**EX.NO: 1 DATE:**

**Perform Creation, indexing, slicing, concatenation and repetition**

**operations on python built-in data types: Strings, List, Tuples, Dictionary, Set.**

**AIM:**

To perform creation, indexing, slicing, concatenation, and repetition operations on Python built-in data types: Strings, List, Tuples, Dictionary, and Set.

# PROCEDURE:

## Creation of Data Types:

* **String:** Define a string using quotes (single or double).
* **List:** Create a list using square brackets [].
* **Tuple:** Create a tuple using parentheses ().
* **Dictionary:** Create a dictionary using curly braces {} with key-value pairs.
* **Set:** Create a set using curly braces {}, containing unique elements.

## Indexing and Slicing:

* **Indexing:** Retrieve elements using their index, starting from 0.
* **Slicing:** Access a range of elements using the slicing operator [start:end].

## Concatenation:

* **String:** Concatenate two strings using the + operator.
* **List and Tuple:** Use the + operator to concatenate lists and tuples.
* **Dictionary:** Merge two dictionaries using the update() method or | operator (Python 3.9+).
* **Set:** Use the union() method or | operator to concatenate sets.

## Repetition:

* String, List, Tuple: Repeat elements using the \* operator.
* Set and Dictionary: Repetition is not supported directly but can be demonstrated through list comprehensions or loops for dictionary and set creation.

# PROGRAM:

## 1. Creation of Data Types String

str1 = "Hello" str2 = "World"  **List**

list1 = [1, 2, 3, 4]

list2 = ['a', 'b', 'c']

## Tuple

tuple1 = (10, 20, 30)

tuple2 = ('x', 'y', 'z')

## Dictionary

dict1 = {1: 'one', 2: 'two'}

dict2 = {3: 'three', 4: 'four'}

## Set

set1 = {1, 2, 3}

set2 = {3, 4, 5}

## 2. Indexing and Slicing String

print("Indexing in String:", str1[1]) print("Slicing in String:", str1[1:4])  **List**

print("Indexing in List:", list1[2]) print("Slicing in List:", list1[1:3])  **Tuple**

print("Indexing in Tuple:", tuple1[0]) print("Slicing in Tuple:", tuple1[1:])  **Dictionary (indexing by key)**

print("Accessing Dictionary by Key:", dict1[1])  **Set (No indexing/slicing, but can be looped)** for item in set1:

print("Item in Set:", item)

## 3. Concatenation String

concat\_str = str1 + " " + str2 print("Concatenated String:", concat\_str)  **List**

concat\_list = list1 + list2 print("Concatenated List:", concat\_list)  **Tuple**

concat\_tuple = tuple1 + tuple2 print("Concatenated Tuple:", concat\_tuple)  **Dictionary**

dict1.update(dict2) or dict3 = dict1 | dict2 (Python 3.9+) print("Merged Dictionary:", dict1)

## Set

concat\_set = set1 | set2 print("Union of Sets:", concat\_set)

## 4. Repetition String

repeat\_str = str1 \* 3 print("Repeated String:", repeat\_str)  **List**

repeat\_list = list1 \* 2 print("Repeated List:", repeat\_list)  **Tuple**

repeat\_tuple = tuple1 \* 2 print("Repeated Tuple:", repeat\_tuple)

**Set and Dictionary: Repetition not directly supported**

# OUTPUT:

## Indexing in String: e Slicing in String: ell Indexing in List: 3 Slicing in List: [2, 3]

**Indexing in Tuple:** 10

**Slicing in Tuple:** (20, 30)

## Accessing Dictionary by Key: one

**Item in Set:** 1 **Item in Set:** 2 **Item in Set:** 3

**Concatenated String:** Hello World

**Concatenated List:** [1, 2, 3, 4, 'a', 'b', 'c']

**Concatenated Tuple:** (10, 20, 30, 'x', 'y', 'z')

**Merged Dictionary:** {1: 'one', 2: 'two', 3: 'three', 4: 'four'}

**Union of Sets:** {1, 2, 3, 4, 5}

**Repeated String:** HelloHelloHello

**Repeated List:** [1, 2, 3, 4, 1, 2, 3, 4]

**Repeated Tuple:** (10, 20, 30, 10, 20, 30)

# RESULT:

The creation, indexing, slicing, concatenation, and repetition operations were successfully performed on Python built-in data types: String, List, Tuple, Dictionary, and Set