**LAB 0: LIST , SET ,DICTIONARY, TUPLE**

# Creating a list

fruits = ["Apple", "Banana", "Mango"]

# Adding an element

fruits.append("Orange")

# Removing an element

fruits.remove("Banana")

# Accessing elements

print(fruits[0]) # Output: Apple

# Iterating over the list

for fruit in fruits:

print(fruit)

**#**Creating a set

numbers = {1, 2, 3, 4, 4, 5} # Duplicate '4' will be removed automatically

# Adding elements

numbers.add(6)

# Removing elements

numbers.discard(2)

# Checking membership

print(3 in numbers) # Output: True

# Iterating over the set

for num in numbers:

print(num)

# Creating a dictionary

student = {"name": "Rahul", "age": 21, "course": "Python"}

# Adding a new key-value pair

student["grade"] = "A"

# Removing a key

student.pop("age")

# Accessing values

print(student["name"]) # Output: Rahul

# Iterating over dictionary

for key, value in student.items():

print(f"{key}: {value}")

# Creating a tuple

coordinates = (10, 20, 30)

# Accessing elements

print(coordinates[1]) # Output: 20

# Tuples are immutable, so the following will cause an error:

# coordinates[1] = 40 # TypeError: 'tuple' object does not support item assignment

# Iterating over a tuple

for value in coordinates:

print(value)

**LAB 1: numpy,pandas,scipy**

**exnumpy.py**

import numpy as np

# Creating an array

arr = np.array([1, 2, 3, 4, 5])

print("Array:", arr)

# Basic Operations

print("Mean:", np.mean(arr)) # Calculate mean

print("Sum:", np.sum(arr)) # Calculate sum

# Reshaping an array

matrix = arr.reshape(1, 5)

print("Reshaped Array:", matrix)

**expandas.py**

import pandas as pd

# Creating a DataFrame

data = {"Name": ["Amit", "Sara", "Raj"], "Age": [25, 30, 22]}

df = pd.DataFrame(data)

print(df)

# Accessing a column

print("Ages:", df["Age"])

# Filtering Data

filtered\_df = df[df["Age"] > 25]

print("Filtered Data:", filtered\_df)

**exscipy.py**

from scipy import stats

import numpy as np

# Creating sample data

data = np.array([2, 3, 5, 6, 9])

# Calculating statistics

mean\_value = stats.tmean(data)

median\_value = np.median(data)

print("Mean:", mean\_value)

print("Median:", median\_value)

MATPLOTLIB:

### **1. Line Plot**

import matplotlib.pyplot as plt

# Data

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

y = [10, 20, 15, 25, 30]

# Plot

plt.plot(x, y, marker='o', linestyle='-', color='blue')

# Labels & Title

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.title("Simple Line Plot")

# Show Plot

plt.show()

✅ **Creates a basic line graph with markers.**

### **2. Scatter Plot**

import matplotlib.pyplot as plt

# Data

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

y = [10, 20, 15, 25, 30]

# Scatter Plot

plt.scatter(x, y, color='red', marker='s')

# Labels & Title

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.title("Scatter Plot")

# Show Plot

plt.show()

✅ **Creates a scatter plot where each point is separate.**

### **3. Bar Chart**

import matplotlib.pyplot as plt

# Data

categories = ['A', 'B', 'C', 'D', 'E']

values = [10, 25, 15, 30, 20]

# Bar Plot

plt.bar(categories, values, color='green')

# Labels & Title

plt.xlabel("Categories")

plt.ylabel("Values")

plt.title("Bar Chart Example")

# Show Plot

plt.show()

✅ **Creates a bar chart with categories.**

### **4. Histogram**

import matplotlib.pyplot as plt

import numpy as np

# Generate Random Data

data = np.random.randn(1000) # 1000 random numbers

# Histogram

plt.hist(data, bins=20, color='purple', edgecolor='black')

# Labels & Title

plt.xlabel("Value Range")

plt.ylabel("Frequency")

plt.title("Histogram Example")

# Show Plot

plt.show()

✅ **Creates a histogram to show data distribution.**

### **5. Pie Chart**

import matplotlib.pyplot as plt

# Data

labels = ['Apple', 'Banana', 'Orange', 'Grapes']

sizes = [30, 25, 20, 25]

colors = ['red', 'yellow', 'orange', 'purple']

# Pie Chart

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)

# Title

plt.title("Fruit Distribution")

# Show Plot

plt.show()

## **Lab Exercise: Visualizing Data with Matplotlib**

Write a Python program using Matplotlib to:

1. Create a **line plot** for given data.
2. Create a **scatter plot** for the same data.
3. Customize the plots with labels, titles, and legends.
4. Display both plots in a **subplot layout**.

### **Dataset (Sample Data):**

* **Years:** [2015, 2016, 2017, 2018, 2019, 2020, 2021]
* **Sales (in ₹ Lakhs):** [25, 40, 35, 50, 65, 80, 95]

import matplotlib.pyplot as plt

# Data

years = [2015, 2016, 2017, 2018, 2019, 2020, 2021]

sales = [25, 40, 35, 50, 65, 80, 95]

# Create figure and subplots

fig, axs = plt.subplots(1, 2, figsize=(10, 4))

# Line Plot

axs[0].plot(years, sales, marker='o', color='blue', linestyle='--')

axs[0].set\_title("Annual Sales Growth")

axs[0].set\_xlabel("Years")

axs[0].set\_ylabel("Sales (₹ Lakhs)")

axs[0].grid(True)

# Scatter Plot

axs[1].scatter(years, sales, color='red', marker='s')

axs[1].set\_title("Sales Distribution")

axs[1].set\_xlabel("Years")

axs[1].set\_ylabel("Sales (₹ Lakhs)")

axs[1].grid(True)

# Adjust layout and display

plt.tight\_layout()

plt.show()