

# o-build-a-line-graph-in-matplotlib

August 27, 2024

How to Build a Line Graph in Matplotlib

Step 1: Import the necessary libraries

Step 2: Prepare your data

Step 3: Create the line plot

Step 4: Customize your graph

Step 5: Display the graph

Import the necessary libraries

```
[ ]: import matplotlib.pyplot as plt
import numpy as np
```

## 2. prepare data

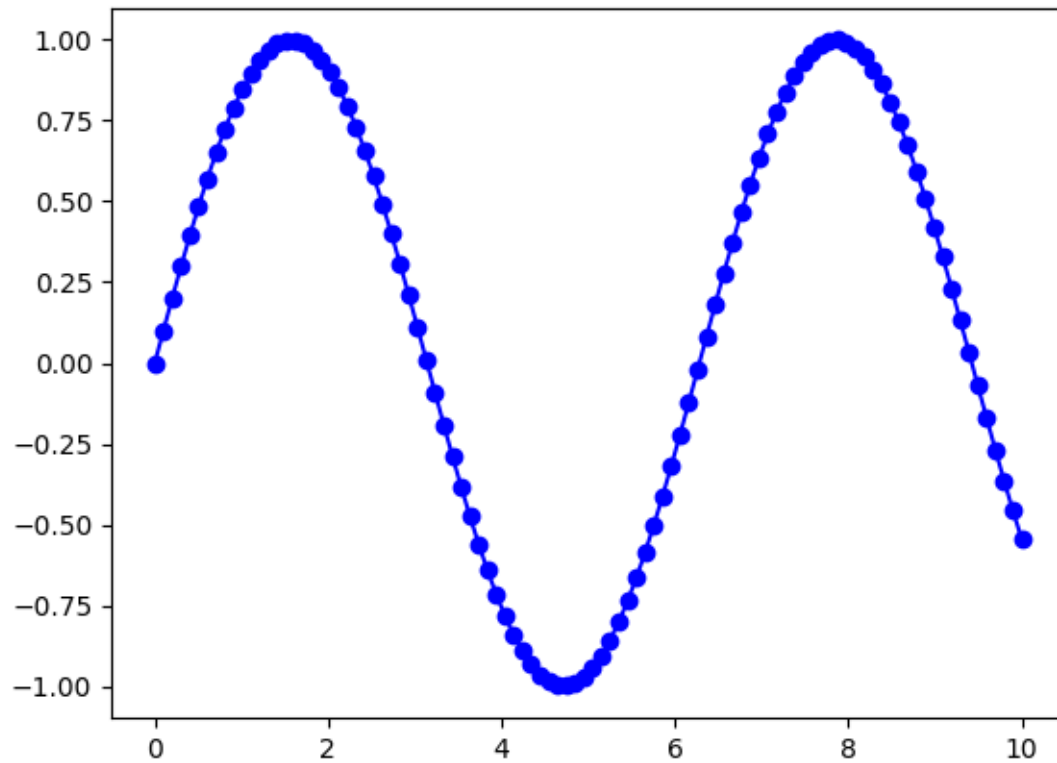
```
[ ]: # sin(x) graph
x = np.linspace(0, 10, 100)
y = np.sin(x)
```

```
[ ]: # cos(x) graph
x = np.linspace(0,10,100)
y = np.cos(x)
```

