

Construction de droite graduée et de repère



Résumé : dans cette activité, les élèves doivent programmer un lutin ou `turtle` afin de tracer une droite graduée, puis par extension un repère du plan, afin de pouvoir tracer des fonctions dans l'activité suivante.

Mots-clés : bibliothèque `turtle` ; droite graduée ; repère du plan

Compétences visées

Chercher : « Observer, s'engager dans une démarche, expérimenter en utilisant éventuellement des outils logiciels » avec ici des résultats graphiques à observer directement.

Représenter : « passer d'un mode de représentation à un autre » en comprenant bien comment représenter un axe gradué puis en finalisant la construction d'un repère orthonormé.

Calculer : « mettre en œuvre des algorithmes simples », en utilisant ici des fonctions à réutiliser au cours du programme principal.

Situation déclenchante

Tracer un simple repère orthonormé ou du moins orthogonal du plan peut s'avérer long pour certains élèves. L'utilisation du mode graphique d'une calculatrice, comme la TI-83 Premium CE Edition Python, peut sembler être une boîte noire qu'il est intéressant d'ouvrir aux élèves afin de remédier à certaines difficultés.

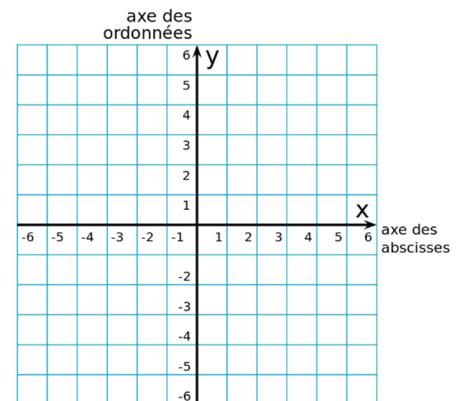


Image libre de droits d'après [Wikipédia](#)

Problématique

Comment tracer dans un premier temps une droite graduée, puis, dans un deuxième temps, un repère du plan par programmation ?



Construction de droite graduée et de repère



Scénario pédagogique

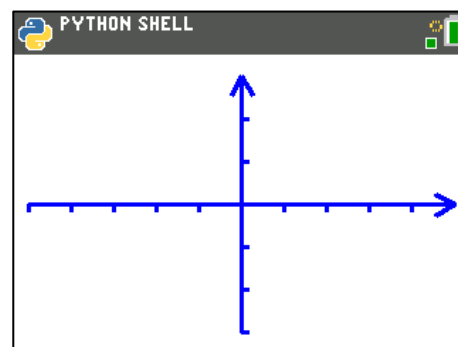
- **Avec la classe :** lors de la restitution d'un devoir sur la représentation graphique ou avec la volonté d'ouvrir une boîte noire de la calculatrice : le tracé des axes d'une fenêtre graphique, proposer aux élèves cette activité.
- **Individuellement ou en groupe pour la droite graduée :** il est possible de donner le code du script `T05_DTEG.py` et de demander de comprendre à quoi peut servir ce code. Il est possible aussi de demander la création du script permettant de tracer une droite graduée, en devoir hors classe ou lors d'une séance informatique.
- **Avec la classe :** faire un point en listant les différentes propositions des élèves, et donner la programmation possible `T05_GRAD.py` que les élèves peuvent récupérer en utilisant un câble mini-USB pour ceux dont la programmation est défectueuse ou ne permettant pas son utilisation.
- **Individuellement ou en groupe pour le repère :** il s'agit d'inciter les élèves à utiliser le script de la droite graduée pour tracer le repère, en leur indiquant comment récupérer d'autres scripts par l'instruction `from nom du script import *`.
- **Mise en commun :** faire un point avec les élèves et donner le script `T05_REP.py`, ou un script équivalent créé par un élève si l'activité suivante de création de représentation graphique d'une fonction est envisagée.
- **Pour les élèves les plus en avance :** il est possible de leur proposer un ou plusieurs prolongements possibles, décrit en [fin de fiche](#).
- **Difficultés rencontrées :**
 - la notion de « sous-fonction ».

Voici les visuels à l'issue des programmes :

en Scratch pour la demi-droite graduée




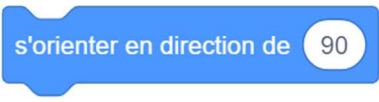
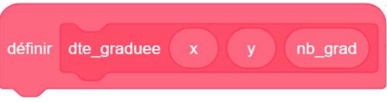
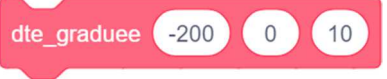
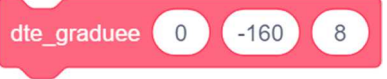
avec la TI-83 Premium CE Edition Python pour le repère




Construction de droite graduée et de repère



Avec Scratch

Les briques de codes principales en Scratch pour ce programme	Explications Cette instruction permet de :	Traduction en langage Python sur la TI-83 Premium CE Edition Python
	Avancer de 10 pas ou pixels.	<pre>turtle.forward(10)</pre> <p><i>Forward</i> signifie « vers l'avant » en anglais.</p>
	D'orienter le lutin selon un angle défini. Attention, les orientations sont différentes : voir la partie « Pour mieux lire le code Python ».	<pre>turtle.setheading(90)</pre> <p><i>Set heading</i> signifie « définir le cap » en anglais.</p>
	Créer un bloc utilisateur avec trois paramètres : x ; y et nb_grad, les deux premiers désignent les coordonnées de départ et le troisième détermine le nombre de graduations à faire. Il faut les définir lors de la création ou modifier par la suite.	<pre>def dte_grad(x,y,n): ♦♦instructions</pre> <p>L'indentation avec ♦♦ est nécessaire pour définir les instructions à exécuter dans un bloc du code Python.</p>
 	Utiliser le bloc utilisateur défini au préalable. Attention, les fenêtres d'exécution du programme sont différentes : voir la partie « Pour mieux lire le code Python ».	<pre>dte_grad(-150,0,10) dte_grad(0,-90,6)</pre>

A noter que dans la dernière version de Scratch, il faut chercher ce qui concerne le stylo dans les extensions : 

Une programmation possible est disponible sur le site de Scratch : scratch.mit.edu/studios/27615196/

Construction de droite graduée et de repère



Avec Python

Suivant le choix de procédure, soit le code complet est à construire par les élèves, soit partiellement, en ayant donné le script T05_DTEG.py.

