FICHE PRATIQUE :

PYTHON - MATPLOTLIB

Lycée

Algorithmique

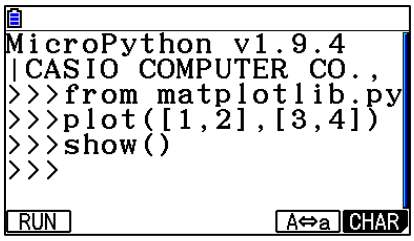
Python

FICHE PRATIQUE

Menu Python

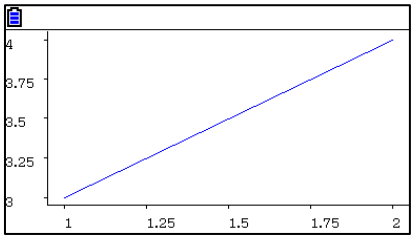
Matplotlib est une bibliothèque disponible dans le langage Python qui permet de tracer et de visualiser des données sous formes de graphiques.

Pour pouvoir l'utiliser dans vos programmes Python ou directement dans le Shell, il faudra l'importer à l'aide de l'instruction *from matplotlib.pyplot import* *, instruction disponible dans le catalogue.



PLOT :

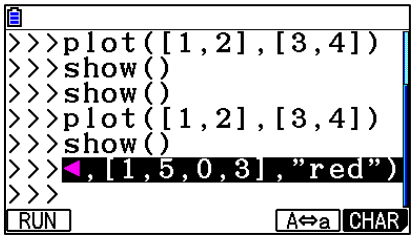
La fonction *plot* permet de tracer une ligne brisée continue entre plusieurs points dont on donne la liste des abscisses puis la liste des ordonnées, séparées par une virgule.



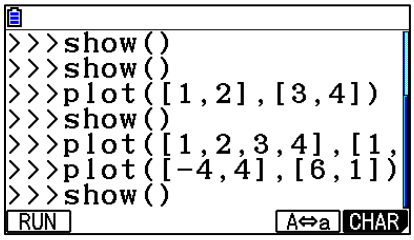
Préparons nous à tracer le segment joignant les points de coordonnées (1 ; 3) et (2 ; 4) avec *plot([1,2],[3,4])*.

En appuyant sur EXE, rien ne se passe.

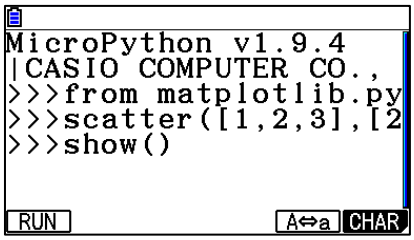
Il faudra utiliser la fonction *show()* pour visualiser le graphique. Sauf mention contraire, la fenêtre s'adapte automatiquement aux valeurs que nous avons entrées.
On revient à la console d'exécution (*Shell*) en appuyant sur EXIT.

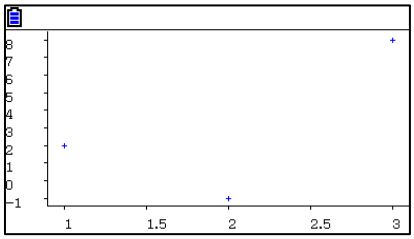


La fonction *plot* peut s'utiliser avec davantage de points pour faire une ligne brisée continue, par exemple avec quatre points de coordonnées respectives (0 ; 1), (2 ; 5), (3 ; 0) et (-1 ; 3).
Il est aussi possible de spécifier la couleur de notre ligne brisée en la rajoutant dans un troisième argument (en anglais et entre guillemets), ici "*red*".



Nous pouvons aussi enchaîner plusieurs *plot* avant l'affichage du graphique final. Là encore la fenêtre s'adapte automatiquement.

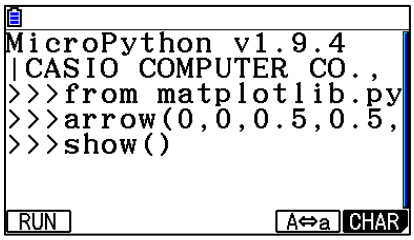


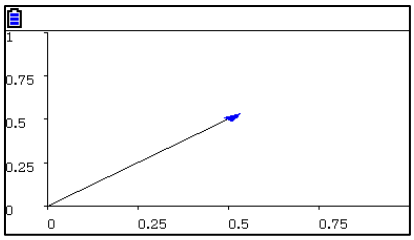


SCATTER :

La fonction *scatter* permet de faire un nuage de points, en entrant comme arguments la liste des abscisses des points puis celle des ordonnées correspondantes.

Exemple : *scatter([1,2,3],[2,-1,8])*





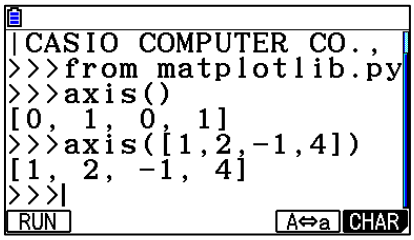
ARROW :

Le tracé de vecteurs est disponible avec la fonction *arrow*. Celle-ci fonctionne avec :

- 4 arguments obligatoires : l'abscisse du point d'origine, son ordonnée, l'écart horizontal entre l'origine et l'arrivée, l'écart vertical,
- éventuellement deux autres : la largeur (*head_width*) et la longueur (*head_length*) de la flèche (optionnels mais nécessaires pour voir la flèche du vecteur).

Par défaut, la largeur de la flèche est égale 0,003 et la longueur à 1,5 fois la largeur.

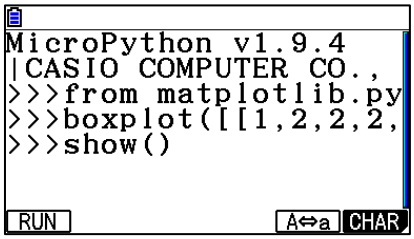
A noter que pour la fonction *arrow*, la fenêtre ne s'adapte pas automatiquement.



AXIS :

Pour adapter la fenêtre graphique manuellement, il faut utiliser la fonction *axis*. Si on utilise cette dernière sans argument, elle renvoie les valeurs minimales puis maximales des abscisses et des ordonnées de la fenêtre (par défaut [0,1,0,1]).

Pour modifier la fenêtre graphique, il suffit d'entrer comme argument la liste des minimums et maximums des abscisses puis des ordonnées de la fenêtre voulue. Dans l'exemple ci-contre, la fenêtre sera modifiée pour avoir x_{min}=1, x_{max}=2, y_{min}=-1 et y_{max}=4.



BOXPLOT :

La fonction *boxplot* permet d'obtenir les diagrammes en boîte d'une ou plusieurs séries de valeurs. Ces séries doivent être contenues dans la même liste.

Pour obtenir le diagramme en boîte des deux séries de valeurs, nous écrirons : *boxplot([[1,2,2,2,2,3,-1,5,2],[2,3,6,7,0,0,0]])*

Remarque : la fonction *boxplotFR* permet de tracer des diagrammes en boîte avec les calculs à la française.

Retrouvez toutes nos ressources pédagogiques sur www.casio-education.fr