



Département: Mathématiques & Informatique

Séance 11:

Analyse de données en Python

Module : matplotlib

Licence : Physique Chimie

Pr: Youssef Ouassit

1. Introduction
2. Tracer de courbes
3. Travailler avec des Sous-Graphiques
4. Visualisation des fonctions à deux variables

Matplotlib est une bibliothèque Python puissante et flexible pour la visualisation de données en 2D. Elle permet de créer divers types de graphiques, comme des courbes, des diagrammes en barres, des nuages de points et des histogrammes.

Matplotlib est une bibliothèque de visualisation de données multiplateforme construite sur des tableaux NumPy et conçue pour fonctionner avec l'ensemble plus large de SciPy. Elle a été introduite par John Hunter en 2002.

Matplotlib comprend plusieurs types de tracés comme les courbes, les barres, les nuages de points, les histogrammes, etc.

Les modules de la bibliothèque matplotlib:

- matplotlib.pyplot
- matplotlib.figure
- matplotlib.axes
- matplotlib.patches
- matplotlib.lines
- matplotlib.text
- matplotlib.animation
- matplotlib.image
- ...

Importer les modules :

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

Introduction au module matplotlib

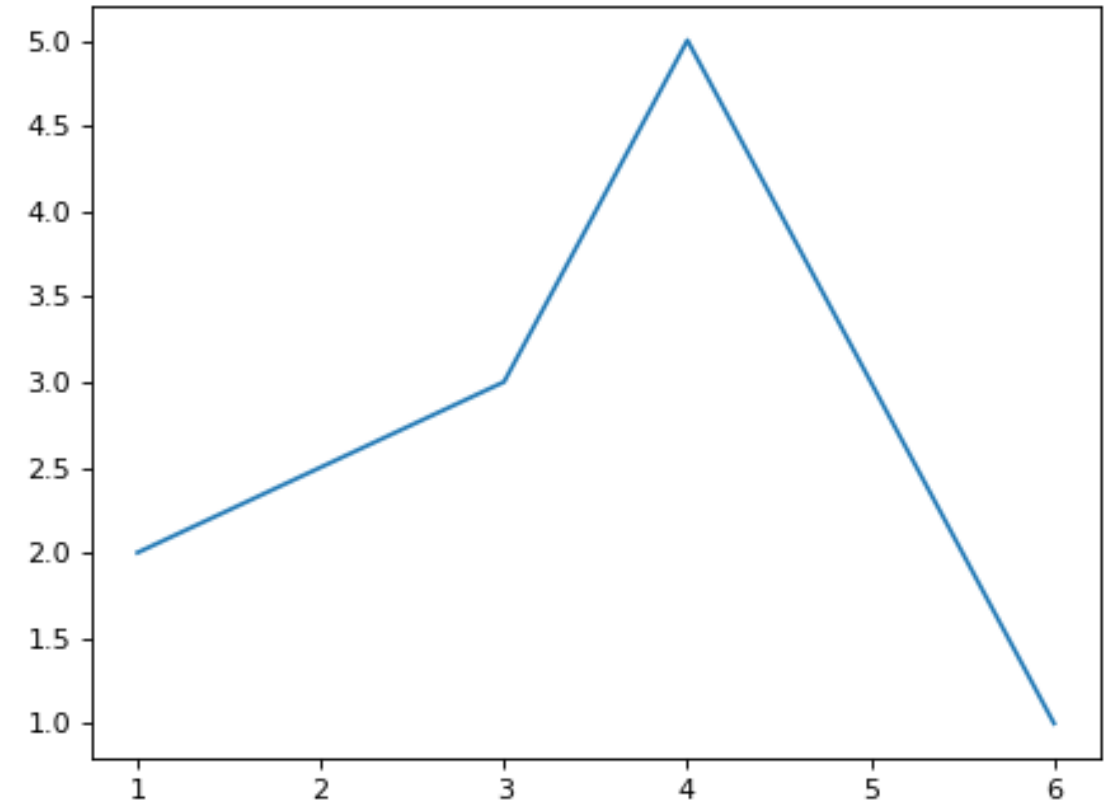
Tracer une courbe simple

Pour tracer une courbe reliant des points:

