

Unité 8 : Module Turtle et Python

Compétence 1 : Installation et découverte du module Turtle

Dans cette leçon, vous allez installer le module Turtle dans la calculatrice TI-83 Premium CE Edition Python et découvrir les instructions permettant à la « Tortue » d'effectuer des déplacements.

**Objectifs :**

- Installer le module Turtle dans la calculatrice.
- Réaliser le tracé de quelques figures simples.
- Créer une figure de « Math-Art ».

Remarque : les sections suivantes supposent que vous avez mis à jour votre TI-83 Premium CE Edition Python vers la version 5.7 de l'OS et des Apps ou une version ultérieure. Si vous n'avez pas effectué la mise à jour vers la version 5.7 ou une version ultérieure, veuillez consulter le site : <https://education.ti.com/fr/produits-ressources/mise-a-jour-ti-83-premium-ce>. Veuillez à sauvegarder ou archiver les fichiers et les données de votre TI-83 Premium CE Edition Python avant de procéder à la mise à jour.

**Transférer le module Turtle à votre calculatrice.**

Transférer le module Turtle et le fichier Grid à votre TI-83 Premium CE Edition  
Après avoir téléchargé le Zip et extrait les fichiers :

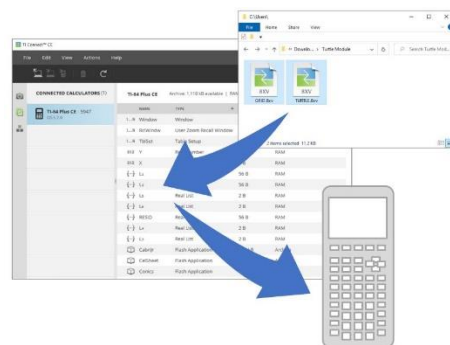
- Ouvrir le logiciel de bureau TI Connect CE.
- Connecter votre TI-83 Premium CE Edition Python à votre ordinateur en utilisant le câble USB ordinateur-calculatrice fourni avec la calculatrice.
- Vérifier que votre calculatrice connectée apparaît dans le panneau Calculatrices connectées de l'espace de travail Explorateur de calculatrices.
- Transférer les fichiers TURTLE.8xv et GRID.8xv vers la ou les calculatrices connectées en faisant glisser les fichiers dans la fenêtre des calculatrices connectées.

Remarque : le fichier GRID.8xv est une image utilisée comme grille de fond dans les programmes Turtle.

Remarque : les fichiers Turtle et Grid peuvent être aussi transférés d'une calculatrice à une autre à l'aide d'une liaison par câble USB d'unité à unité. Consulter le guide de votre calculatrice pour connaître le processus de transfert de fichiers de calculatrice à calculatrice.

Cette étape nécessite l'utilisation du logiciel (gratuit) TI Connect CE pour ordinateur PC ou Mac.

<https://education.ti.com/fr/produits/logiciel-ordinateur/ti-connect-ce-sw>



Ce document est mis à disposition sous licence Creative Commons

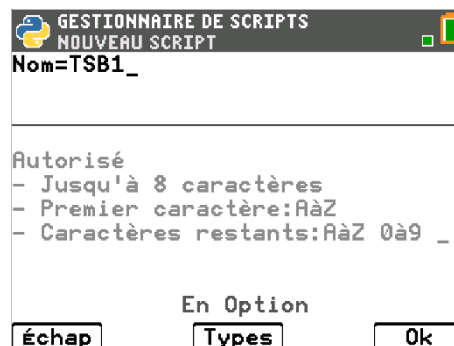
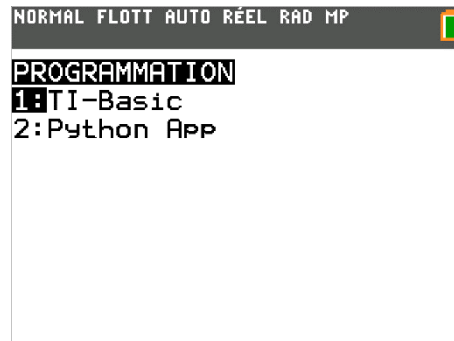
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



## Importation du module Turtle.

Une fois le module Turtle et le fichier Grid transférés, créer un script Python et importer le module Turtle afin d'accéder aux sélections de son menu.

