```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [1,2,3,4]
plt.plot(x, y)
plt.show()
```

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [1,2,3,4]
plt.plot(x, y)
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.show()
```

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [1,2,3,4]
plt.plot(x, y)
plt.xlabel('x – axis')
plt.ylabel('y – axis')
plt.title('My first graph')
plt.show()
```

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [1,2,3,4]
plt.plot(x, y, color='green', linestyle='dashed', linewidth = 3,  marker='o',
markerfacecolor='blue', markersize=12)
plt.xlabel('x – axis')
plt.ylabel('y – axis')
plt.title('My first graph')
plt.show()
```

```
import matplotlib.pyplot as plt
x1 = [1,2,3,4]
y1 = [1,2,3,4]
plt.plot(x1, y1)
x2 = [1,2,3,4]
y2 = [4,3,2,1]
plt.plot(x2,y2)
plt.xlabel('x-axis', fontsize=16)
plt.ylabel('y-axis', fontsize=16)
plt.title('Two lines on same graph!')
plt.show()

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

```
y1 = [1, 2, 3, 4]
y2 = [1, 2, 3, 4]
tick_label = ['one', 'two', 'three', 'four']

# Plot bars with corrected tick label usage
plt.bar(np.arange(len(y1)) - 0.2, y1, width=0.4, label='a')
plt.bar(np.arange(len(y2)) + 0.2, y2, width=0.4, label='b')

# Set tick labels
plt.xticks(np.arange(len(tick_label)), tick_label)

plt.title('My Bar Chart')
plt.legend()
plt.show()


import matplotlib.pyplot as plt
ages =[2,5,70,40,30,45,50,45,43,40,44,60,7,13,57,18,90,77,32,21,20,40]
range = (0, 100)
bins = 10
plt.hist(ages, bins, range, edgecolor='black')
plt.xlabel('age')
plt.ylabel('# of people')
plt.title('My histogram')
plt.show()

import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D  # Correct import

# Create a 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")  # Create 3D axes

x = [1, 2, 3, 4, 5]
y = [3, 5, 2, 1, 4]
z = [5, 2, 1, 4, 3]

# Plot the 3D line
ax.plot(x, y, z)

plt.show()


mport numpy as np
import matplotlib.pyplot as plt

# Given data points
x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = np.array([2.3, 3.1, 4.0, 4.8, 5.7, 6.6, 7.4, 8.2, 9.0, 9.9])

# Fit the data with polynomials of different orders
linear_fit = np.polyfit(x, y, 1)
```

```
# Print the solutions (coefficients)
print("Linear Fit (1st Order) Coefficients: ", linear_fit)

# Generate a smooth range of x values for plotting the fitted curves
x_smooth = np.linspace(min(x), max(x), 500)

# Evaluate the polynomials on the smooth x range
y1_fit = np.polyval(linear_fit, x_smooth)

# Plot the data points
plt.scatter(x, y, color='black', label='Data points')

# Plot the polynomial fits
plt.plot(x_smooth, y1_fit, label='Linear Fit (1st Order)', color='blue')

# Labels and legend
plt.xlabel('x')
plt.ylabel('y')
plt.title('Polynomial Fits of oder 1')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```