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

```
x = [1, 3, 4, 6]  
y = [2, 3, 5, 1]  
plt.plot(x , y)
```

```
plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

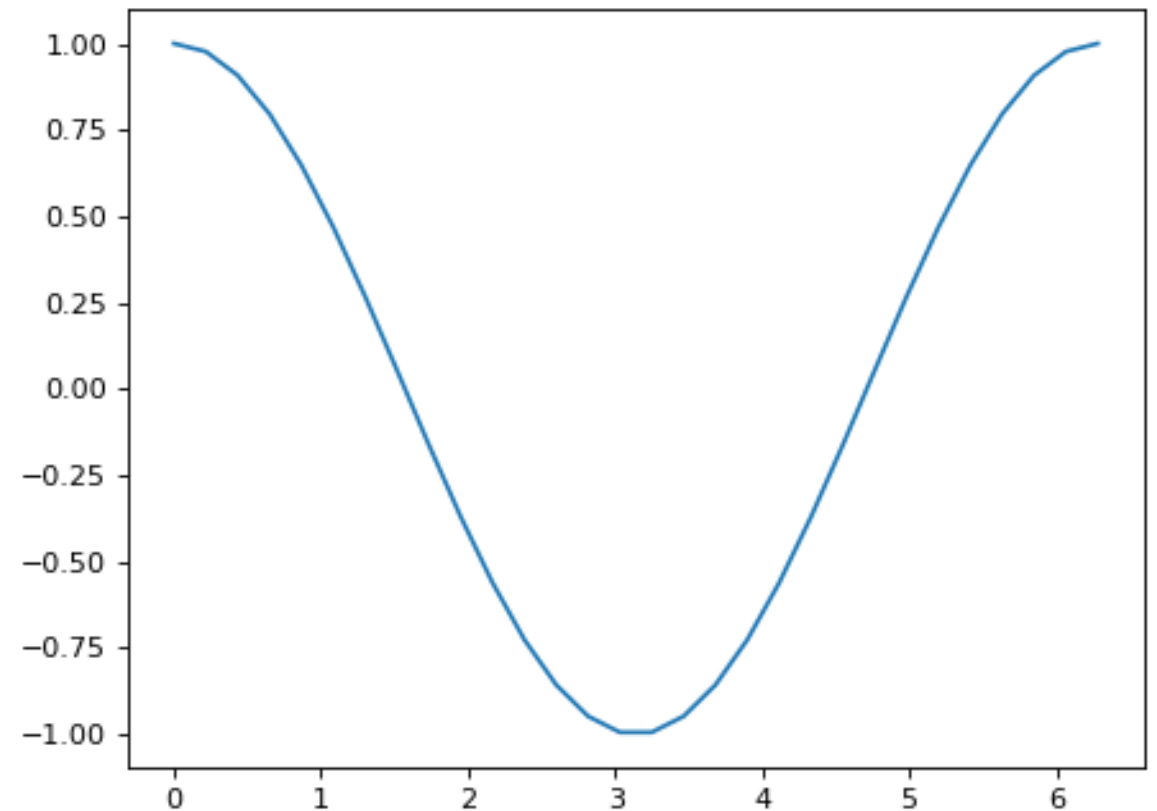
La courbe de la fonction cos :

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

x = np.linspace(0, 2*np.pi, 30)
y = np.cos(x)

plt.plot(x, y)

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

