**Python Code**

import matplotlib.pyplot as plt

def linear\_interpolation(x1, y1, x2, y2, x):

# Linear interpolation formula: y = y1 + ((y2 - y1) / (x2 - x1)) \* (x - x1)

return y1 + ((y2 - y1) / (x2 - x1)) \* (x - x1)

def plot\_linear\_interpolation(x\_values, y\_values, interpolated\_points):

plt.plot(x\_values, y\_values, 'bo-', label='Original Points')

plt.plot(interpolated\_points[0], interpolated\_points[1], 'ro-', label='Interpolated Points')

plt.xlabel('X values')

plt.ylabel('Y values')

plt.title('Linear Interpolation Graph')

plt.legend()

plt.grid(True)

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

# Example data

x\_values = [1, 2, 3, 4, 5]

y\_values = [2, 4, 1, 6, 8]

# Interpolation point

x\_interpolate = 2.5

# Find the indices of the two known points for interpolation

index\_before = max(i for i, x in enumerate(x\_values) if x <= x\_interpolate)

index\_after = min(i for i, x in enumerate(x\_values) if x >= x\_interpolate)

# Perform linear interpolation

interpolated\_y = linear\_interpolation(

x\_values[index\_before], y\_values[index\_before],

x\_values[index\_after], y\_values[index\_after],

x\_interpolate

)

print(f"Linear Interpolation at x = {x\_interpolate}: {interpolated\_y}")

# Plot the graph

interpolated\_points = ([x\_interpolate], [interpolated\_y])

plot\_linear\_interpolation(x\_values, y\_values, interpolated\_points)

Output

Linear Interpolation at x = 2.5: 2.5