creating the plot

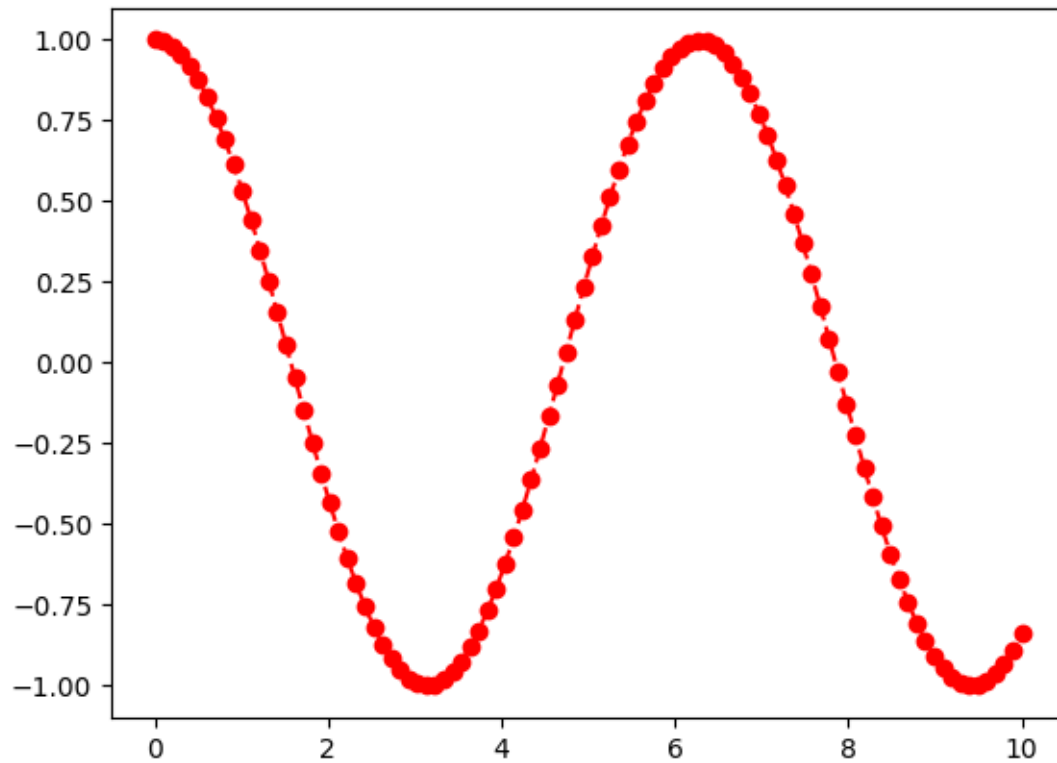
```
[ ]: plt.plot(x,y,label = 'sine Wave', color='blue',linestyle = '-', marker='o')
```

```
[ ]: [ <matplotlib.lines.Line2D at 0x7bc9fb985870>]
```



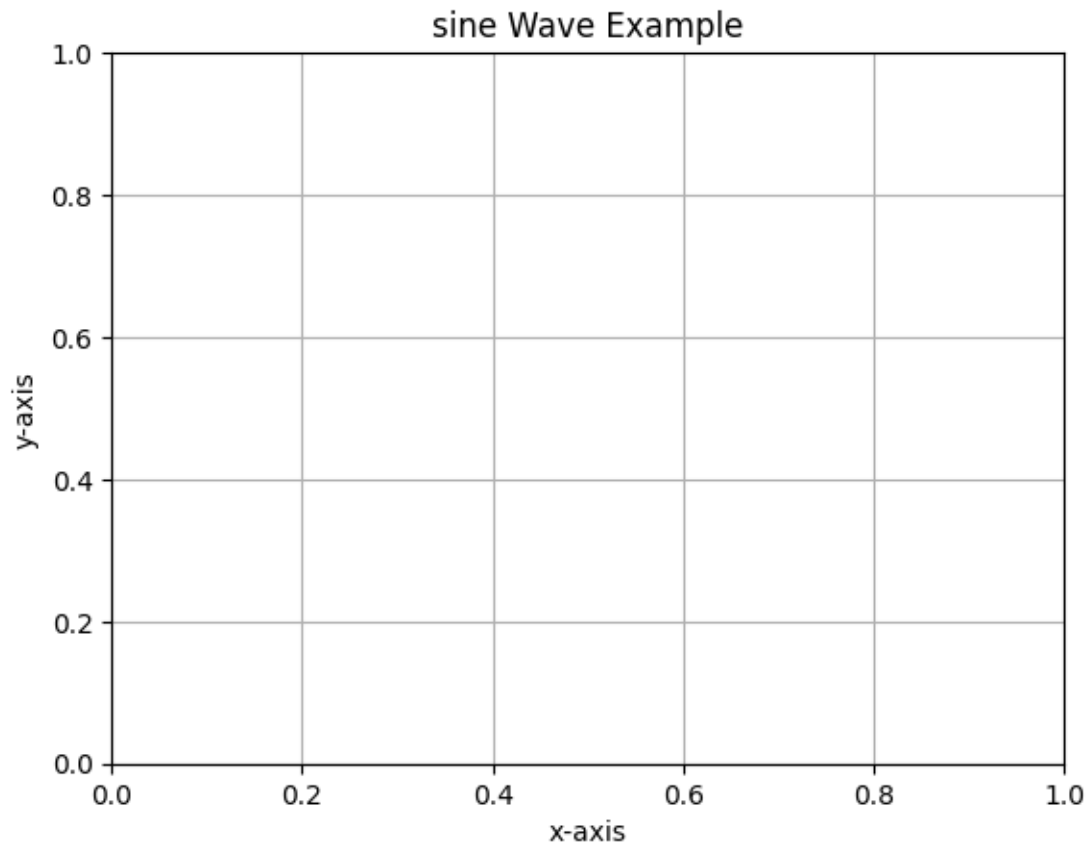
```
[ ]: # cos(x) graph
plt.plot(x,y, label = 'cos Wave', color = 'Red', linestyle='--', marker = 'o')
```

```
[ ]: [<matplotlib.lines.Line2D at 0x7bc9fba95c60>]
```