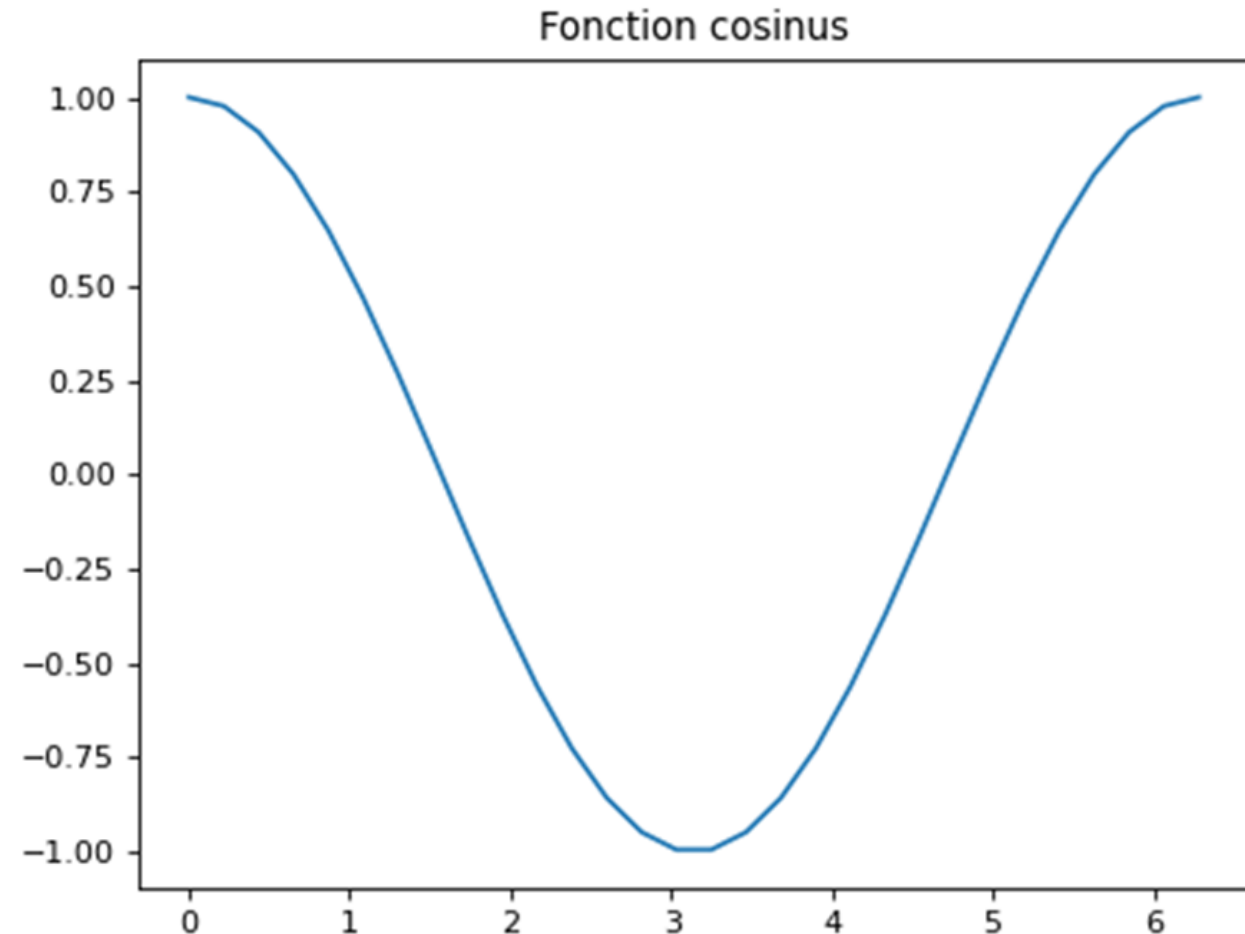
Ajouter un titre :

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

x = np.linspace(0, 2*np.pi, 30)
y = np.cos(x)

plt.plot(x, y)
plt.title("Fonction cosinus")

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

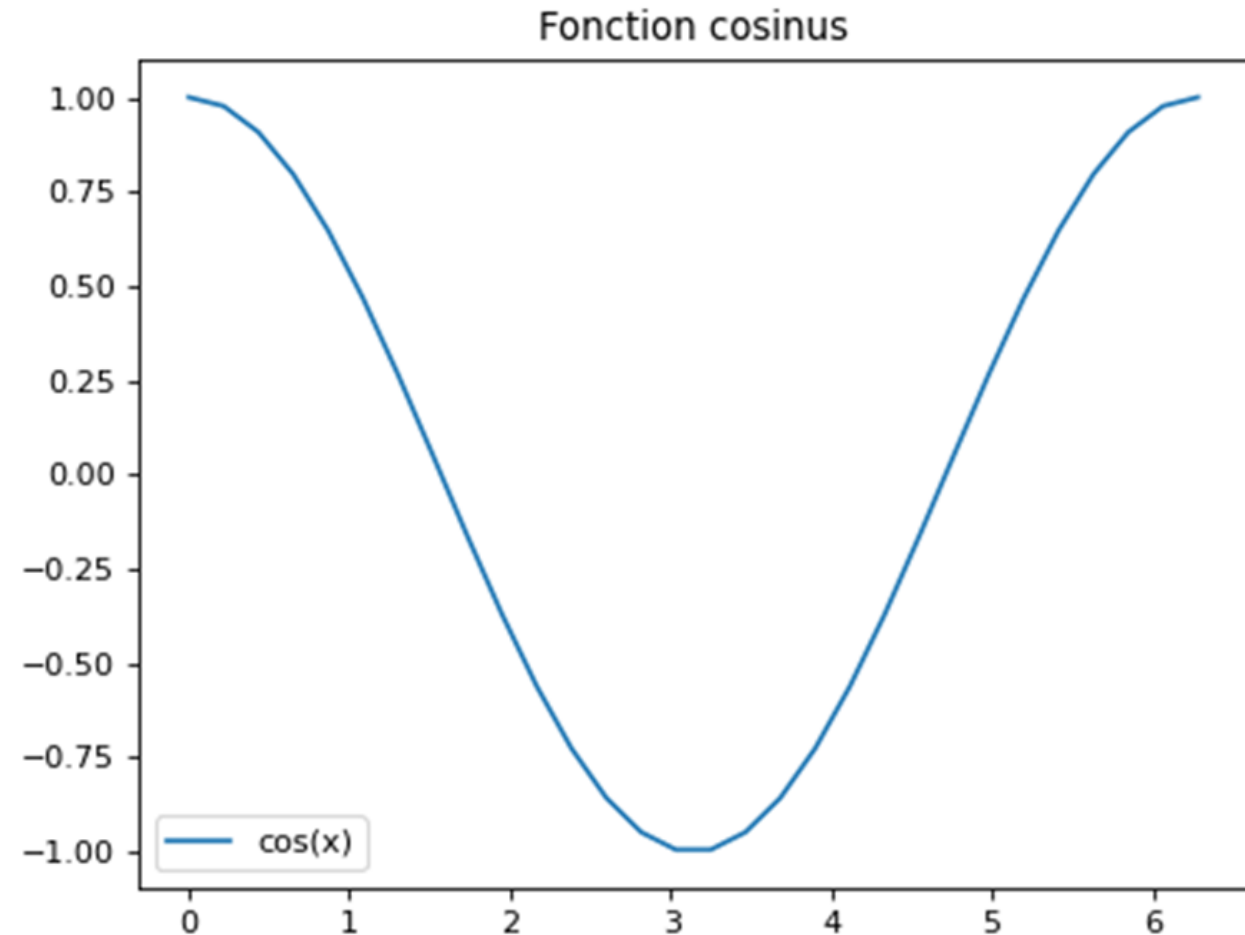
Ajouter une légende:

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

x = np.linspace(0, 2*np.pi, 30)
