b)).

```

EDITEUR : TRIANGLE
LIGNE DU SCRIPT 0002
from turtle import *
t=Turtle()
c=60
b=pi*70/180
h=c/(2*cos(b))
t.speed(10)
t.pencolor(251,163,26)
t.fillcolor(251,163,26)
t.begin_fill()
t.goto(50,25)
t.forward(c/2)

```



- Pour éliminer le segment entre les points de coordonnées (0,0) et (50,25) aller dans le sous menu **Pen** du module Turtle et choisir les instructions **2 : t.penup()** et **3 : t.pendown()**.
- Naturellement, l'instruction de déplacement de la tortue doit être située entre ces deux instructions.

```

EDITEUR : TRIANGLE
LIGNE DU SCRIPT 0016
h=c/(2*cos(b))
t.speed(10)
t.pencolor(251,163,26)
t.fillcolor(251,163,26)
t.begin_fill()
t.penup()
t.goto(50,25)
t.pendown()
t.forward(c/2)
t.left(110)
t.forward(h)

```

Conseil à l'enseignant : **t.penup()** permet à la tortue de se déplacer sans tracer de ligne. **t.pendown()** utilisé après **t.penup()**, permet à la tortue de reprendre le tracé.





- Exécuter le script et constater l'effet produit par ce trio d'instructions.



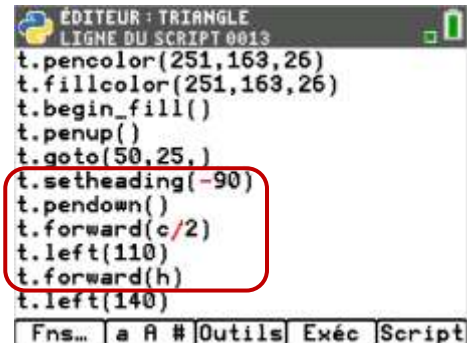
b) Modification de l'orientation

- Dans le sous menu **State** du module Turtle, choisir l'instruction 2 : **t.setheading()**.
- Placer cette instruction avant le tracé du triangle.



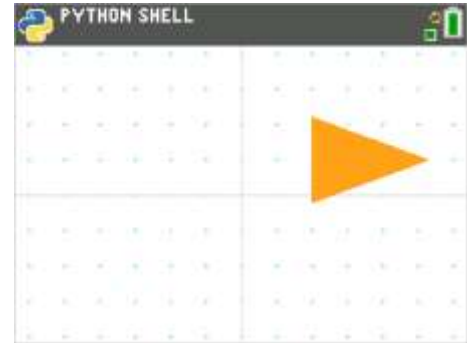
Conseil à l'enseignant : t.setheading(degrés) – définit le cap de la tortue en degrés. Les valeurs positives sont dans le sens inverse des aiguilles d'une montre à partir du cap zéro. Les valeurs négatives sont dans le sens des aiguilles d'une montre.

- Par rapport à la position précédente, le triangle devrait tourner de 90° vers la droite.





- Exécuter de nouveau le script et observer le résultat obtenu.
- Lors de la réalisation du soleil, il sera ainsi possible à l'aide de cette instruction de créer la couronne de rayon. Bien entendu, nous aurons besoin de le faire au sein d'une boucle fermée.



B : Tracé du disque solaire

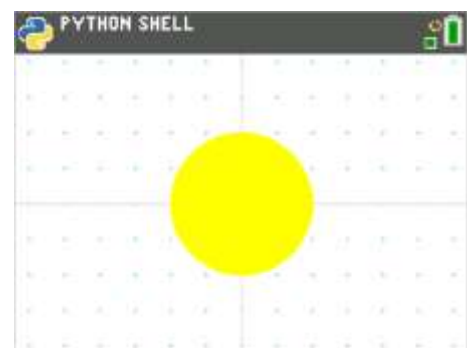
Les instructions **t.fillcolor(r,g,b)** associées à **t.begin_fill()** et **t.end_fill()** ne fonctionnent que pour des polygones et non pour remplir des cercles d'une couleur donnée.

t.circle() trace un cerle et non un disque.

En remplacement, utiliser l'instruction **t.dot(diamètre)** que l'on trouve dans le sous menu **Draw** dessinera un point de diamètre fixé en pixels comme argument de la fonction.

L'instruction **t.pencolor(r,g,b)** permettra de choisir la couleur du crayon utilisé pour dessiner ce point.

- Créer un nouveau script et le nommer DISQUE.
- Recopier les instructions ci-contre afin de réaliser le disque de couleur jaune.
- Tester votre script.



Ce document est mis à disposition sous licence Creative Commons

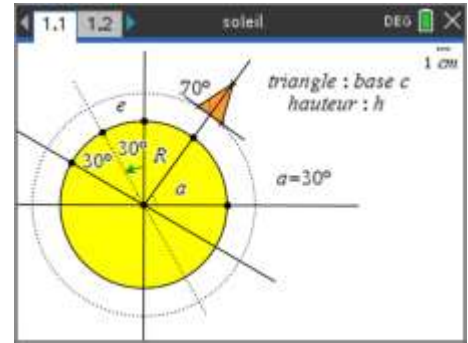


<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

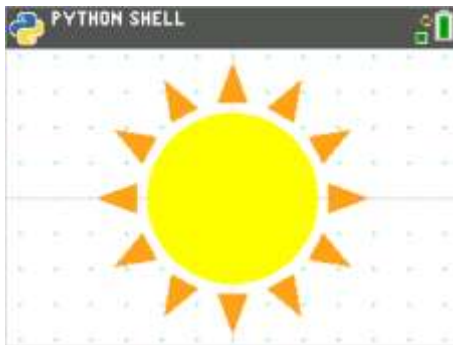


C : Réalisation du soleil (Pour aller plus loin)

- Créer un nouveau script et le nommer TSB2.
- La représentation ci-contre donne le nom des variables utilisées dans le script.
- Autour du cœur du soleil de rayon R , on dispose tous les 30° , un triangle isocèle de base c et de hauteur h . Ces triangles sont représentés sur un cercle de rayon $R + e$.
- Afin d'orienter correctement chaque triangle représentant un rayon du soleil, on modifie le cap de la tortue à l'aide de l'instruction `t.setheading()`.
- La position initiale de la tortue correspond à un angle $\alpha = 0^\circ$ à partir d'une orientation `t.setheading(-90^\circ)`.
- Le script complet reprend en tout point ce qui a été vu dans cette leçon.



Exécuter le script, admirer le résultat obtenu.



```

ÉDITEUR : TSB2
LIGNE DU SCRIPT 0001
from math import *
from turtle import *
t=Turtle()
#sol
r=60
c=20
e=8
p=30
i=0
t.pencolor(255,255,0)
t.dot(2*r)
#rayons
h=c/(2*cos(70*pi/180))
for g in range(0,360,30):
    t.penup()
    t.goto((r+e)*cos(g*pi/180),(r+e)*sin(g*pi/180))
    t.setheading(-90+pi)
    t.pendown()
    t.pencolor(251,163,26)
    t.fillcolor(251,163,26)
    t.begin_fill()
    t.forward(c/2)
    t.left(110)
    t.forward(h)
    t.left(140)
    t.forward(h)
    t.end_fill()
    i+=1
t.hideturtle()
t.done()

```

Ce document est mis à disposition sous licence Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