- Sur votre TI-83 Premium CE Edition Python, ouvrir l'application Python. Vous pouvez la trouver à l'aide de la touche **[prgm]** ou de la touche **[apps]**. L'application Python s'ouvre sur un écran de gestion des scripts.
- Créer un nouveau script Python en sélectionnant l'onglet intitulé **Nouv** (appuyez sur la touche **[zoom]**).
- Donner un nom au script Python (par exemple : **TSB1**), puis sélectionner **Ok**.



Ce document est mis à disposition sous licence Creative Commons

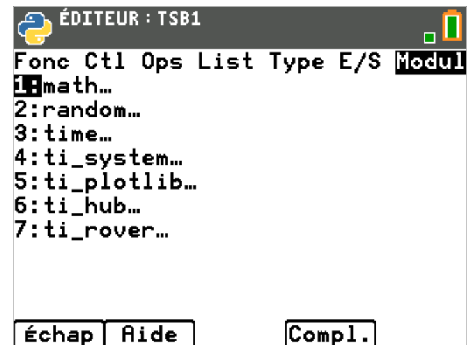
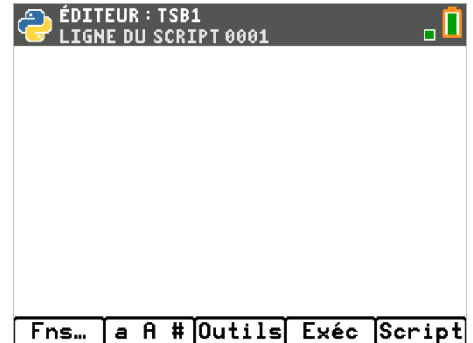
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



Vous êtes maintenant dans l'écran de l'éditeur de scripts Python. Saisir, ensuite, l'instruction d'importation Turtle et un objet Turtle.

- Sélectionner l'onglet **[Fns...]** (en bas de l'écran), puis flèche gauche vers l'onglet **Modul** (en haut de l'écran).
- En bas de l'écran, sélectionner l'onglet **[Compl.]**.
- Sélectionner l'option **3:from turtle import \*** pour coller l'instruction d'importation du module Turtle.

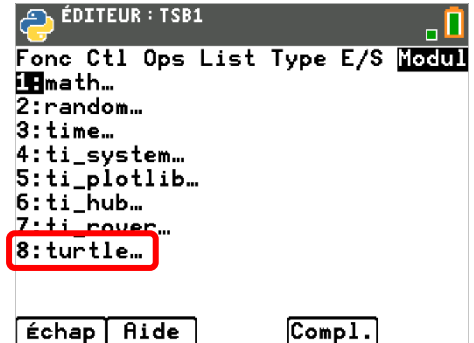
Remarquer qu'il colle aussi un constructeur **t=Turtle()** pour assigner à t l'objet Turtle.





Retourner à l'onglet **Modul** et remarquer l'apparition de **8 : turtle...** dans le menu.

**Conseil à l'enseignant :** Remarque : l'onglet [Compl.] n'est visible qu'avec la version 5.7 ou ultérieure du système d'exploitation. Si vous utilisez une version antérieure, mettez à jour vers la version 5.7 ou ultérieure ; sinon, vous devez saisir manuellement l'instruction d'importation et le constructeur **t=Turtle()**.



Maintenant, vous êtes prêt à accéder aux sélections de menu du module Turtle et à écrire un script Turtle !

**Conseil à l'enseignant :** « Turtle Graphics » a été introduit dans le langage de programmation Logo créé par Wally Feurzeig, Cynthia Solomon et Seymour Papert au milieu des années 1960. La 'tortue' est un objet graphique qui répond à des commandes de programmation comme **forward(distance)** et **right(angle)**. Le module tortue est orienté objet et les fonctions de base tortue sont incluses dans les menus tortues.