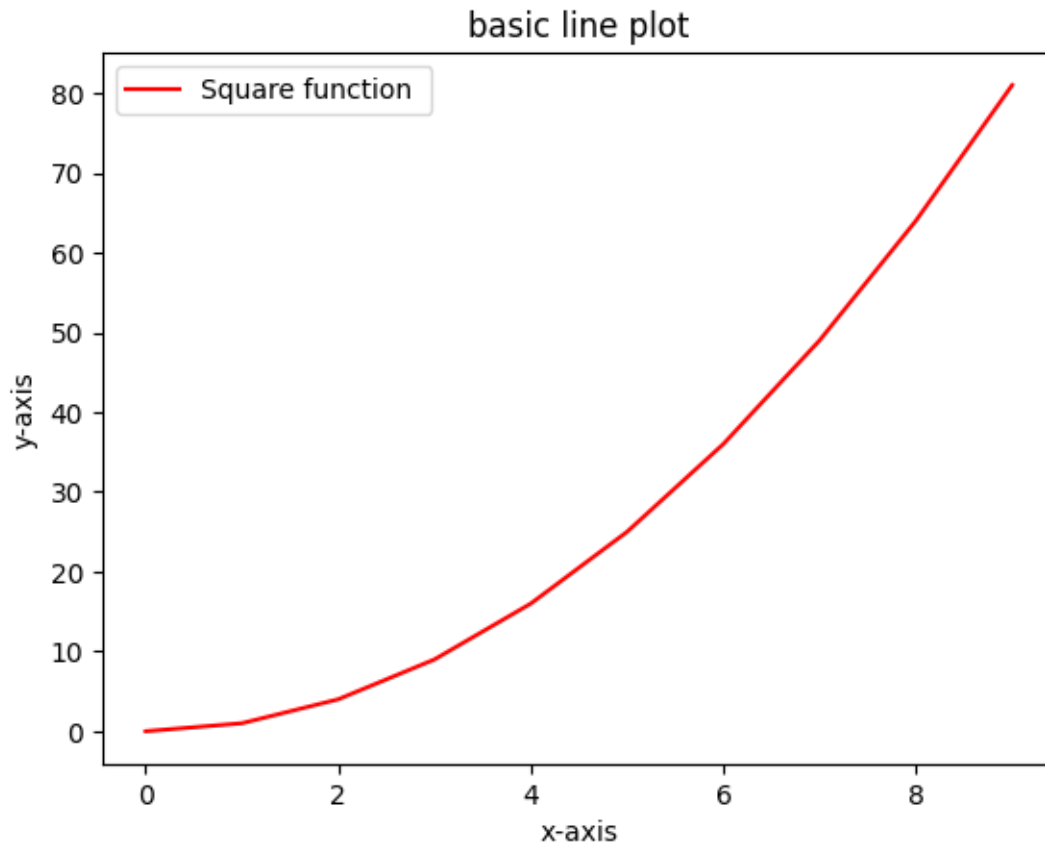
customizing the graph

```
[ ]: plt.title('sine Wave Example')
      plt.xlabel('x-axis')
      plt.ylabel('y-axis')
      plt.grid(True)
      plt.show()
```