Ce code est composé de deux scripts T05_DTEG.py et T05_REP.py, composés eux-mêmes de plusieurs fonctions. Il est d'ailleurs possible de rappeler un script précédent comme dans l'exemple choisi ici :

- Dans T05_DTEG.py, la fonction `gt` de paramètres `x` et `y` qui permet au stylo d'aller directement au point de coordonnées `(x,y)` sans écrire.

```
ÉDITEUR : T05_DTEG
LIGNE DU SCRIPT 0001
from ce_turtl import *

def gt(x,y):
    turtle.penup()
    turtle.goto(x,y)
    turtle.pendown()
```

Dans T05_DTEG.py, la fonction `dte_grad` de paramètres `x` ; `y` et `n`.

`x` et `y` sont les coordonnées de départ de la droite graduée, tandis que `n` désigne le nombre de graduation à effectuer.

Il a été choisi de faire une graduation tous les 30 pixels, visible par l'instruction `turtle.forward(30)`.

Les cinq dernières instructions permettent de tracer la flèche indiquant le sens de la droite graduée.

```
ÉDITEUR : T05_DTEG
LIGNE DU SCRIPT 0018
def dte_grad(x,y,n):
    gt(x,y)
    for i in range(n):
        turtle.right(90)
        turtle.forward(5)
        turtle.backward(5)
        turtle.left(90)
        turtle.forward(30)
    turtle.left(30)
    turtle.backward(15)
    turtle.forward(15)
    turtle.right(60)
    turtle.backward(15)
```

- Dans T05_DTEG.py, la fonction `main` sans paramètre.

C'est la fonction principale permettant de tracer la droite graduée.

Après une initialisation classique, en mettant tout de même la taille du stylo au niveau moyen, le script lance la fonction `dte_grad`.

L'instruction `turtle.show()` permet de tracer le dessin demandé.

```
ÉDITEUR : T05_DTEG
LIGNE DU SCRIPT 0028
def main():
    turtle.home()
    turtle.clear()
    turtle.pensize(1)
    turtle.color(0,0,255)
    dte_grad(-150,0,10)
    turtle.show()
```

- Dans T05_REP.py, la fonction `main2` sans paramètre permet de tracer le repère orthogonal du plan demandé.

A noter que dans cette programmation, le script T05_DTEG.py est appelé par l'instruction `from T05_DTEG import *`. Cela permet de pouvoir utiliser les fonctions définies dans ce script.

```
ÉDITEUR : T05_REP
LIGNE DU SCRIPT 0001
from ce_turtl import *
from T05_DTEG import *

def main2():
    turtle.home()
    turtle.clear()
    turtle.pensize(1)
    turtle.color(0,0,255)
    dte_grad(-150,0,10)
    turtle.setheading(90)
    dte_grad(0,-90,6)
    turtle.show()
```

Une programmation possible est disponible sur le site TI : education.ti.com/fr/scratch-python



Construction de droite graduée et de repère



Mode opératoire

Une fois le script exécuté, il faut appuyer sur la touche [var] : les fonctions définies dans le script apparaissent.

Par les flèches directionnelles, il faut sélectionner la fonction `main()` ou `main2()` suivant le script choisi, valider par Ok, puis lancer sans avoir besoin de mettre de paramètre.

Pour sortir du mode dessin, il faut appuyer sur la touche [on].

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de T05_REP
>>> from T05_REP import *
>>> main2()
Fns... a A # Outils Éditer Script
```

Prolongements possibles

Voici des pistes pour les élèves les plus rapides ou qui ont envie de prolonger le travail :

- programmer pour définir les axes par rapport à un point d'origine défini par l'utilisateur, de façon à pouvoir obtenir uniquement que le premier quadrant, utile en physique par exemple ;
- programmer afin de gérer l'espacement des graduations en fonction du nombre de celles-ci.

Pour mieux lire le code Python

A noter ici qu'il est nécessaire d'importer la bibliothèque `turtle` en l'appelant en début de programmation par la ligne : `from ce_turtl import *`.

Dans Scratch, 0 [degré] dirige le lutin vers le haut, et 90 [degrés] vers la droite, et ainsi de suite dans le sens horaire. Pour la calculatrice TI-83 Premium CE Edition Python, de même que dans Python, dans la version standard, le 0 [degré] désigne la droite, puis 90 [degrés] le haut, en tournant dans le sens trigonométrique.

La fenêtre d'exécution, autant dans Scratch que sur la calculatrice TI-83 Premium CE Edition Python, étant rectangulaire non carrée, il est nécessaire d'adapter les coordonnées de départ, ainsi que le nombre de graduation.

Pour information, sur Scratch, la taille d'écran est de 480 pixels par 360 pixels. Sur la calculatrice, elle est de 320 pixels par 240 pixels. Ces différences nécessitent donc des aménagements pour passer d'une programmation à l'autre, occasionnant des problèmes mathématiques qu'il peut être intéressant à développer pour les élèves les plus en avance.