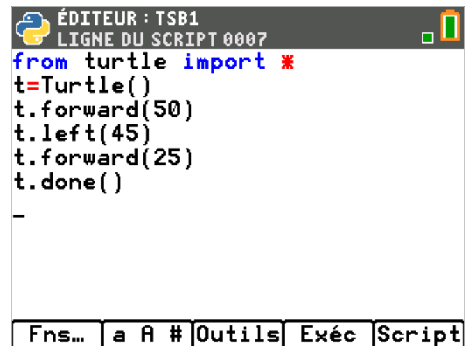
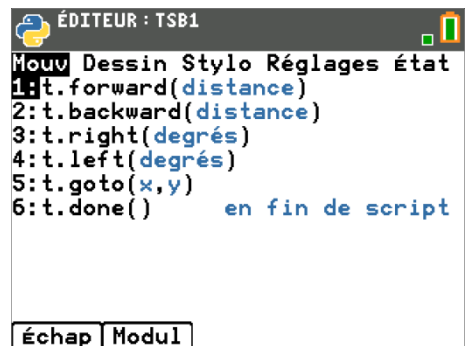
### A) Déplacer la tortue.

- Appuyer sur **f1 (Fns...)** afin d'accéder aux menus, puis choisir **Modul**.
- Choisir **8 : turtle...** afin d'accéder à l'ensemble des menus de la bibliothèque. Les instructions du menu **Move** (mouvement) sont assez explicites. Les distances à renseigner entre parenthèses sont en pixels.
- Compléter le script afin de faire réaliser à la tortue l'algorithme suivant :

```

Début de l'algorithme
t ← Turtle()
Avance de 50 pixels
Tourne vers la gauche de 45°
Avance de 25 pixels
Fin de l'algorithme

```



- L'instruction **t.done()** est nécessaire afin de pouvoir afficher la figure lorsque l'exécution du script est terminée.


Ce document est mis à disposition sous licence Creative Commons

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



Appuyer sur la touche **f4 (Exéc)** et observer la réalisation.

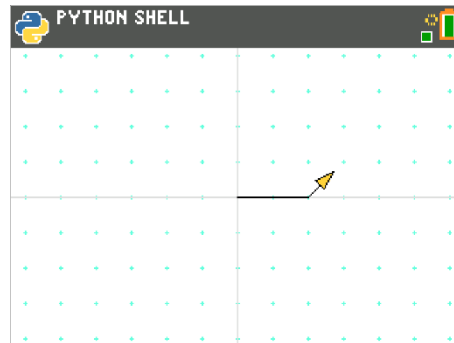
La tortue est par défaut placée à l'origine du repère.

Par défaut également, la tortue  est visible et vous la voyez dessiner. Noter également que la grille est visible, (l'échelle de la grille est de 25 pixels par carré).

Les instructions du menu Settings permettent de modifier ces options.

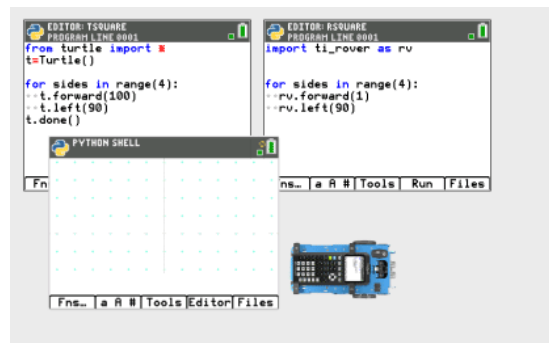
- `t.hideturtle()` – rend la tortue invisible.
- `t.showturtle()` – rend la tortue visible.
- `t.hidegrid()` – masque la grille.