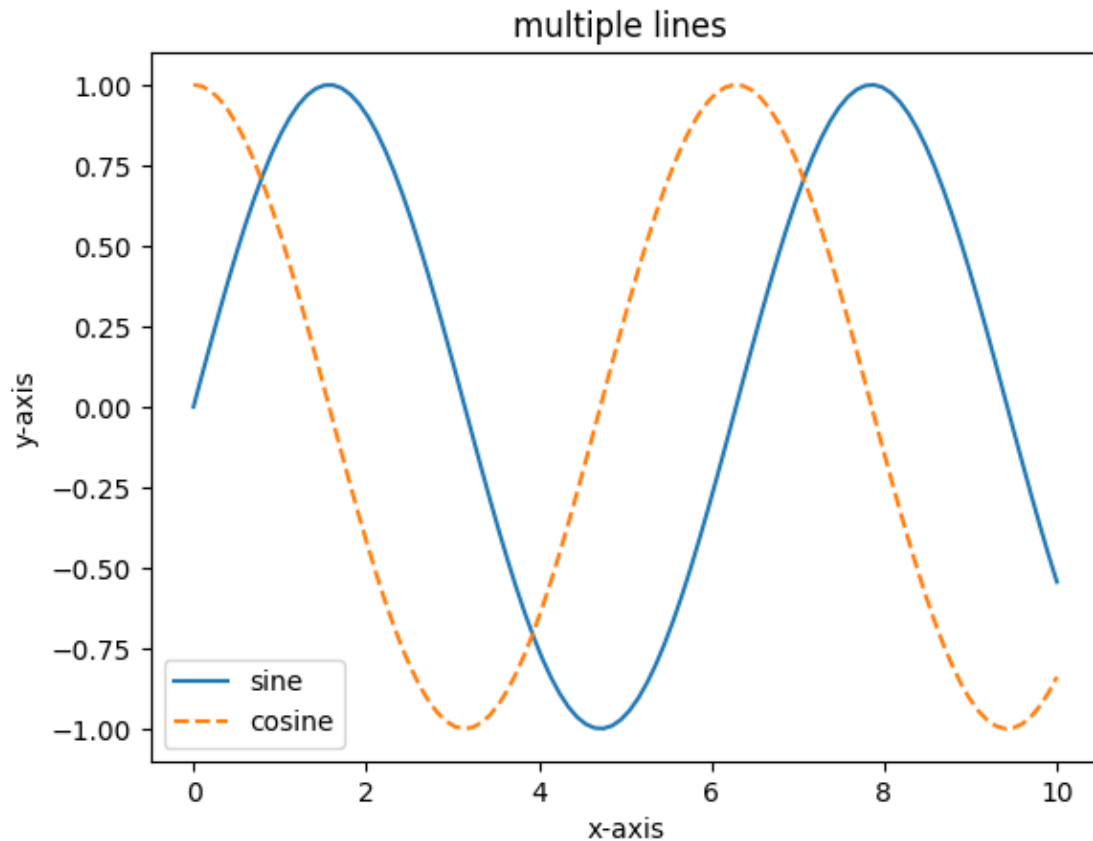
Example - 1 basic plot

```
[ ]: x = np.arange(0,10,1)
      y= x**2
      plt.plot(x,y, label = 'Square function ', color = 'red')
      plt.title('basic line plot')
      plt.xlabel('x-axis')
      plt.ylabel('y-axis')
      plt.legend()
      plt.show()
```



