**ASSIGNMENT1**

**Accessing List**

**Understanding how to create and access elements in a list.**

**1 Creating a list:**

You can create a list by placing elements inside square brackets[],separated by commas

Example:

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

fruits=[“apple”,”banana”,”cherry”]

mixed=[42,”Hello”,3.14,True]

empty\_list=[]

**2. Accessing Elements in a List:**

Elements in a list are accessed by their **index**. Python uses **zero-based indexing**.

* The first element has an index of 0.
* The last element can be accessed using -1 (negative indexing).

Example:

my\_list = ["Python", "Java", "C++", "JavaScript"]

print(my\_list[0]) #Output:Python

print(my\_list[1]) # Output: Java

print(my\_list[-1]) # Output: JavaScript

**3. Modifying List Elements:**

Lists are mutable, meaning you can change elements after creation.

Example:

my\_list[1] = "C#"

print(my\_list)

Output:

# Output: ['Python', 'C#', 'C++', 'JavaScript']

### 4. ****Adding Elements to a List:****

* append() – Adds an element to the end of the list.
* insert() – Inserts an element at a specific position.

Example:

my\_list.append("Ruby")

print(my\_list)

my\_list.insert(2, "Go")

print(my\_list)

### 5. ****Removing Elements:****

* remove() – Removes the first occurrence of a value.
* pop() – Removes the element at a specific index (or the last element by default).

Example:

my\_list.remove("C++")

print(my\_list)

my\_list.pop(1)

print(my\_list)

### 6. ****Looping Through a List:****

### ****Example:****

for item in my\_list:

print(item)

**Indexing in lists (positive and negative indexing).**

Indexing allows you to access individual elements in a list by their position. Python supports **positive indexing** (left to right) and **negative indexing** (right to left).

### ****1. Positive Indexing:****

* Starts from 0 (the first element).
* Increases by 1 for each subsequent element.

**Example:**

fruits = ["apple", "banana", "cherry", "date"]

print(fruits[0])

print(fruits[1])

print(fruits[3])

### ****2. Negative Indexing:****

* Starts from -1 (the last element).
* Moves left as the index decreases.

Example:

fruits = ["apple", "banana", "cherry", "date"]

print(fruits[-1])

print(fruits[-2])

print(fruits[-4])

**Practical Example:**

colors = ["red", "green", "blue", "yellow

print("First color (positive):", colors[0])

print("Third color (positive):", colors[2])

print("Last color (negative):", colors[-1])

print("Second last color (negative):", colors[-2])

Output:

**Output:**

First color (positive): red

Third color (positive): blue

Last color (negative): yellow

Second last color (negative): blue

### ****Slicing a List: Accessing a Range of Elements****

Slicing allows you to extract a portion of a list by specifying a start, stop, and optional step value. The result is a new list containing the selected elements.

### ****Basic Syntax:****

sliced\_list = my\_list[start : stop : step]

* **start** – The index where the slice begins (inclusive).
* **stop** – The index where the slice ends (exclusive).
* **step** – (Optional) The interval between elements.

### ****Examples:****

numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

#### ****Basic Slicing (Start and Stop):****

Example:

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

* Starts at index 2 (value 2).
* Stops at index 6 (value 6, but not included).

#### ****Slicing with Default Values:****

Example:

print(numbers[:5]) # Output: [0, 1, 2, 3, 4] # Start from the beginning

print(numbers[5:]) # Output: [5, 6, 7, 8, 9] # Go till the end

print(numbers[:]) # Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] # Copy entire list

* **[:5]** – Start from the beginning, stop at index 5.
* **[5:]** – Start at index 5, go till the end.
* **[:]** – Copy the entire list.

#### ****Slicing with Negative Indexing:****

Example:

print(numbers[-5:-2]) # Output: [5, 6, 7]

print(numbers[:-3]) # Output: [0, 1, 2, 3, 4, 5, 6]

print(numbers[-3:]) # Output: [7, 8, 9]

* **-5:-2** – Slices from the 5th-last to the 3rd-last element.
* **[:-3]** – All except the last 3 elements.
* **[-3:]** – Last 3 elements.

#### ****Slicing with Step:****

Example:

print(numbers[1:8:2]) # Output: [1, 3, 5, 7]

print(numbers[::2]) # Output: [0, 2, 4, 6, 8] # Every 2nd element

print(numbers[::-1]) # Output: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0] # Reversed list

* **[1:8:2]** – Start at index 1, stop at 8, and take every second element.
* **[::-1]** – Reverses the list by using a step of -1.

### ****Practical Example:****

names = ["Alice", "Bob", "Charlie", "David", "Eva"]

# First three names

print(names[:3]) # Output: ['Alice', 'Bob', 'Charlie']

# Last two names

print(names[-2:]) # Output: ['David', 'Eva']

# Skip every other name

print(names[::2]) # Output: ['Alice', 'Charlie', 'Eva']

# Reverse the list

print(names[::-1]) # Output: ['Eva', 'David', 'Charlie', 'Bob', 'Alice']

Slicing provides powerful and flexible ways to manipulate lists, making it an essential tool in Python programming.