Lorsque vous avez fini d'admirer votre travail, appuyer sur la touche **[annul]** pour effacer l'écran. Remarquer que vous vous retrouvez sur un écran du Shell de Python. C'est là que votre script Python a été traité.



### B) Tracer un polygone régulier.

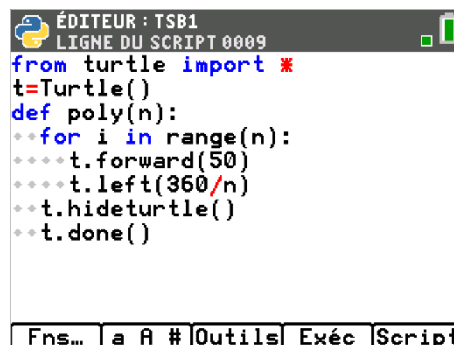
On se propose de réaliser un script permettant de tracer un polygone régulier. En effet, le fonctionnement de la tortue est analogue dans ses déplacements à celui du TI-Rover.



On propose de créer une fonction **poly(n)** permettant de tracer un polygone régulier de  $n$  côtés. Il semble raisonnable de conserver 50 pixels pour longueur de chaque côté.

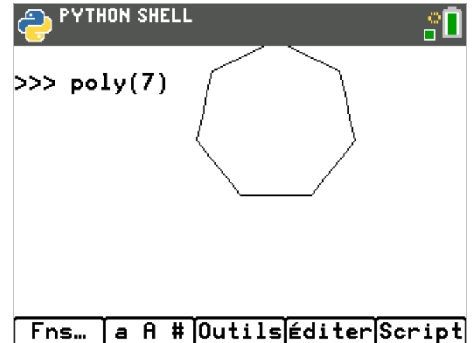
Chaque angle intérieur d'un polygone régulier de  $n$  côtés mesure  $\frac{360^\circ}{n}$ .

Il est préférable de cacher la tortue à la fin de l'exécution du script.



Exécuter le script pour quelques polygones simples.

A partir d'un heptagone (7 côtés), la figure commence à ne pas être totalement visible dans l'écran.

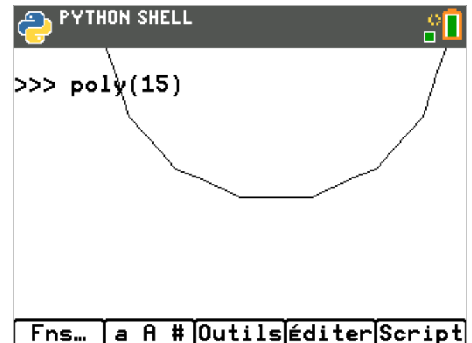


```
PYTHON SHELL
>>> poly(7)
```

Réaliser la représentation d'un pentadécagone (15 côtés).

Comment inscrire la figure complète dans l'écran ?

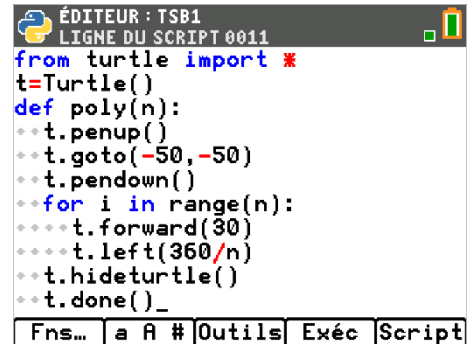
- Modifier l'origine du tracé.
- Réaliser un polygone de côté 30 pixels au lieu de 50.



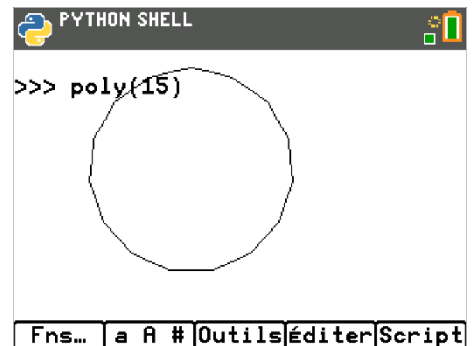
```
PYTHON SHELL
>>> poly(15)
```

Le point de départ du tracé doit être fixé plus bas dans l'écran. Au point de coordonnées (-50 ; -50) par exemple.

