String Manipulation

Based on CBSE Curriculum

Class -11

Chapter-5

Introduction

- As we know that a sequence of characters enclosed in single quotes, double quotes or triple quotes (‘ ‘ , “ “, ‘’’ ‘’’) is called a string.
- In python, strings are immutable meaning they can’t be changed.
- In a String, each character remains at a unique position number or index number which goes from 0 to n-1 (n is the total number of characters in the string).
- In this chapter, we will see techniques of string manipulation.

String Creation

- String can be created in following ways-
  1. By assigning value directly to the variable

```
>>> str="I love my india"
>>> str
'I love my india'
```

String Literal

  2. By taking Input

```
>>> str1=input("Enter a string")
Enter a stringThis is python
>>> str1
'This is python'
```

Input ( ) always return input in string form.

Traversal of a string

- Process to access each and every character of a string for the purpose of display or for some other purpose is called string traversal.

```
name="superb"
for ch in name:
    print(ch,"-",end="")
```

Output

```
s -u -p -e -r -b -
```

Program to print a String after reverse -

```
str=input("Enter a String")
print("The string ", str," in reverse order is: ")
length=len(str)
for a in range(-1,(-length-1),-1):
    print(str[a],end="")
```

Output

```
Enter a Stringsanjeev
The string  sanjeev  in reverse order is:
veejnas
```

String Operators

- There are 2 operators that can be used to work upon strings + and *.

» **+** (it is used to join two strings)

- Like - “tea” + “pot” will result into “teapot”
- Like- “1” + “2” will result into “12”
- Like – “123” + “abc” will result into “123abc”

» ***** (it is used to replicate the string)

- like - 5*”@” will result into “@@@@@”
- Like - “go!” * 3 will result “go!go!go!”

note : - “5” * “6” expression is invalid.

Membership Operators in Strings

- 2 membership operators works with strings are in and not in. To understand the working of these operators, look at the given examples -
- in operator results into True or False. Like-
  - “a” in “Sanjeev” will result into True.
  - “ap” in “Sanjeev” will result into False.
  - “anj” in “Sanjeev” will result into True.
- not in operator also results into True or False. Like-
  - “k” not in “Sanjeev” will result into True.
  - “ap” not in “Sanjeev” will result into True.
  - “anj” not in “Sanjeev” will result into False.

String Comparison Operators

- Look carefully at following examples -
  - “a” == “a” True
  - “abc”==“abc” True
  - “a”!=“abc” True
  - “A”==“a” False
  - “abc” ==“Abc” False
  - ‘a’<‘A’ False (because Unicode value of lower case is higher than upper case)

How to get Ordinal/Unicode Values?

- Look at following examples-

```
>>>ord (‘A’)
65
>>>ord(‘a’)
97
```

```
>>>char(97)
a
>>>char(65)
A
```

String Slicing

- Look at following examples carefully-

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Word	R	E	S	P	O	N	S	I	B	I	L	I	T	Y
Reverse index	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

word = “RESPONSIBILITY”

word[0 : 14] will result into‘RESPONSIBILITY’

word[0 : 3] will result into‘RESP’

word[2 : 5] will result into‘SPO’

word[-7 : -3] will result into‘IBIL’

word[: 14] will result into‘RESPONSIBILITY’

word[: 5] will result into ‘RESPO’

word[3 :] will result into ‘PONSIBILITY’

String Functions

String.capitalize()	Converts first character to Capital Letter
String.find()	Returns the Lowest Index of Substring
String.index()	Returns Index of Substring
String.isalnum()	Checks Alphanumeric Character
String.isalpha()	Checks if All Characters are Alphabets
String.isdigit()	Checks Digit Characters
String.islower()	Checks if all Alphabets in a String.are Lowercase
String.isupper()	returns if all characters are uppercase characters
String.join()	Returns a Concatenated String
String.lower()	returns lowercased string
String.upper()	returns uppercased string
len()	Returns Length of an Object
ord()	returns Unicode code point for Unicode character
reversed()	returns reversed iterator of a sequence
slice()	creates a slice object specified by range()

Assignment

1. WAP to print the following pattern
   1. INDIA
      INDI
      IND
      IN
      I
   2. I
      IN
      IND
      INDI
      INDIA
2. WAP to search a substring from a given line of string.

3. WAP to find the length of a string.