y = np.cos(x)

plt.plot(x, y, label="cos(x)")
plt.title("Fonction cosinus")
plt.legend()

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

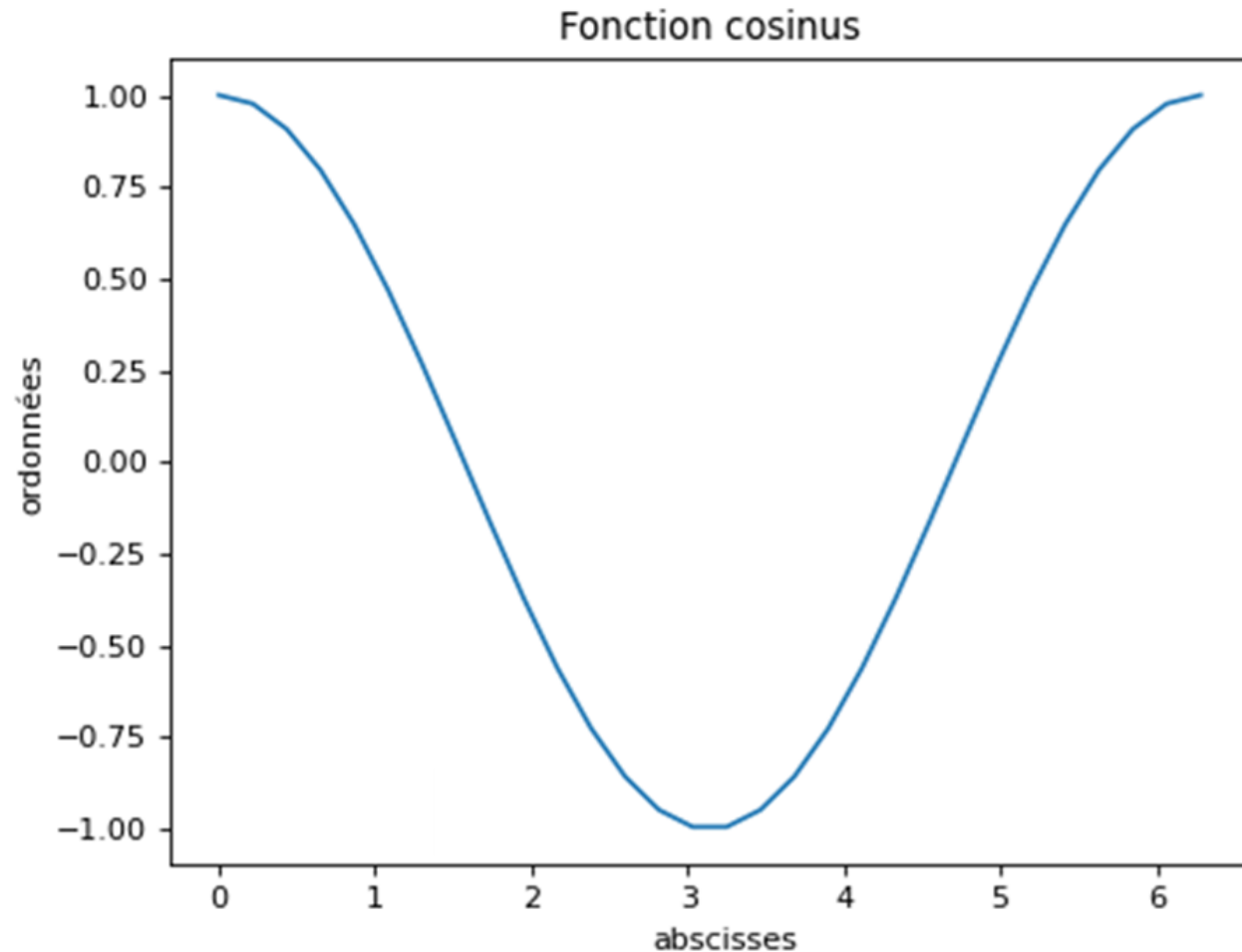
Ajouter des labels sur les axes:

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

x = np.linspace(0, 2*np.pi, 30)
y = np.cos(x)

plt.plot(x, y)
plt.xlabel("abscisses")
plt.ylabel("ordonnées")

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