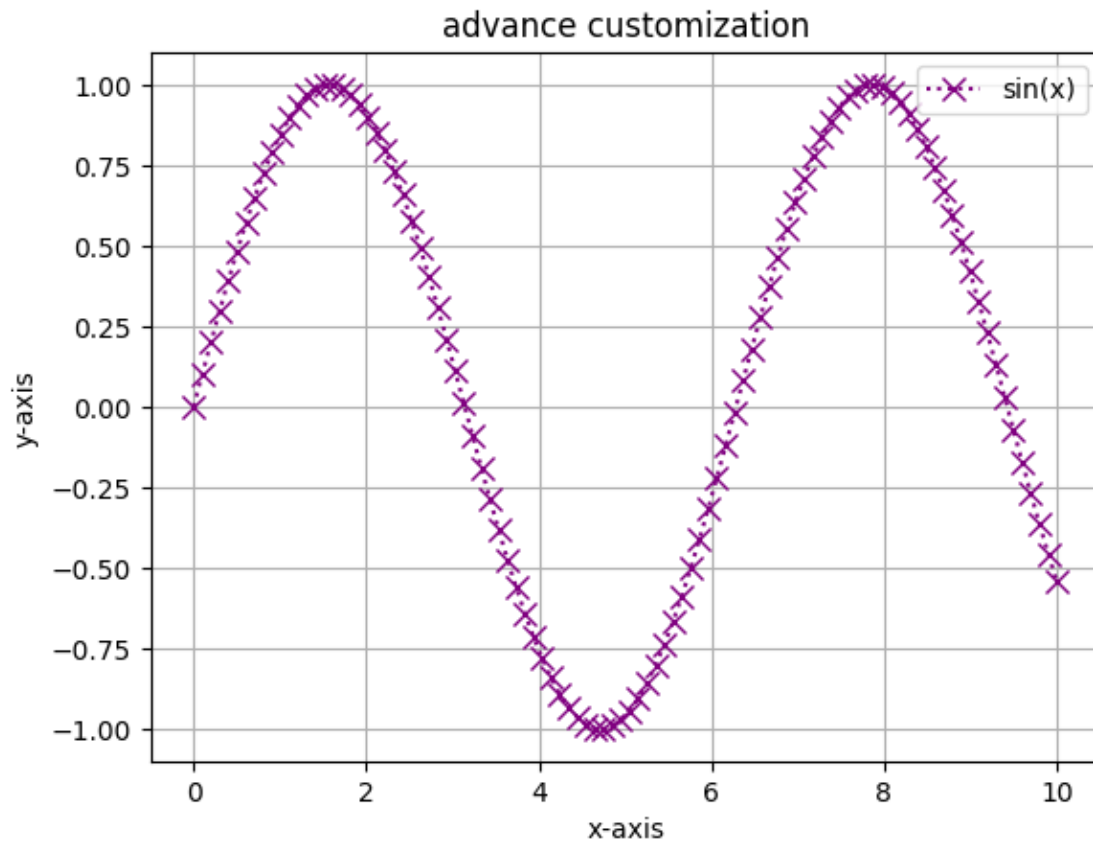
ex- 2 multiple lines plot

```
[ ]: x= np.linspace(0,10,100)
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot(x,y1,label = 'sine')
plt.plot(x,y2,label = 'cosine', linestyle = '--')
plt.title('multiple lines ')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.legend()
plt.show()
```



ex-3 Advanced customization of a graph

```
[ ]: x = np.linspace(0,10,100)
y = np.sin(x)
plt.plot(x,y, label = 'sin(x)', color='purple',linestyle=':',marker='x',
        markersize=8)
plt.title('advance customization')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.legend()
plt.grid(True)
plt.savefig('advanced_plot.png')
plt.show()
```

