

Le L : retour au point de départ !



Résumé : dans cette activité, l'élève doit déterminer, dans un triangle rectangle la mesure d'un angle ainsi que la longueur du troisième côté du triangle afin de revenir au point de départ. L'élève commence par chercher des pistes à l'aide d'un programme fourni, puis raisonner pour démontrer mathématiquement son résultat.

Mots-clés : trigonométrie ; chercher ; raisonner ; bibliothèque `turtle` ; théorème de Pythagore

Compétences visées

Chercher : « observer, s'engager dans une démarche, expérimenter en utilisant éventuellement des outils logiciels » avec ici un résultat qui s'observe directement par la construction de la figure.

Modéliser : « utiliser, comprendre, élaborer une simulation numérique ». Il s'agit en particulier ici de placer ce problème dans un contexte mathématique.

Calculer : « contrôler les calculs » ; ici la construction permet de savoir si la réponse est juste ou pas.

Situation déclenchante

Un bras robot doit graver un **L** sur un objet au laser. Pour cela, il commence à marquer un premier segment vertical de 100 mm, puis un deuxième segment, cette fois horizontal, de 80 mm.

Avant que l'objet suivant se présente pour être gravé, le robot doit reprendre sa position initiale en parcourant le chemin le plus court possible.

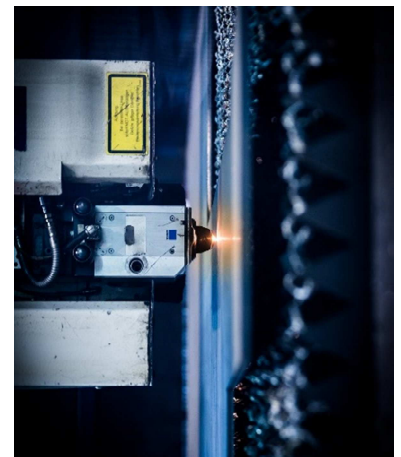


Image libre de droits d'après [Pixabay](https://pixabay.com/)

Problématique

Quelles instructions donner au robot pour revenir le plus rapidement à son point de départ ?



Le L : retour au point de départ !



Scénario pédagogique

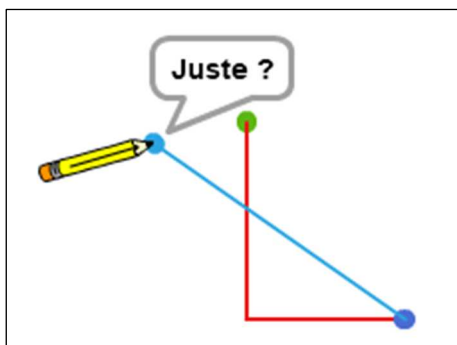
- **Avec la classe** : explicitation du problème par un court échange à l'oral avec la classe.
- **Tests** : les élèves testent individuellement le script pour s'approprier le problème.

Utilisation d'un code :

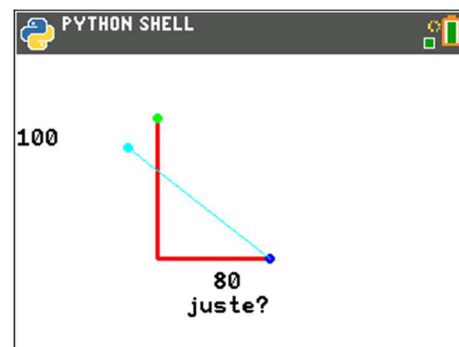
- Dans cette activité, l'outil informatique fournit une visualisation du résultat obtenu après calculs de l'élève et lui permet de se rendre compte par lui-même de la justesse de ses résultats.
- Le code est fourni à l'élève : charge à lui de saisir les valeurs numériques de l'angle et de la longueur de l'hypoténuse pour observer le résultat et réagir en conséquence.
- **Mise en commun** (éventuellement au sein de petits groupes) : quels outils mathématiques sont en jeu ? Le problème mathématique posé est double ; quel est l'angle au sommet du triangle qui sera formé après le retour, et quelle est la longueur de l'hypoténuse ?
- **Preuve** : les élèves doivent utiliser le théorème de Pythagore pour déterminer la longueur du 3^e côté et la trigonométrie pour déterminer la mesure de l'angle de retour.
- **Pour les élèves les plus en avance** : il est possible de leur proposer un ou plusieurs prolongements possibles, décrit en [fin de fiche](#).

Voici les visuels à l'issue des programmes :

en Scratch



avec la TI-83 Premium CE Edition Python



Le L : retour au point de départ !



