**Day 4 Notebook 1**

**numpy arrays**

import numpy as np

def main():

# Step 1: Create a NumPy array from scratch

# Using np.zeros to create an array of zeros

array\_zeros = np.zeros((3, 3))

print("Array of zeros (3x3):")

print(array\_zeros)

print()

# Using np.ones to create an array of ones

array\_ones = np.ones((2, 4))

print("Array of ones (2x4):")

print(array\_ones)

print()

# Using np.arange to create an array with a range of values

array\_range = np.arange(10)

print("Array with a range of values (0 to 9):")

print(array\_range)

print()

# Step 2: Create a NumPy array from a list

my\_list = [1, 2, 3, 4, 5]

array\_from\_list = np.array(my\_list)

print("Array created from a list:")

print(array\_from\_list)

print()

# Call the main function to start the program

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

main()

**array dimensions**

import numpy as np

def main():

    # Step 1: Create a NumPy array of random floating-point numbers

    random\_array = np.random.rand(4, 4)

    # Step 2: Display the array

    print("Array:\n", random\_array)

    print()

    # Step 3: Display the array's attributes

    print("Array attributes:")

    print(f"Number of dimensions: {random\_array.ndim}")

    print(f"Shape: {random\_array.shape}")

    print(f"Size: {random\_array.size}")

    print(f"Data type: {random\_array.dtype}")

    print(f"Item size (bytes): {random\_array.itemsize}")

    print(f"Total size (bytes): {random\_array.nbytes}")

# Call the main function to start the program

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

main()

**indexing**

import numpy as np

def main():

    # Create a 3x3 NumPy array of random integers between 1 and 10

    array\_3x3 = np.random.randint(1, 11, size=(3, 3))

    # Display the original array

    print("Original array:")

    print(array\_3x3)

    print()

    # Access the element at row 1, column 2

    element\_1 = array\_3x3[1, 2]

    print("Element at row 1, column 2:")

    print(element\_1)

    print()

    # Access the entire second row

    row\_2 = array\_3x3[1]

    print("Entire second row:")

    print(row\_2)

    print()

# Top-level scope check

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

    main()

**ufuncs**

import numpy as np

def main():

    # Step 1: Create two 3x3 NumPy arrays

    array1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

    array2 = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])

    # Display the original arrays

    print("Array 1:")

    print(array1)

    print()

    print("Array 2:")

    print(array2)

    print()

    # Step 2: Perform element-wise addition

    sum\_array = np.add(array1, array2)

    print("Element-wise addition (Array 1 + Array 2):")

    print(sum\_array)

    print()

    # Step 3: Perform element-wise multiplication

    product\_array = np.multiply(array1, array2)

    print("Element-wise multiplication (Array 1 \* Array 2):")

    print(product\_array)

    print()

# Top-level scope check

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

    main()

**aggregated mean**

import numpy as np

def main():

# Create a 5x6 array of random numbers between 0 and 1

m = np.random.random((5, 6))

# Print the generated array

print("Generated Array:")

print(m)

# Calculate the overall mean of the entire array

overall\_mean = m.mean()

# Print the overall mean value

print("Overall Mean of the entire array:")

print(overall\_mean)

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

main()

**mask and count**

import numpy as np

def main():

    # Create a new random number generator with a seed of 42

    rng = np.random.RandomState(42)

    # Generate a 20x20 array of random integers between 0 and 30

    x = rng.randint(0, 31, size=(20, 20))

    # Print the generated array

    print("Generated Array:")

    print(x, "\n")

    # Count the number of values that are greater   
 # than 5, less than 15, and even

    condition = (x > 5) & (x < 15) & (x % 2 == 0)

    count\_condition = np.count\_nonzero(condition)

    # Print the count of values that meet the condition

    print("Number of target values:")

    print(count\_condition)

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

    main()

**Day 4 Notebook 2**

**pandas series**

import pandas as pd

def main():

    # Data for the Series

    data = [85, 90, 78]

    index = ['Alice', 'Bob', 'Charlie']

    # Create the Series

    scores\_series = pd.Series(data, index=index)

    # Print the Series

    print(scores\_series)

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

    main()

**pandas dataframe**

import pandas as pd

def main():

    # Data for the DataFrame

    data = {

        'Name': ['Alice', 'Bob', 'Charlie'],

        'Math': [85, 90, 78],

        'Science': [92, 88, 95],

        'English': [87, 85, 90]

    }

    # Create the DataFrame

    df = pd.DataFrame(data)

    # Print the DataFrame

    print(df)

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

    main()

**Day 4 Notebook 3**

**line plot**

import matplotlib as mpl

import matplotlib.pyplot as plt

plt.style.use('classic')

print("matplotlib version =", mpl.\_\_version\_\_)

# Define the data

x = range(21)

# Create a figure and axes

fig, ax = plt.subplots()

# Plot the data with the required modifications

ax.plot(x, [xi\*\*2 for xi in x], color='red', linestyle='dotted')  # Modified color, linestyle, and function

# Set limits and labels with the required modifications

ax.set(xlim=(0, 20),

       ylabel='Output',  # Modified y-axis label

       xlabel='Input',   # Modified x-axis label

       title='Modified Plot of y = x^2')  # Modified title

# Show the plot

plt.show()