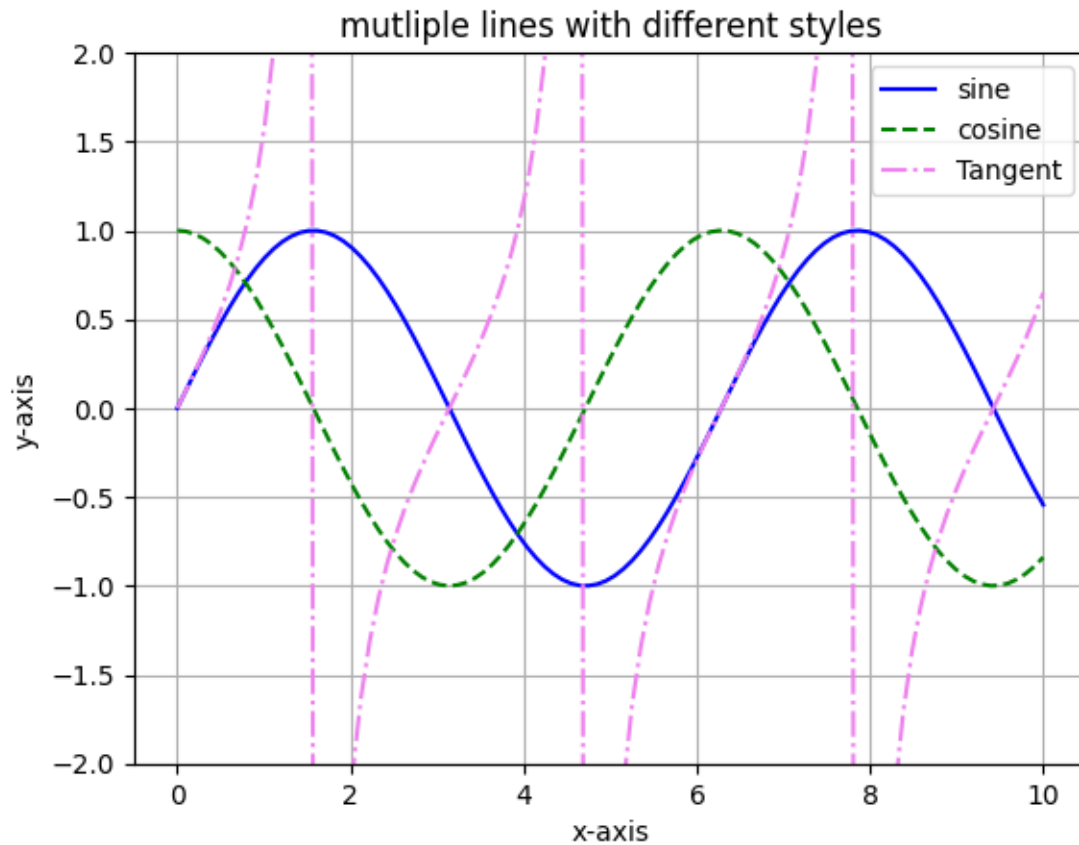


ex-4 plotting multiple lines with different styles

```
[ ]: x = np.linspace(0,10,100)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = np.tan(x)

# create a plotting
plt.plot(x, y1, label = 'sine', color='blue',linestyle='-')
plt.plot(x, y2, label = 'cosine', color='green',linestyle='--')
plt.plot(x, y3, label = 'Tangent', color='violet',linestyle='-.')

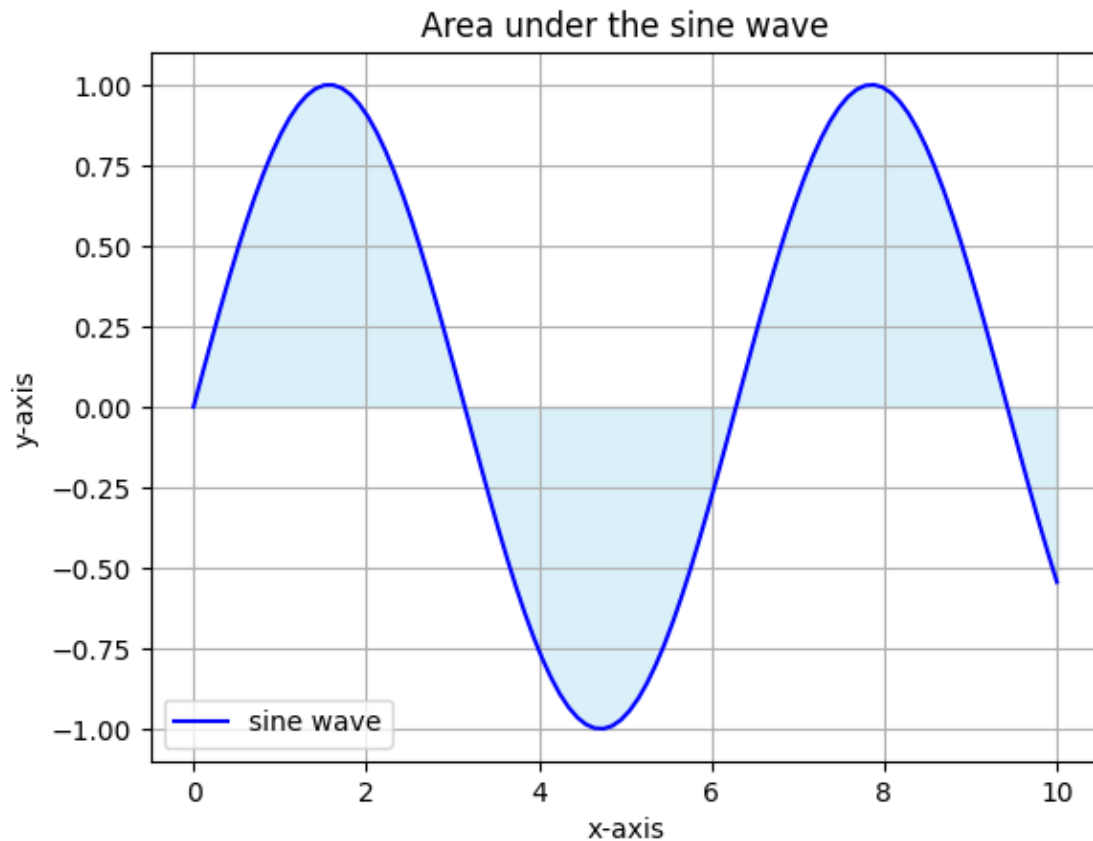
plt.title('mutliple lines with different styles')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.ylim(-2,2) # limit y- axis for better visualization
plt.legend()
plt.grid(True)
plt.show()
```