Afin que ne soit tracé un segment du point de coordonnées (0,0) au point de coordonnées (-50,-50), il convient de soulever le crayon **t.penup( )**, puis de le replacer en position d'écriture une fois le changement effectué **t.pendown( )**.



```
ÉDITEUR : TSB1
LIGNE DU SCRIPT 0011
from turtle import *
t=Turtle()
def poly(n):
    t.penup()
    t.goto(-50,-50)
    t.pendown()
    for i in range(n):
        t.forward(30)
        t.left(360/n)
    t.hideturtle()
    t.done()
```



```
PYTHON SHELL
>>> poly(15)
```



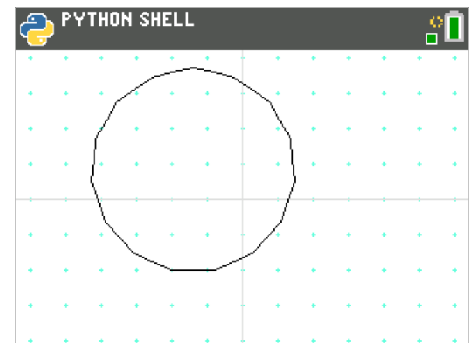
#### Remarque :

Les fonctions sont très utiles en Python, mais dans le cadre de l'exécution d'un script utilisant le module Turtle, l'appel de celles-ci au sein d'un programme évitera d'avoir un affichage peu esthétique. On conserve également la représentation de la grille, ce qui a l'avantage de faciliter les réglages des paramètres des instructions de déplacement.

```

ÉDITEUR : TSB1
LIGNE DU SCRIPT 0013
def poly(n):
    t.penup()
    t.goto(-50,-50)
    t.pendown()
    for i in range(n):
        t.forward(50)
        t.left(360/n)
    t.hideturtle()
    t.done()
#polygone
poly(15)

```



#### Un peu de « maths arts » si vous en avez le temps

- Dupliquer le script précédent (Menu **Gérer** à partir du Gestionnaire de scripts) et le nommer TSB11.
- Modifier le script afin que soit représenté un carré.
- La tortue effectue ensuite une rotation de 5° à droite.
- Le tracé est réitéré 36 fois par exemple (tracé un peu long : un essai à 18 donne aussi une belle image).

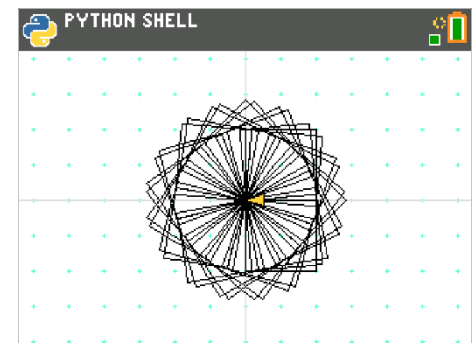
Il ne nous manque plus qu'à mettre un peu de couleur, ce que nous ferons lors de la leçon suivante.

**Conseil à l'enseignant :** Lorsque vous voulez travailler sur un script existant, il peut être pratique d'en faire une copie, puis de modifier le code de la copie. Ceci peut être réalisé à partir du gestionnaire de scripts 1) en sélectionnant le script à copier, puis 2) en sélectionnant l'onglet **[Gérer]**, puis 3) en sélectionnant l'option **1: Dupliquer le script...** et enfin 4) en entrant le nom du nouveau script.

```

ÉDITEUR : TSB11
LIGNE DU SCRIPT 0001
from turtle import *
t=Turtle()
def carre():
    t.forward(50)
    for i in range(3):
        t.right(90)
        t.forward(50)
    t.right(5)
#math art
for i in range(36):
    carre()
t.done()

```



Ce document est mis à disposition sous licence Creative Commons

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

