**Bangladesh-Bharot Digital Service for Employment & Training (BDSET)**

**Module: Python Programming**

**Lab 04  
Problem Solving using Basics of Python for Artificial Intelligence**

**Advanced data storage technique in python**

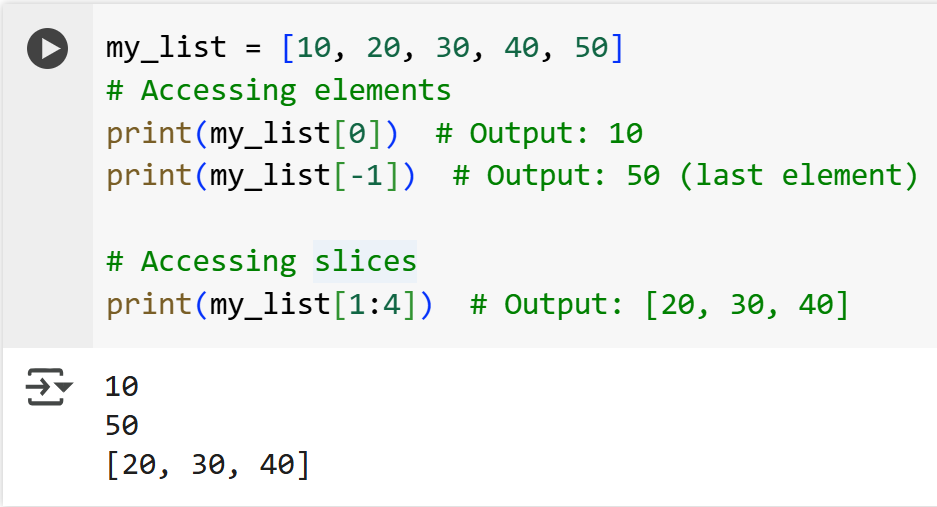
**Objective:**

Learn advanced data storage techniques in Python, focusing on indexing, manipulation, and performing basic operations with lists and dictionaries.

### ****1. Indexing in Lists and Dictionaries****

#### **Lists**

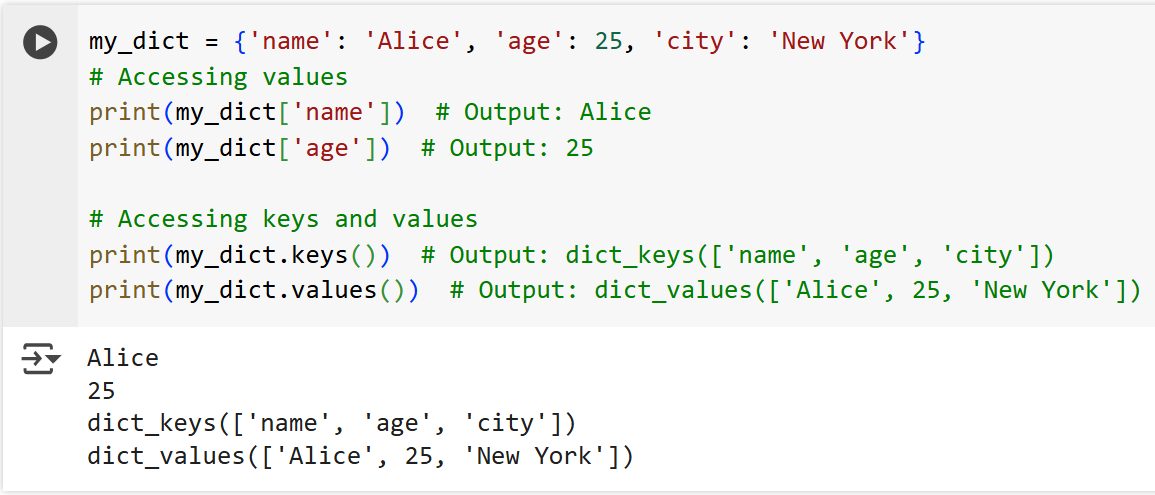
* Lists are ordered collections of items. Each element in a list is indexed, starting at 0.
* Syntax for accessing an element: list[index] where index contain 0, 1, 2, 3…

**Example: **

#### **Dictionary**

* Dictionary stores data as key-value pairs. Access elements using their keys.
* Syntax for accessing a value: dict[key] where key is user defined

**Example**

****

### ****2. Create, Update, and Delete List and Dictionary Elements****

#### **Lists**

**Create a List:**

my\_list = [] # Empty list

my\_list = [1, 2, 3] # List with elements

**Update Elements:**

my\_list[1] = 20

print(my\_list) # Output: [1, 20, 3]

**Add Elements:**

my\_list.append(4) # Add at the end

my\_list.insert(1, 10) # Add at index 1

print(my\_list) # Output: [1, 10, 20, 3, 4]

**Delete Elements:**

del my\_list[2] # Delete by index

my\_list.remove(10) # Remove by value

print(my\_list) # Output: [1, 20, 3, 4]

#### **Dictionary**

**Create a Dictionary:**

my\_dict = {} # Empty dictionary

my\_dict = {'name': 'Alice', 'age': 25}

**Update Elements:**

my\_dict['age'] = 26

my\_dict['city'] = 'Los Angeles' # Add new key-value pair

print(my\_dict) # Output: {'name': 'Alice', 'age': 26, 'city': 'Los Angeles'}

**Delete Elements:**

del my\_dict['age'] # Delete by key

print(my\_dict) # Output: {'name': 'Alice', 'city': 'Los Angeles'}

my\_dict.pop('city') # Remove and return value

print(my\_dict) # Output: {'name': 'Alice'}

### ****3. Perform Basic Operations on List and Dictionary Elements****

#### **Lists**

**Concatenation:**

list1 = [1, 2, 3]

list2 = [4, 5, 6]

combinedList = list1 + list2

print(combinedList) # Output: [1, 2, 3, 4, 5, 6]

**Repetition:**

my\_list = [1, 2]

repeated = my\_list \* 3

print(repeated) # Output: [1, 2, 1, 2, 1, 2]

**Membership Testing:**

my\_list = [10, 20, 30]

print(20 in my\_list) # Output: True

print(40 in my\_list) # Output: False

**Sorting:**

my\_list = [50, 10, 30, 20, 40]

my\_list.sort()

print(my\_list) # Output: [10, 20, 30, 40, 50]

Moreover, there are many methods for manipulation List methods like:

clear()

count()

extend()

index()

reverse()

copy()

len()

min()

max()

del() etc…..

**Dictionary**

**Merge Dictionaries**

dict1 = {'a': 1, 'b': 2}

dict2 = {'b': 3, 'c': 4}

dict1.update(dict2)

print(dict1) # Output: {'a': 1, 'b': 3, 'c': 4}

**Iterate Over Keys and Values**

my\_dict = {'name': 'Alice', 'age': 25}

for key, value in my\_dict.items():

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

# Output:

# name: Alice

# age: 25

**Membership Checking**

my\_dict = {'name': 'Alice', 'age': 25}

print('name' in my\_dict) # Output: True

print('city' in my\_dict) # Output: False

**Exercise:**

1. Create a list of 5 integers. Replace the 3rd element with a new value, add two more elements, and remove the last element.
2. Create a dictionary to store information about a book (title, author, price). Update the price, add a publication year, and then delete the author.
3. Write a Python program to merge two dictionaries and handle any duplicate keys.
4. Write a Python program to sort a list of numbers and remove duplicates.