Experiment 7

AIM

Write a Python program to find the index of an item in a specified list.

Description

List:

A list in Python is used to store the sequence of various types of data. Python lists are mutable type its mean we can modify its element after it created. A list can be defined as a collection of values or items of different types. The items in the list are separated with the comma (,) and enclosed with the square brackets [].

Characteristics of Lists

The list has the following characteristics:

1. The lists are ordered
2. The element of the list can be accessed by index
3. The lists are the mutable type
4. A list can store elements of different data types
5. A list can contain duplicate items

Creating a List of numbers

List = [10, 20, 14]

accessing a element from the

List using index number

print("Accessing a element from the list")

print(List[0])

print(List[2])

Output:

10

14

List Comprehension

List comprehensions are used for creating new lists from other iterables like tuples, strings, arrays, lists, etc. A list comprehension consists of brackets containing the expression, which is executed for each element along with the for loop to iterate over each element.

Syntax

newList = [expression(element) for element in oldList if condition]

Using list comprehension to iterate through loop

List = [character for character in 'Hello World!']

enumerate():

Enumerate() method adds a counter to an iterable and returns it in a form of enumerating object. This enumerated object can then be used directly for loops or converted into a list of tuples using the list() method.

enumerate(iterable, start=0)

Parameters:

Iterable: any object that supports iteration

Start: the index value from which the counter is to be started, by default it is 0

Example:

mylist = ['A', 'B' ,'C', 'D']

e_list = enumerate(mylist)

print(list(e_list))

Output:

[(0, 'A'), (1, 'B'), (2, 'C'), (3, 'D')]

index():

The index() method returns the position at the first occurrence of the specified value.

Example:

mylist = ["Hello", 1,1.1,12,14,15,16,100,-90,80]

print(f"Index position of -90 in mylist: {mylist.index(-90)}")

Output:

Index position of -90 in mylist: 8

Program

```
# Consider this string
str1 = 'Today is a Good day!'
print(str1,type(str1))
# Form a list from the string
print("String converted to List: ")
list1 = [x for x in str1]
print(list1)

# Find index of 'o' in the list
print("Index of 'o' in the list: ")
print(list1.index('o'))
```

```
Today is a Good day! <class 'str'>
String converted to List:
['T', 'o', 'd', 'a', 'y', ' ', 'i', 's', ' ', 'a', ' ', 'G', 'o', 'o', 'd', ' ', 'd',
Index of 'o' in the list:
1
```

```
## Use list comprehension to form
## a list with index of all occurances of 'o' in the list
print('Method 1:')
list2 = [x for x in range(len(list1)) if list1[x]=='o']
print(f"List with indices of 'o': {list2}")
print('\n')
##Method2
print('Method 2:')
list3 = [x[0] for x in enumerate(list1) if x[1]=='o']
print(f"List with indices of 'o': {list3}")
```

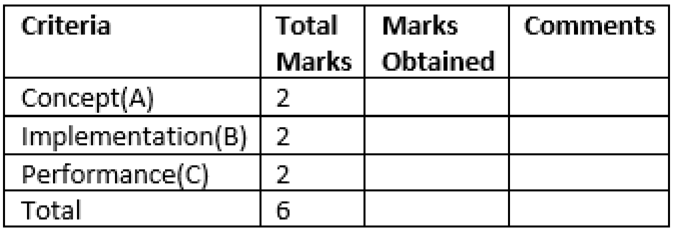
```
Method 1:
List with indices of 'o': [1, 12, 13]


Method 2:
List with indices of 'o': [1, 12, 13]
```

Conclusion

Hence, we successfully found the index of the specified item from the specified list.

Double-click (or enter) to edit

Evaluation



Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		