



# **High Impact Skills Development Program**

## **AI & Data Science**

### **Lab 05: String and List Methods**



## String Methods:

upper()

lower()

find()

index()

count()

join(lis1)

isnum()

isalnum()

split()

## List Methods:

- |           |  |
|-----------|--|
| append()  | Adds an element at the end of the list                                       |
| clear()   | Removes all the elements from the list                                       |
| copy()    | Returns a copy of the list   |
| count()   | Returns the number of elements with the specified value                      |
| extend()  | Add the elements of a list (or any iterable), to the end of the current list |
| index()   | Returns the index of the first element with the specified value              |
| insert()  | Adds an element at the specified position                                    |
| pop()     | Removes the element at the specified position                                |
| remove()  | Removes the first item with the specified value                              |
| reverse() | Reverses the order of the list   |
| sort()    | Sorts the list   |

## Built-in Operators:



- 'is': is operator is used to check if two items are sharing same memory location or not. If they are sharing same then it will be true and that was in the case of x and z. z was referencing to x location.

```
x = ["apple", "banana"]
```

```
y = ["apple", "banana"]
```

```
z = x
```

```
print(x is z)
```

```
# returns True because z is the same object as x
```

```
print(x is y)
```

```
# returns False because x is not the same object as y, even if they have the same content
```

```
print(x == y)
```

```
#to demonstrate the difference between "is" and "==": this comparison returns True because x is equal to y
```

- 'in': returns True because a sequence with the value "banana" is in the list

```
x = ["apple", "banana"]
```

```
print("banana" in x)
```

## Lab Tasks

1. Write a program that inputs a line of text. Output the text in uppercase letters and in lowercase letters using built-in functions.

```
Enter a line of text:
A line with UPPER- and lowercase LeTters

The line in uppercase is:
A LINE WITH UPPER- AND LOWERCASE LETTERS

The line in lowercase is:
a line with upper- and lowercase letters
```



2. A simple way to estimate the number of words in a string is to count the number of spaces in the string. Write a program that asks the user for a string and returns an estimate of how many words are in the string.
3. Write a program that asks the user to enter a string *s* and then converts *s* to lowercase, removes all the periods and commas from *s*, and prints the resulting string.(Hint: use the split method)

```
Enter two strings: Greg Dave
"Greg" is greater than "Dave"
```

```
Enter two strings: Bill Bill
"Bill" is equal to "Bill"
```

```
Enter two strings: Pete Tim
"Pete" is less than "Tim"
```

4. A variable `ids` refers to the list `[4353, 2314, 2956, 3382, 9362, 3900]`. Using list methods, do the following:
  - a. Remove 3382 from the list.
  - b. Get the index of 9362.
  - c. Insert 4499 in the list after 9362.
  - d. Extend the list by adding `[5566, 1830]` to it.
  - e. Reverse the list.
  - f. Sort the list.
5. When playing games where you have to roll two dice, it is nice to know the odds of each roll. For instance, the odds of rolling a 12 are about 3%, and the odds of rolling a 7 are about 17%. You can compute these mathematically, but if you don't know the math, you can write a program to do it. To do this, your program should simulate rolling two dice about 10,000 times and compute and print out the percentage of rolls that come out to be 2, 3, 4, . . . , 12. Print the results in a table format.
  - a.
6. Write a function each for *mean*, *median* and *mode* that accepts a list as an argument. The mean, median, and mode functions return an integer.



- a. Using these functions, write a program that generates a 99-element list of random numbers between 1 and 100 (inclusive) and computes its mean, median, mode.
7. Write a program that rotates the elements of a list so that the element at the first index moves to the second index, the element in the second index moves to the third index, etc., and the element in the last index moves to the first index.
8. Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.
9. Read the size of the matrix from the user. Populate it with data from the user. Print the original ,transpose and product of the matrices on the screen.