

# Notice du module « graphpython »

## Pour Python 2.5, 2.6 et versions 3

Ce module n'a pas vocation à être un traceur de courbes ou un logiciel de géométrie dynamique. Il s'agit de permettre à un élève de seconde de construire, avec quelques instructions simples, des programmes graphiques en Python.

### 1. Les instructions du module graphpython.

Pour les instructions qui suivent le paramètre *couleur* est une chaîne et il est optionnel. Huit couleurs disponibles, ce sont : noir, bleu, vert, orange, violet, rose, rouge et jaune.

- **affiche** : cette instruction **doit obligatoirement figurer** à la fin du programme de tracé. Elle permet d'afficher le graphique et de le fermer dans de bonnes conditions à l'aide du bouton « Quitter » de la fenêtre.
- **fenetre**( $x_{min}, x_{max}, y_{min}, y_{max}$ ) : ouvre et affiche une fenêtre graphique. Cette instruction est équivalente à « window » de certaines calculatrices.
  - ☞ Si  $x_{min} \geq x_{max}$  ou  $y_{min} \geq y_{max}$  il n'y a pas d'erreurs, la procédure rétablissant l'ordre.
- **axes**(*gradx*, *grady*, *couleur*) : cette instruction affiche deux axes gradués (s'ils sont visibles dans la fenêtre). Les nombres *gradx* et *grady* sont les graduations respectives sur (*Ox*) et (*Oy*). Les paramètres sont optionnels, s'ils sont omis la valeur par défaut de *gradx* et *grady* vaut 1. La valeur par défaut de couleur est « rouge ». Par exemple : **axes**() a le même effet que **axes**(1,1,rouge) et **axes**(grady=0.5) a le même effet que **axes**(1,0.5,rouge).
- **point**(*x*, *y*, *couleur*) : trace un point aux coordonnées (*x*, *y*) avec la couleur « couleur » (valeur par défaut : noir). Par exemple : **point**(1,1) a même effet que **point**(1,1,noir), c'est à dire place un point de couleur noire au point de coordonnées (*xy*).
- **segment**( $x_1, y_1, x_2, y_2, couleur$ ) : trace un segment joignant les points ( $x_1; y_1$ ) et ( $x_2; y_2$ ) avec la couleur « couleur » (valeur par défaut : noir).
- **cercle\_cr**(*x*, *y*, *r*, *couleur*) : trace un cercle de centre (*x*, *y*) et de rayon *r*. Si le rayon est nul un point est tracé, s'il est négatif rien n'est tracé et il n'y a pas de message d'erreur.
  - ☞ Le repère doit être orthonormé, c'est à dire que la fenêtre graphique doit être ouverte avec une instruction du type « fenetre(-*x*; *x*; -*x*; *x*) » sinon on obtiendra une ellipse.
- **cercle\_cp**( $x_c, y_c, x_p, y_p, couleur$ ) : trace un cercle de centre ( $x_c, y_c$ ) passant par le point ( $x_p, y_p$ ) avec la couleur « couleur ».
  - ☞ Même remarque qu'au dessus.

De façon générale si on trace un segment ou un cercle qui n'est pas contenu entièrement dans la fenêtre définie, seule la partie contenue dans cette fenêtre sera affichée.

### 2. Pour utiliser le module « graphpython » dans un programme il faut insérer en début de programme la ligne : `from graphpython import *`

- ☞ Il est alors inutile d'importer le module `math` qui est importé par le module « graphpython »

### 3. Installation du module

*Le module fonctionne avec les versions 2 et 3 de python. Mais lors de l'écriture des programmes l'utilisant il faudra tenir compte de ces différences (voir les sources des fichiers « `essai_pour_python2.py` » et « `essai_pour_python3.py` »).*

Pour installer le module : soit on le copie dans le même répertoire que le ou les programmes qui l'utilisent, ou bien on l'installe de façon « permanente » en le copiant dans un répertoire où l'éditeur puisse le trouver. Par exemple : C : Program Files Python Lib

Pour lancer l'installation : ouvrir le fichier « `essai_pour_python2.py` » ou

« `essai_pour_python3.py` » (qui est un petit exemple de programme illustrant l'utilisation du module) à l'aide de l'éditeur IDLE, puis l'exécuter avec la commande RUN, cela crée un nouveau fichier nommé « `graphpython.pyc` ». Refermer, puis rouvrir et relancer le programme « `essai_pour_python2.py` » ou « `essai_pour_python3.py` ». Une nouvelle fenêtre doit s'ouvrir avec une courbe et des cercles. Maintenant le module « `graphpython` » peut être insérer dans les nouveaux programmes.

☞ L'instruction **fenetre**( $x_{min}, x_{max}, y_{min}, y_{max}$ ) ouvre une fenêtre graphique de  $600 \times 600$  pixels. Si on veut modifier cette taille il faut, dans le module « `graphpython.py` », modifier les valeurs de variables `self.H` et `self.L` aux lignes 19 et 20. Par exemple : si on veut remplacer 600 pixels par 900 pixels il faudra modifier ces lignes de la façon suivante : `self.H=900` et `self.L=900`.