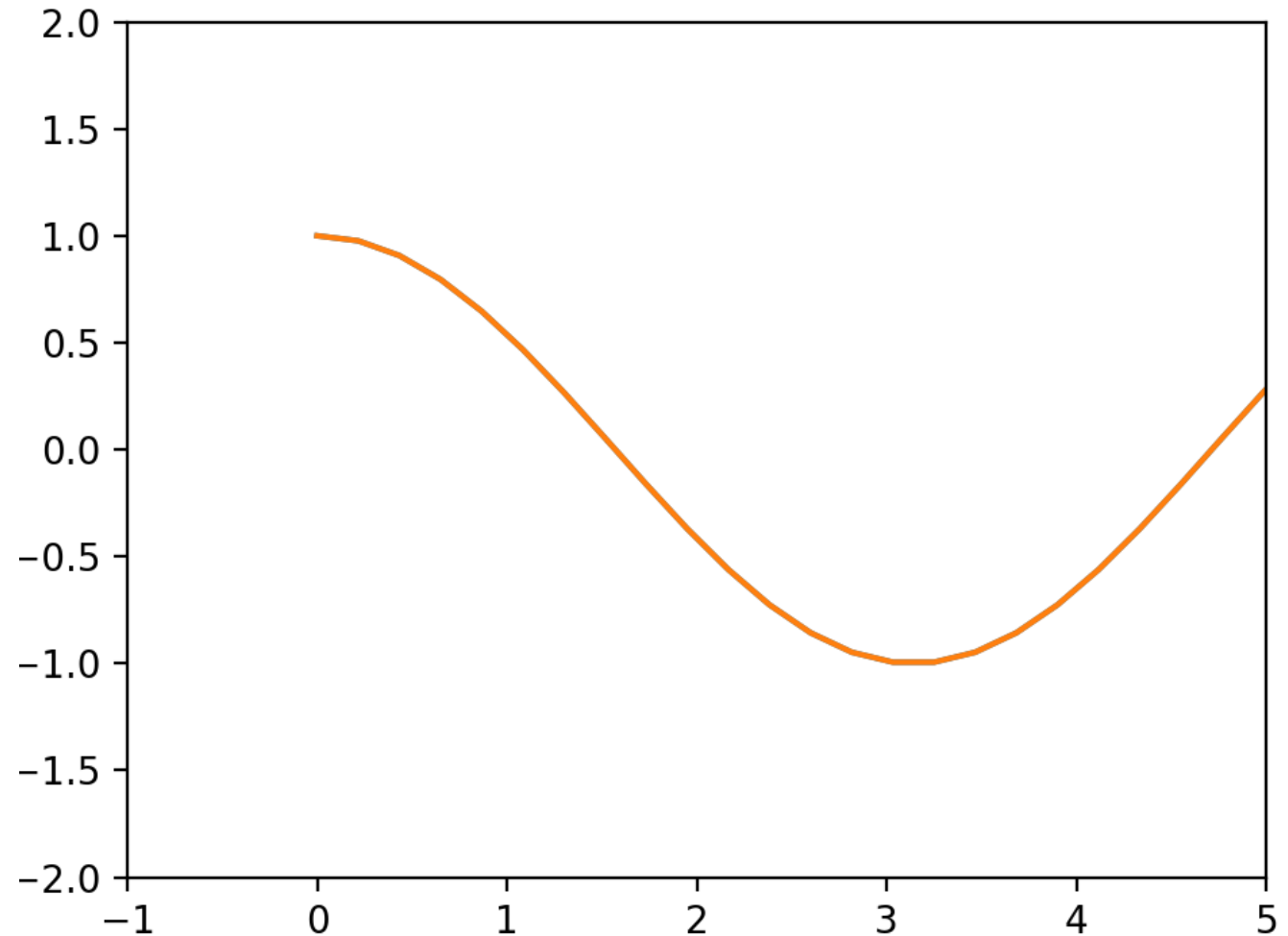
Définition du domaine des axes:

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

x = np.linspace(0, 2*np.pi, 30)
y = np.cos(x)

plt.plot(x, y)
plt.axis([-1, 5, -2, 2])

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

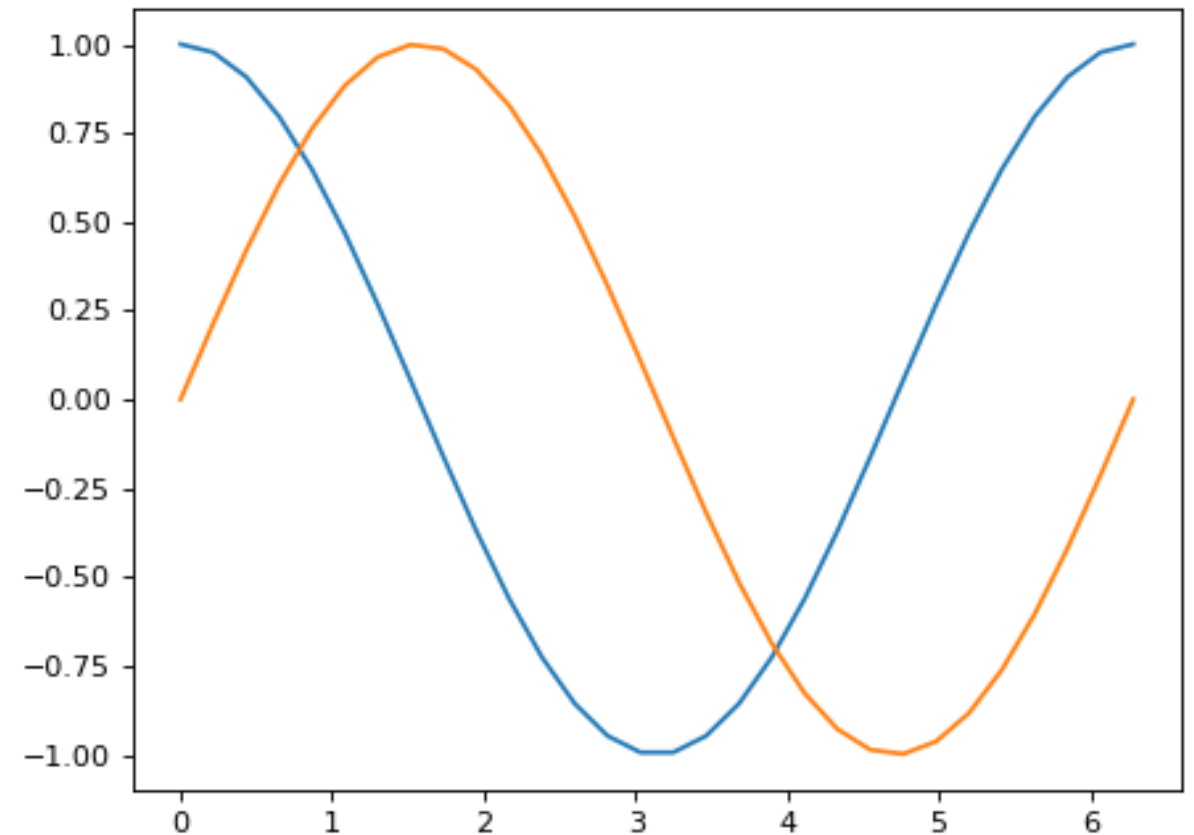
Afficher plusieurs courbes:

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

