

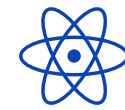


Langage Python

Tracer et représenter des données avec Matplotlib.

ACTIVITÉ

LANGAGE PYTHON TRACÉ DE COURBES



Prof-TC

SNT

1 - TRACÉ D'UNE COURBE AVEC MATPLOTLIB

En utilisant Spyder recopier et exécuter le programme ci-dessous.

```
import numpy as np
import matplotlib.pyplot as plt

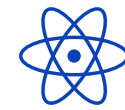
x = np.array([0, 1.01, 2.02, 2.99, 3.98]) # Données en abscisse
y = np.array([10.02, 7.96, 6.03, 4.04, 2.01]) # Données en ordonnée

plt.plot(x, y) # Tracé de la courbe
plt.show() # Affichage de la courbe
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg1.py.

Quelques remarques :

- La collection pyplot du module matplotlib est importée avec l'alias plt.
- La fonction plot() trace la courbe $y=f(x)$ à partir des tableaux x et y.
- La fonction show() appelée en dernier affiche la fenêtre graphique.



2 - AJOUTER UN TITRE, DES LÉGENDES ET UNE GRILLE

Modifier le programme en ajoutant un titre, une légende et une grille.

```
import numpy as np
import matplotlib.pyplot as plt

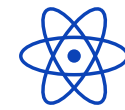
x = np.array([0, 1.01, 2.02, 2.99, 3.98])    # Données en abscisse
y = np.array([10.02, 7.96, 6.03, 4.04, 2.01]) # Données en ordonnée

plt.plot(x, y, 'x')      # Tracé de la courbe
plt.title('Mom titre')   # Ajout d'un titre
plt.xlabel('x')          # Nom de la grandeur en abscisse
plt.ylabel('y')          # Nom de la grandeur en ordonnée
plt.grid()               # Ajout d'une grille
plt.show()               # Affichage
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg2.py.

Quelques remarques :

- Le paramètre x dans plot() met en évidence les points avec des croix sans les relier par des segments de droite.
- Les fonctions title(), xlabel() et ylabel() ajoutent un titre et les légendes sur les axes.
- La fonction grid() ajoute une grille.

**3 - DÉFINIR L'ÉCHELLE DU GRAPHIQUE**

Modifier le programme afin de définir l'échelle.

```
import numpy as np
import matplotlib.pyplot as plt

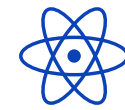
x = np.array([0, 1.01, 2.02, 2.99, 3.98]) # Données en abscisse
y = np.array([10.02, 7.96, 6.03, 4.04, 2.01]) # Données en ordonnée

plt.plot(x, y, 'x') # Tracé de la courbe
plt.title('Mon titre') # Ajout d'un titre
plt.xlabel('x') # Nom de la grandeur en abscisse
plt.xlim(0,4) # Echelle sur l'axe des x
plt.ylabel('y') # Nom de la grandeur en ordonnée
plt.ylim(0,11) # Echelle sur l'axe des y
plt.grid() # Ajout d'une grille
plt.show() # Affichage
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg3.py.

Remarque :

- Les fonctions xlim() et ylim() fixent respectivement les limites des axes des abscisses et des ordonnées.

**4 - TRACER UNE COURBE À PARTIR D'UNE FONCTION**

En utilisant Spyder recopier et exécuter le programme ci-dessous.

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100) # Création d'un tableau de valeurs pour x
y = 10*np.sin(x)           # Calcul de y à partir de la fonction mathématique

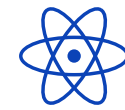
plt.plot(x, y)              # Tracé de la courbe
plt.title("A partir d'une fonction") # Titre
plt.xlabel('x')             # Légende abscisse
plt.ylabel('y')             # Légende ordonnée
plt.grid()                  # Ajout d'une grille
plt.show()                  # Affichage
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg4.py.

QUESTION Quel type de courbe trace ce programme ? _____

Remarque :

- La fonction linspace() crée une série de valeurs régulièrement espacées pour calculer puis tracer $y=f(x)$.

**5 - TRACER DEUX COURBES**

Modifier le programme afin de tracer deux courbes.

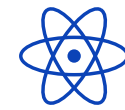
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(1, 10, 100) # Création d'un tableau de valeurs pour x
y1 = 10*np.sin(x)           # Calcul de y1
y2 = 6*np.sin(x-1)          # Calcul de y2

plt.plot(x, y1, label='10.sin(x)') # Tracé de la courbe y1 avec texte légende
plt.plot(x, y2, label='6.sin(x-1)') # Tracé de la courbe y1 avec texte légende
plt.title('Ma première courbe')    # Titre
plt.xlabel('x')                     # Légende abscisse
plt.ylabel('y')                     # Légende ordonnée
plt.legend()                        # Ajout de la légende
plt.grid()                          # Ajout d'une grille
plt.show()                          # Affichage
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg5.py.

QUESTION Quel types de courbes trace ce programme ?



6 - TRACER UN HISTOGRAMME

En utilisant Spyder recopier et exécuter le programme ci-dessous.

```
import numpy as np
import matplotlib.pyplot as plt

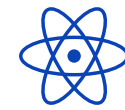
x = [5, 6, 4, 7, 6, 7, 6, 8, 6, 5, 6, 5, 3, 9, 4, 6, 5, 8, 7, 6]

plt.hist(x, range=(0, 10), bins=10, rwidth = 0.8, align='left')
plt.show()
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg6.py.

Quelques remarques :

- range=(0, 10) fixe les limites de la plage d'étude du tableau de données.
- bins=10 est le nombre d'intervalles dans la plage d'étude.
- rwidth=0.8 fixe la largeur des barres pour une meilleure visibilité.
- align='left' centre les barres sur la borne gauche des intervalles.



7 - FRÉQUENCES, MOYENNE ET ÉCART TYPE

Modifier le programme afin d'afficher les valeurs, leurs fréquences, la valeur moyenne et l'écart type.

```
import numpy as np
import matplotlib.pyplot as plt

x = [5, 6, 4, 7, 6, 7, 6, 8, 6, 5, 6, 5, 3, 9, 4, 6, 5, 8, 7, 6]

freq, valx, opt = plt.hist(x, range=(0,10),bins=10,rwidth = 0.8, align='left')
plt.show()

print("Valeur moyenne = ", np.mean(x))
print("Ecart type = ",np.std(x).round(2))
print('frequence = ',freq)
print('valeurs x = ',valx)
```

Sauvegarder ce programme dans le dossier nommé SciPython sous le nom prg7.py.

Remarque :

- Les fonctions de Numpy mean() et std() calculent respectivement la valeur moyenne et l'écart type.