Name: Atharva Kesarkar
Class: AI&DS Batch -1
Roll no: 17

Experiment 2

AIM: Experiment to demonstrate basics of python data structure (string, list).

Tools: Anaconda Navigator (Jupyter notebook)

◆ Objective

The purpose of this experiment is to:

- Understand how **strings** and **lists** are used to handle data in Python.
- Practice string operations such as **joining**, **indexing**, **slicing**, and using built-in **string methods**.
- Perform common **list operations** like adding, deleting, sorting, and counting elements.
- Learn the difference between **mutable** (list) and **immutable** (string) data types.
- Build readiness for working with collections and manipulating data in upcoming Python programs.

📓 Theory

◆ Strings in Python

A **string** is a series of characters inside quotes (' ', " ", or ''' '''). Strings are **immutable**, which means once created, they cannot be modified directly.

☞ Concatenation (Joining Strings)

Using +

```
first = "Atharva"
last = "Kesarkar"
full = first + " " + last
print(full)
```

Using ,

```
print(first, last)
```

Using join()

```
full = " ".join([first, last])
```

Using f-string

```
age = 18
greeting = f"My name is {first} and I am {age} years old."
print(greeting)
```

☞ Slicing and Methods

```
text = "Python"
print(text[0:3])        # Output: Pyt
print(text.upper())     # Output: PYTHON
print(text.replace("Py", "My"))  # Output: Mython
```

◆ Lists in Python

A **list** is an ordered collection that can store different data types. Lists are **mutable**, so their contents can be changed.

☞ Common List Functions

```
fruits = ["Apple", "Banana", "Cherry"]

fruits.append("Mango")       # Adds Mango
fruits.insert(1, "Orange")   # Inserts at index 1
fruits.remove("Banana")      # Removes Banana
fruits.pop()                 # Removes last item
fruits.sort()                # Sorts list
fruits.reverse()             # Reverses list
```

☞ Other Useful Functions

```
print(len(fruits))
print(fruits.count("Apple"))
print(min([2, 4, 1]))
print(sum([10, 20, 30]))
```

Code:

1. STRING MANIPULATION

```
# === STRING MANIPULATION ===
print("----- String Manipulation -----")
var1 = "Atharva"
var2 = " Kesarkar"
    1. Using + operator
combined = var1 + " " + var2
print("Using + :", combined)
    2. Using join() method
joined = " ".join([var1, var2])
print("Using join():", joined)
    3. Using comma
print("Using comma:", var1, var2)
    4. Using f-string
name = "Atharva Kesarkar"
age = 18
print(f"My name is {name} and I am {age} years old.")
```

Output:

```
# === STRING MANIPULATION ===
print("----- String Manipulation -----")
var1 = "Atharva"
var2 = " Kesarkar"
# 1.    Using + operator
combined = var1 + " " + var2
print("Using + :", combined)
# 2.    Using join() method
joined = " ".join([var1, var2])
print("Using join():", joined)
# 3.    Using comma
print("Using comma:", var1, var2)
# 4.    Using f-string
name = "Atharva Kesarkar"
age = 18
print(f"My name is {name} and I am {age} years old.")
```

```
----- String Manipulation -----
Using + : Atharva  Kesarkar
Using join(): Atharva  Kesarkar
Using comma: Atharva  Kesarkar
My name is Atharva Kesarkar and I am 18 years old.
```

2. LIST MANIPULATION

```
# === LIST MANIPULATION ===
print("\n----- List Manipulation -----")
my_list = [100, 200, 300, 400]
print("Original List:", my_list)
my_list.append(500)
print("After append:", my_list)
my_list.insert(3, 350)
print("After insert at index 3:", my_list)
my_list.remove(200)
print("After removing 200:", my_list)
print("Element at index 2:", my_list[2])
print("Sliced List [2:5]:", my_list[2:5])
print("Length of List:", len(my_list))
print("Max element:", max(my_list))
print("Min element:", min(my_list))
print("Sum of elements:", sum(my_list))
my_list.sort()
print("Sorted List:", my_list)
my_list.reverse()
print("Reversed List:", my_list)
```

Output:

```python
print("\n----- List Manipulation -----")
my_list = [100, 200, 300, 400]
print("Original List:", my_list)
my_list.append(500)
print("After append:", my_list)
my_list.insert(3, 350)
print("After insert at index 3:", my_list)
my_list.remove(200)
print("After removing 200:", my_list)
print("Element at index 2:", my_list[2])
print("Sliced List [2:5]:", my_list[2:5])
print("Length of List:", len(my_list))
print("Max element:", max(my_list))
print("Min element:", min(my_list))
print("Sum of elements:", sum(my_list))
my_list.sort()
print("Sorted List:", my_list)
my_list.reverse()
print("Reversed List:", my_list)
```

```
----- List Manipulation -----
Original List: [100, 200, 300, 400]
After append: [100, 200, 300, 400, 500]
After insert at index 3: [100, 200, 300, 350, 400, 500]
After removing 200: [100, 300, 350, 400, 500]
Element at index 2: 350
Sliced List [2:5]: [350, 400, 500]
Length of List: 5
Max element: 500
Min element: 100
Sum of elements: 1650
Sorted List: [100, 300, 350, 400, 500]
Reversed List: [500, 400, 350, 300, 100]
```

Conclusion:

This experiment taught me how strings and lists are used in Python to handle and organize data. I learned that strings are immutable, so we use slicing and methods to work with them, while lists are mutable and can be directly modified using several built-in functions. These operations are essential for processing and manipulating data efficiently in any Python application.

For Faculty use only:

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of practical [40%]**	**Attendance/ Learning Attitude [20%]**	
Marks Obtained				