x = np.linspace(0, 2*np.pi, 30)
y1 = np.cos(x)
y2 = np.sin(x)

plt.plot(x, y1)
plt.plot(x, y2)

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

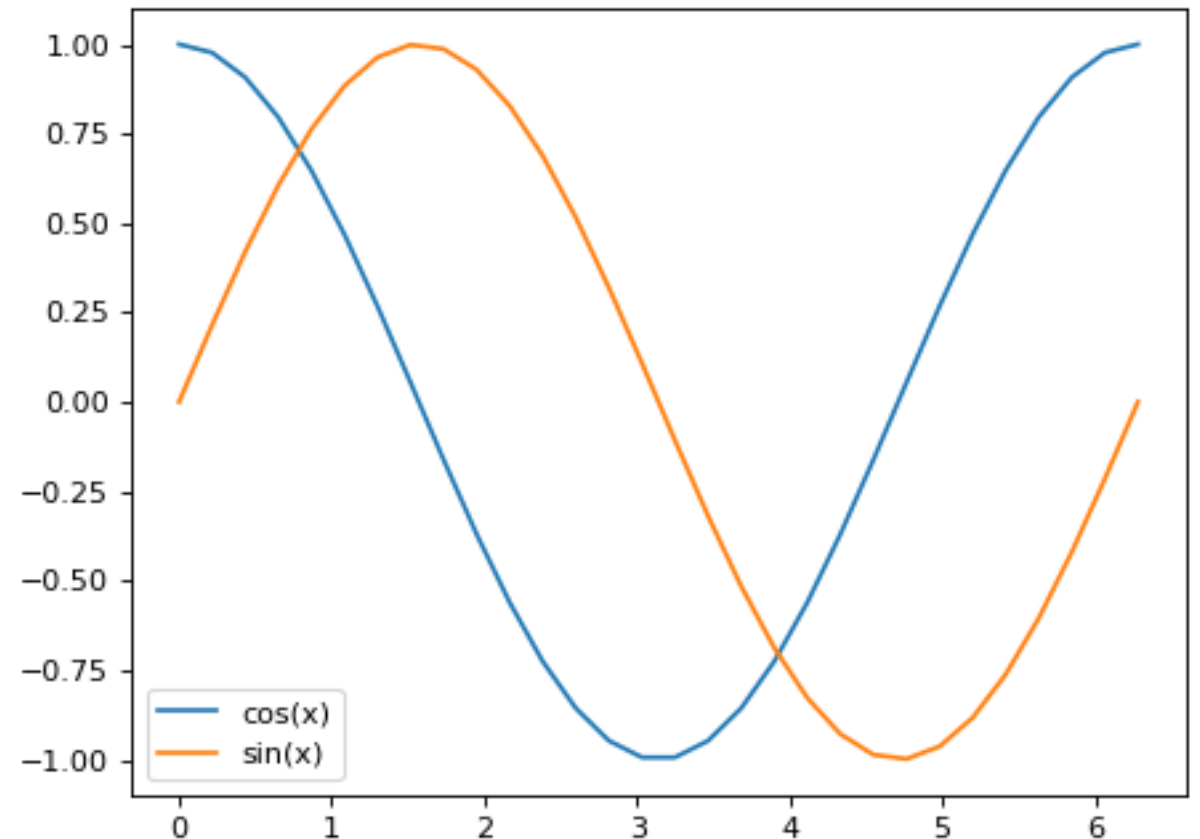
Afficher plusieurs courbes:

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

x = np.linspace(0, 2*np.pi, 30)
y1 = np.cos(x)
y2 = np.sin(x)

plt.plot(x, y1, label="cos(x)")
plt.plot(x, y2, label="sin(x)")
plt.legend()

plt.show()
```



