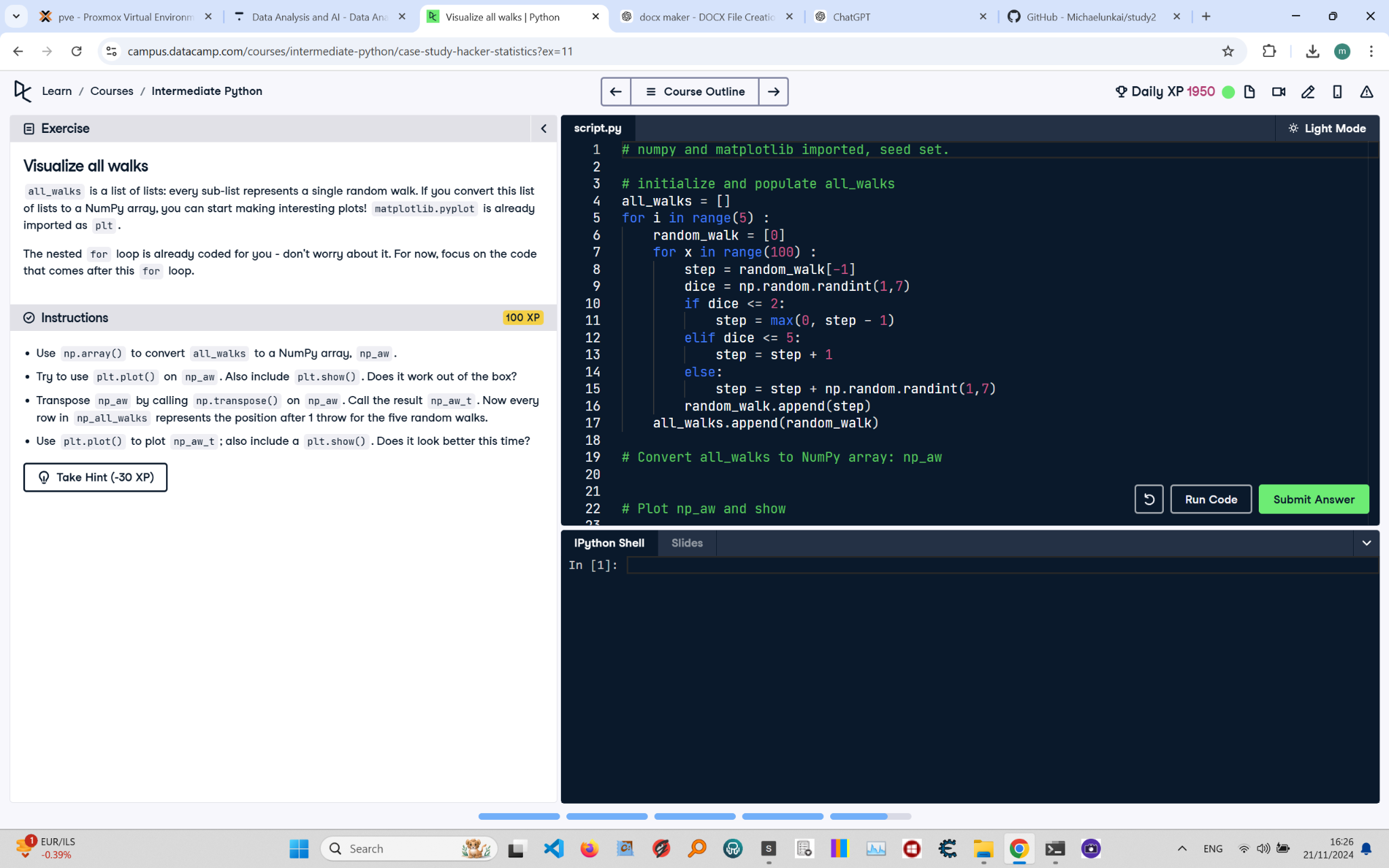
# Visualize All Walks - Final Corrected Solution



Below is the final corrected solution for the 'Visualize All Walks' exercise from the Python course. The image includes the instructions, code, and task details.

Final Corrected Solution:

# NumPy and matplotlib imported, seed set  
import numpy as np  
import matplotlib.pyplot as plt  
np.random.seed(123)  
  
# Initialize and populate all\_walks  
all\_walks = []  
for i in range(5): # Loop runs 5 times  
 random\_walk = [0]  
 for x in range(100): # Loop runs 100 times for each walk  
 # Set step: last element in random\_walk  
 step = random\_walk[-1]  
   
 # Roll the dice  
 dice = np.random.randint(1, 7)  
   
 # Determine next step using max to prevent step from going below 0  
 if dice <= 2:  
 step = max(0, step - 1) # Move down but ensure step doesn't go below 0  
 elif dice <= 5:  
 step = step + 1 # Move up  
 else:  
 step = step + np.random.randint(1, 7) # Move up by a random value  
   
 # Append next\_step to random\_walk  
 random\_walk.append(step)  
   
 # Append random\_walk to all\_walks  
 all\_walks.append(random\_walk)  
  
# Convert all\_walks to NumPy array: np\_aw  
np\_aw = np.array(all\_walks)  
  
# Plot np\_aw and show  
plt.plot(np\_aw)  
plt.show()  
  
# Clear the figure  
plt.clf()  
  
# Transpose np\_aw: np\_aw\_t  
np\_aw\_t = np.transpose(np\_aw)  
  
# Plot np\_aw\_t and show  
plt.plot(np\_aw\_t)  
plt.show()

Explanation:

1. Import numpy as np and matplotlib.pyplot as plt, and set the random seed using np.random.seed(123).

2. Initialize all\_walks as an empty list to store the results of multiple random walks.

3. Use a for loop to repeat the random walk simulation 5 times, appending each completed random\_walk to all\_walks.

4. Convert the list all\_walks into a NumPy array called np\_aw using np.array(all\_walks).

5. Plot the np\_aw array using plt.plot(np\_aw) and display the plot with plt.show().

6. Call plt.clf() to clear the figure, ensuring a clean plot for the next set of data.

7. Transpose the np\_aw array to np\_aw\_t using np.transpose(np\_aw), ensuring that each row represents a random walk.

8. Plot the transposed array np\_aw\_t using plt.plot(np\_aw\_t) and display it with plt.show(), showing each walk clearly.