Avec Scratch

Les briques de codes principales en Scratch pour ce programme	Explications	Traduction en langage Python sur la TI-83 Premium CE Edition Python
	Cette instruction permet de : Poser le stylo, depuis le centre du lutin par défaut, afin de laisser une trace.	<code>turtle.pendown()</code>
	Relever le stylo et donc de déplacer le lutin sans laisser de trace.	<code>turtle.penup()</code>
	Demander une valeur à l'utilisateur.	Une fonction Python peut nécessiter la saisie d'aucun, d'un ou plusieurs paramètres.
	Obtenir la réponse de l'utilisateur.	Cette valeur sera rentrée comme paramètre dans une fonction.
	Tourner vers la gauche de l'angle indiqué, ici de la différence entre 180° et la réponse donnée par l'utilisateur.	<code>turtle.left(180-a)</code>
	Avancer de 10 pixels selon une direction indiquée précédemment, vers la droite par défaut.	<code>turtle.forward(10)</code>

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/



Le L : retour au point de départ !



Avec Python

Le code complet est fourni aux élèves, dont voici quelques extraits.

Ce code est composé de plusieurs fonctions :

- La fonction `gt` de paramètres `x` et `y` qui permet au stylo d'aller directement au point de coordonnées `(x ; y)` sans écrire.
- La fonction `cons1` (fonction sans paramètre) qui permet de rédiger une consigne sans l'afficher.
- La fonction `consigne1` (fonction sans paramètre) qui reprend la fonction précédente en ajoutant une aide à la saisie de la réponse.

La fonction `rep1` : elle a pour paramètres un angle exprimé en degrés et une longueur, celle du troisième côté.

Cette fonction reprend une partie de la consigne et affiche en sortie le segment construit à partir des données saisies par l'élève ainsi que le texte "juste?" centré à la onzième ligne du code.

```
ÉDITEUR : ROBOT
LIGNE DU SCRIPT 0011
from ce_turtl import *
from ti_system import *

turtle.clear()

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

```
ÉDITEUR : ROBOT
LIGNE DU SCRIPT 0022
def cons1():
    * turtle.clear()
    * disp_at(4, "100", "left")
    * disp_at(10, "80", "center")
    * gt(-60,60)
    * turtle.pensize(1)
    * turtle.color(255,0,0)
    * turtle.goto(-60,-40)
    * turtle.goto(20,-40)
    * turtle.color(0,0,255)
    * turtle.dot(3)
    * gt(-60,60)
    * turtle.color(0,255,0)
    * turtle.dot(3)
```

```
ÉDITEUR : ROBOT
LIGNE DU SCRIPT 0033
def rep1(a,l):
    * cons1()
    * turtle.color(0,255,255)
    * turtle.pensize(0)
    * gt(20,-40)
    * turtle.setheading(0)
    * turtle.left(180-a)
    * turtle.forward(l)
    * turtle.dot(3)
    * disp_at(11, "juste?", "center")
    * turtle.show()
```

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



Le L : retour au point de départ !



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 valider la fonction `consigne1`. L'élève voit apparaître la consigne. Il sort de cet écran en appuyant sur la touche [on].

Il va chercher la fonction `rep1` dans laquelle il saisit l'angle et la longueur qu'il avait déterminée : par exemple `rep1(28.6,125)` ; attention à la signification de la virgule en Python, qui permet de séparer des paramètres, et du point. Dans le système anglo-saxon, le point décimal joue le rôle de la virgule décimale dans un nombre.

```
PYTHON SHELL
VARS : ROBOT
▶cons1()
  consigne1()
  gt()
  rep1()
Échapp Ok
```

Prolongements possibles

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

- comprendre pourquoi l'angle à saisir pour tourner à gauche est $180-a$ et non pas a ;
- demander à modifier le code pour effectuer une gravure en **L** avec d'autres dimensions ;
- générer un point de départ aléatoire pour éviter la modification facile du code « aller à $(-60 ; 60)$ » en fin de script par des élèves ;
- demander de coder d'autres lettres simples comme le H, le T ou le X, voire plus complexes avec des lettres arrondies comme le S.

```
ÉDITEUR : ROBOT
LIGNE DU SCRIPT 0040
def rep1(a,l):
  *cons1()
  *turtle.color(0,255,255)
  *turtle.pensize(0)
  *gt(20,-40)
  *turtle.setheading(0)
  *turtle.left((180-a))
  *turtle.forward(l)
  *turtle.dot(3)
  *disp_at(11,"juste?", "center")
  *turtle.show()
Fns... a A # Outils Exéc Script
```

Pour mieux lire le code Python

Il faut bien comprendre que dès que le script rencontre l'instruction `turtle.show`, l'écran propose l'image codée précédemment. Pour quitter cet écran, il faut appuyer sur la touche [on].

C'est pourquoi on a ici proposé une fonction `consigne` et la fonction `rep`.

Ces deux fonctions appellent la fonction `cons1`, ce qui permet d'éviter de recopier plusieurs fois le même texte.

Bien avoir en tête les coordonnées des extrémités de l'écran : $(-160 ; 115)$ pour le coin supérieur gauche et $(160 ; -115)$ pour le coin inférieur droit.