Introduction au module matplotlib

Tracer une courbe simple

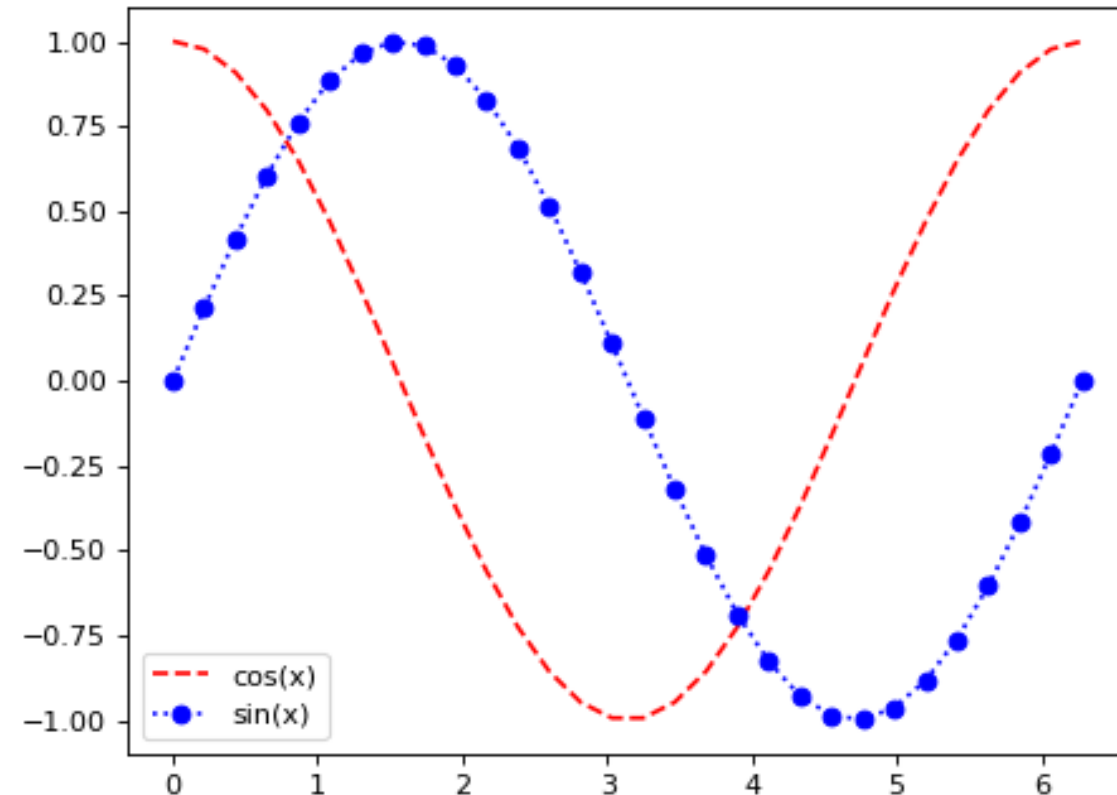
Afficher plusieurs courbes:

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

x = np.linspace(0, 2*np.pi, 30)
y1 = np.cos(x)
y2 = np.sin(x)

plt.plot(x, y1, "r--", label="cos(x)")
plt.plot(x, y2, "b:o", label="sin(x)")
plt.legend()

plt.show()
```



Couleur

Les chaînes de caractères suivantes permettent de définir la couleur :

Chaîne	Effet
b	blue
g	green
r	red
c	cyan
y	yellow
m	magenta
k	black
w	white

Style de ligne

Les chaînes de caractères suivantes permettent de définir le style de ligne :

Chaîne	Effet
-	Ligne continue
- -	Tirets
:	Ligne en pointillé
-.	Tirets points

Symbole (Marker)

Les chaînes de caractères suivantes permettent de définir le symbole (« marker ») :

Chaîne	Effet
.	point marker
,	pixel marker
o	circle marker
v	triangle_down marker
^	triangle_up marker
....	

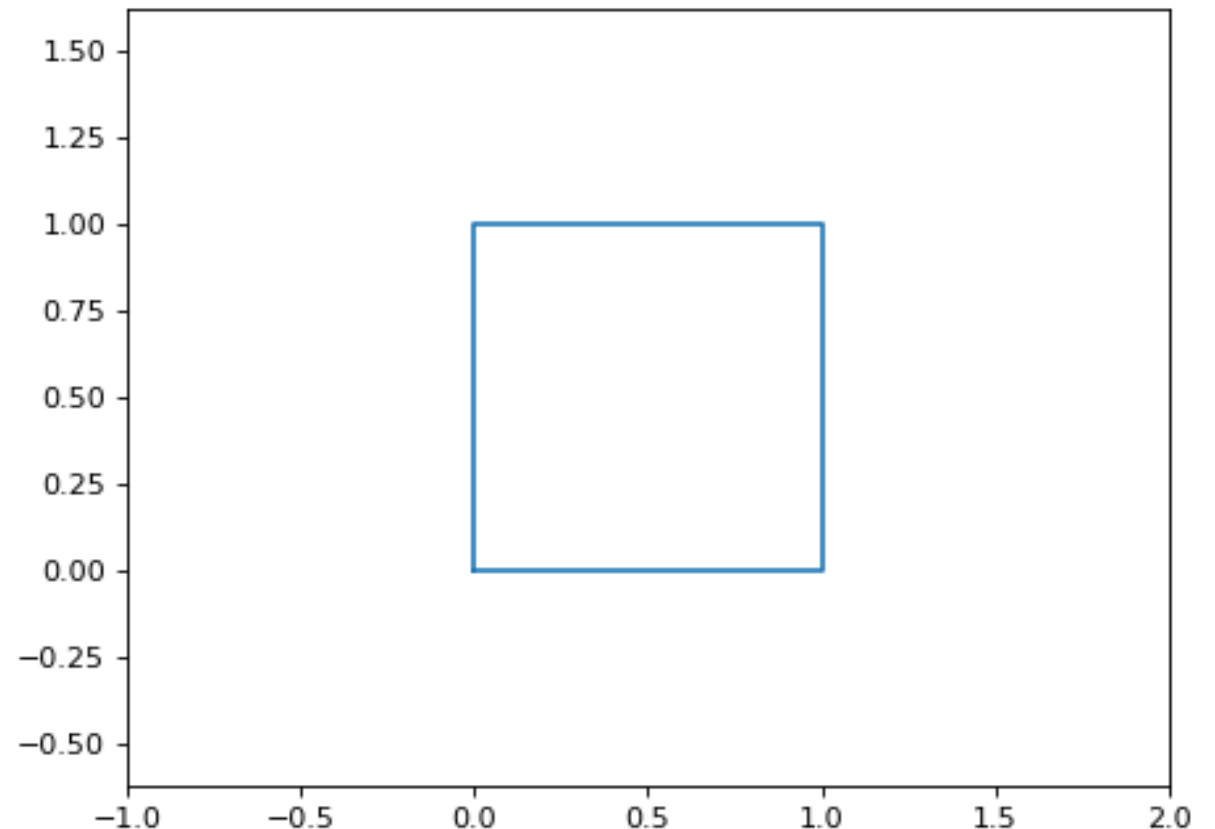
Tracer un rectangle

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