ex-5 filling area under the curve

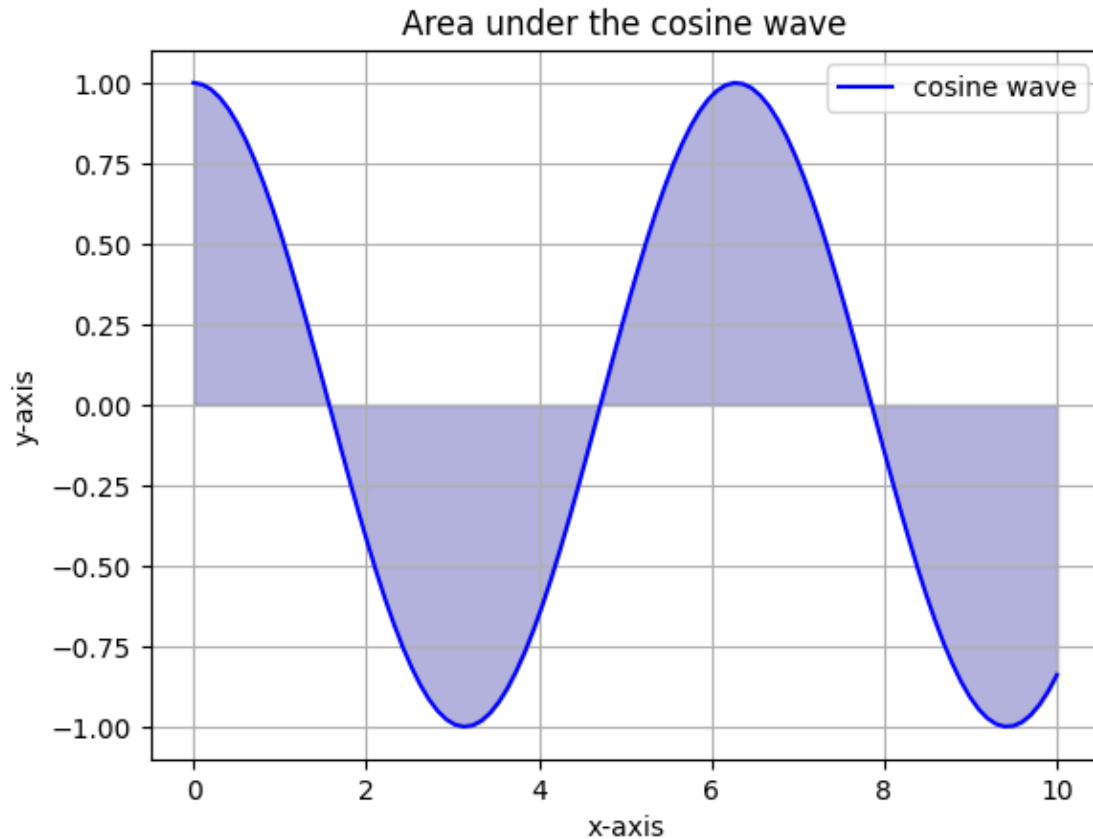
```
[ ]: x= np.linspace(0,10,100)
y = np.sin(x)
plt.plot(x,y,label = 'sine wave', color = 'blue')
plt.fill_between(x,y, color = 'skyblue', alpha = 0.3) # fill the area under
↳the curve
plt.title('Area under the sine wave')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.legend()
plt.grid(True)
plt.show()
```





