COURS 7 : Importer des bibliothèques et tracer des courbes – **1h**

Python est un langage qui permet de réaliser des programmes, des jeux, des mathématiques, etc…

De base, l'interpréteur de Python n'intègre pas toutes ces fonctionnalités (cela réduirait sa vitesse de calcul par une surcharge de la mémoire).

Il est donc nécessaire de faire appel aux bibliothèques que l'on souhaite utiliser.
Chacune de ces bibliothèques comporte une multitude de fonction.
Il existe donc de nombreuses bibliothèques, en voici quelques unes.

Exemple	**Bibliothèques**
Manipuler des bases de données	MySQLdb Gadfly kinterbasdb
Interface graphique	Tix Tkinter Pmw wxPython Turtlte
Manipuler des images matricielles	PIL
Tracé des graphiques	matplotlib
Mathématiques (cosinus, sinus, tangente, racine carrée…)	Math SciPy NumPy
Jeu / Imagerie 3D	VPython Pygame
Ports de communication informatique	pySerial pyParallel

1ère méthode pour importer une bibliothèque :

Le programme doit commencer par la ligne suivante :

```
1  from numpy import *
```

On utilise la bibliothèque « numpy » en important toutes les fonctions de cette bibliothèque.

On pourrait aussi choisir de n'importer qu'une seule fonction.

```
1  from numpy import linspace
```

2ème méthode pour importer une bibliothèque :

Si on souhaite utiliser plusieurs fonctions d'une bibliothèque sans la charger entièrement (certaines sont trop lourdes), on utilise une commande raccourci.

```
1  import numpy as np
```

On souhaite tracer des courbes, on va donc utiliser la bibliothèque matplotlib.

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  def f1(x):
5    y=3*x
6    return y
7
8  x=np.linspace(-10,10,10)
9  y=f1(x)
10
11 plt.plot(x,y,"r--x",label="y=3x")
12
13 plt.xlim(-20,20)
14 plt.ylim(-40,40)
15 plt.grid(True)
16
17 plt.title("fonction linéaire")
18 plt.xlabel("x")
19 plt.ylabel("y")
20
21 plt.savefig("mon image")
22
23 plt.legend()
24 plt.show()
```

Recopier les lignes de codes ci-dessus.

Étude du programme :

Le programme consiste à tracer la courbe représentative de la fonction y=3x

Ligne 1 :
On va choisir de travailler avec la bibliothèque « **matplotlib**» sans la charger entièrement avec le module de tracé «**pyplot** ».

Pour utiliser les fonctions qui nous intéressent de cette bibliothèque, nous utiliserons un raccourci d'appel que l'on nommera « **plt** » (on peut lui donner le nom que l'on souhaite).

Ligne 2 :
Tout comme la ligne 1, on utilisera donc une ou des fonctions de la bibliothèque « **numpy**» sans la charger entièrement, l'appel se fera par le raccourci que l'on nommera «**np**».

Ligne 4, 5 et 6 :
On défini la fonction (Voir cour 6).

Ligne 8 :
La fonction «**linspace**» génère une liste de 10 points compris entre une valeur de Xmin et Xmax
`(-10,10,10)` = (Xmin, Xmax, nombre de points). Cette liste de points est placée dans la variable **x**

Il est possible de voir cette liste en ajoutant à la suite de ligne 8 : print(x)

Ligne 9 :
Linspace calcul la valeur de y en par rapport à la fonction définie en ligne 4.
Les points de cette liste sont calculés en fonction de la liste des valeurs de x, de la ligne 8.

Il est possible de voir cette liste en ajoutant à la suite de ligne 9 : print(y)

Ligne 11 :
A l'aide du raccourci **plt**, on appel la fonction **plot** de **mathlib**.

```
plt.plot(x,y,"r--x",label="y=3x")
```

Affichage des listes de points placées dans les variables x et y

Si on souhaite ajouter une légende à la courbe

On indique la couleur : **r (rouge) b (bleu) g (vert) y (jaune) k (noir)**

On indique le type de trait pour la courbe : **- - (ligne de tiret) . (pointillé) - (ligne continu) -. (point tiret)**
On indique le type de représentation du point : **x (croix) s (carré) d (losange) o (rond)**

Ligne 13 :
On définit l'affichage des valeurs min et max sur l'axe des abscisses.

Ligne 14 :
On définit l'affichage des valeurs min et max sur l'axe des ordonnées.

Si les lignes 13 et 14 n'existent pas, python crée automatiquement les valeurs des axes en fonction de la courbe.

Ligne 15 :
La commande **grid(true)** permet l'affichage de la grille.

Vous ajouterez devant cette ligne le symbole #, et relancer le programme

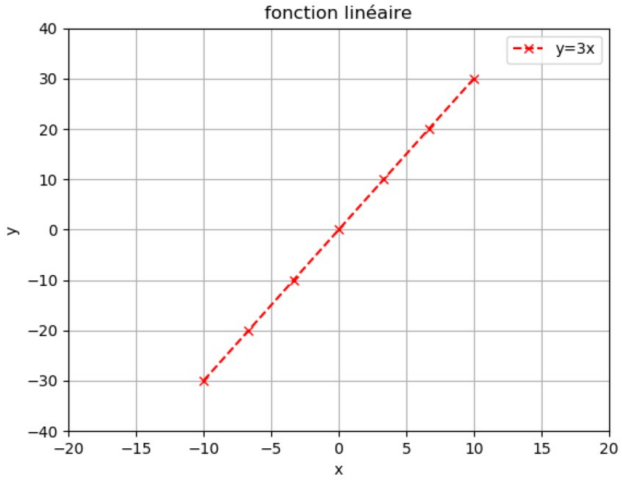
Ligne 17 :
On appel la fonction qui permet d'afficher un titre au graphique

Ligne 18 et 19 :
Permet de nommer les axes des abscisses et des ordonnées.

Ligne 21 :
Permet d'enregistrer automatiquement la courbe sur le bureau de l'ordinateur.

Ligne 23 :
Permet l'affichage du titre de la fenêtre (label) définie en ligne 11.

Ligne 24 :
Permet l'affichage de la fenêtre.



Exercice :

Vous devez écrire un programme vous permettant de tracer les fonctions trigonométriques suivantes :

Y1= cos (x)
Y2= sin(x)

Vous programmerez un affichage de 20 points compris entre 0 et 2 π (voir ligne 8 du programme ci-dessus)

Vous tracerez les courbes avec les couleurs de votre choix et des points différents.

Aide :

Dans la bibliothèque de numpy, la fonction :

- cosinus se nomme cos
- sinus se nomme sin
- π se nomme pi

Pour utiliser ces fonctions, il faudra les appeler par le raccourci **np.**

Exemple : y=np.cos(x)

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  def f1(x):
5      y=np.cos(x)
6      return y
7
8  def f2(x):
9      y=np.sin(x)
10     return y
11
12 x=np.linspace(0,2*np.pi,20)
13
14 y1=f1(x)
15 y2=f2(x)
16
17 plt.plot(x,y1,"g-o",label="y=cos(x)")
18 plt.plot(x,y2,"b-.d",label="y=sin(x)")
19
20 plt.grid(True)
21
22 plt.title("fonction trigonométrique")
23 plt.xlabel("x")
24 plt.ylabel("y")
25
26 plt.savefig("mon image2")
27
28 plt.legend()
29 plt.show()
```