```
x = np.array([0, 1, 1, 0, 0])
y = np.array([0, 0, 1, 1, 0])
```

```
plt.plot(x, y)
plt.axis("equal")
plt.axis([-1, 2, -1, 2])
```

```
plt.show()
```



Tracer des formes

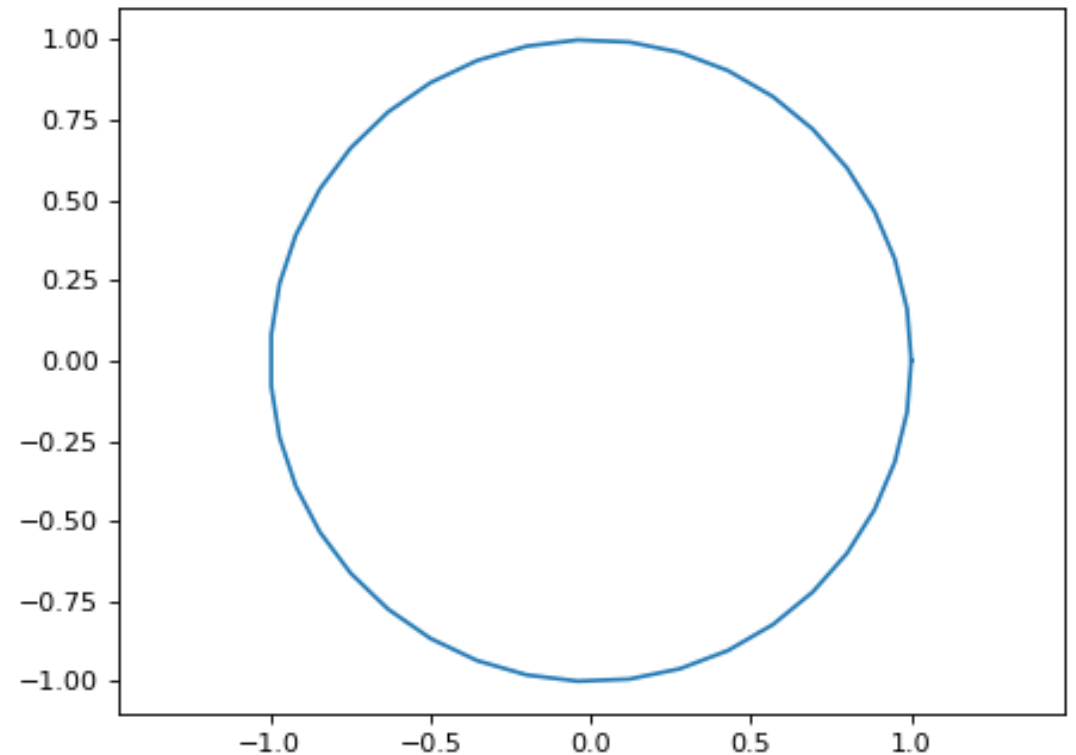
Tracer un cercle

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

theta = np.linspace(0, 2*np.pi, 40)
r = 1
x = r*np.cos(theta)
y = r*np.sin(theta)

plt.plot(x, y)
plt.axis("equal")

plt.show()
```



```
N = 100
```

```
x = np.random.rand(N)
```

```
y = np.random.rand(N)
```

```
plt.scatter(x, y, marker='o', color='red', label="Temp")
```

```
plt.show()
```

Introduction au module matplotlib

Dessiner des nuages de points