$\cos(x)$

```
[ ]: x= np.linspace(0,10,100)
y = np.cos(x)
plt.plot(x,y,label = 'cosine wave', color = 'blue')
plt.fill_between(x,y, color = 'darkblue', alpha = 0.3) # fill the area under
↳the curve
plt.title('Area under the cosine wave')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.legend()
plt.grid(True)
plt.show()
```



ex- 6 subplots- multiple plots in one figure

```
[ ]: x = np.linspace(0,10,100)
y1 = np.sin(x)
y2 = np.cos(x)

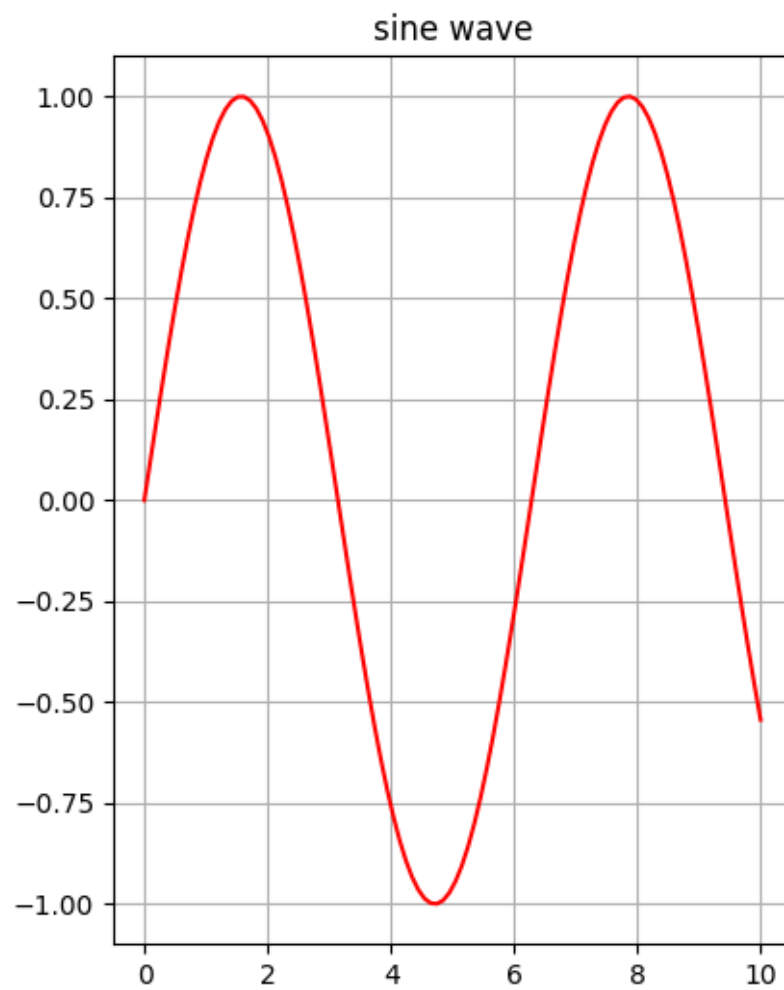
# create a subplots
plt.figure(figsize=(10,6))

plt.subplot(1,2,1) # row , column , subplot
plt.plot(x, y1, label='sine', color = 'red')
plt.title('sine wave')
plt.grid(True)

plt.figure(figsize=(10,6))

plt.subplot(1,2,1) # row , column , subplot
plt.plot(x, y1, label='cosine wave', color = 'blue')
plt.title('cosine wave')
plt.grid(True)
```

```
plt.suptitle('subplots example')  
plt.show()
```



## subplots example

