UNIT: IV	Strings	Lectures: 07 Hrs

- **Strings and Operations:**
- concatenation, appending, multiplication and slicing
- Strings are immutable, strings formatting operator, built in string methods and functions.
- Slice operation, ord() and chr() functions, in and not in operators,
- comparing strings, Iterating strings
- the string module.

Strings

- String is a sequence of characters.
- String may contain alphabets, numbers and special characters.
- Usually strings are enclosed within a single quotes and double quotes.
- Strings is **immutable** in nature.
- **Example:**

a=„hello world“

b=“Python”

Inbuilt String functions

- Python mainly contains 3 inbuilt string functions.
- They are

–len()

–max()

–min()

- len()- Find out the length of characters in string
- min()- Smallest value in a string based on ASCII values
- max()- Largest value in a string based on ASCII values

Example for Inbuilt string functions

```
name=input("Enter Your name:")

print("Welcome",name)

print("Length of your name:",len(name))

print("Maximum value of chararacter in your name", max(name))

print("Minimum value of character in your name",min(name))
```

Strings Concatenation

- The + operator used for string concatenation.

Example:

a=“Hai”

b=“how are you”

c=a+b

print(c)

```
Haihow are you
>>> a="Hai"
>>> b=" how are you"
>>> c=a+b
>>> print(c)
Hai how are you
```

```
>>> a="Hai"
>>> b=' how are you'
>>> c=a+b
>>> print(c)
Hai how are you
```

Operators on String

- The Concatenate strings with the “*” operator can create multiple concatenated copies.
- Example:

```
>>> print("Python"*10)
PythonPythonPythonPythonPythonPythonPython PythonPythonPythonPython
```

String Slicing

- Slicing operation is used to return/select/slice the particular substring based on user requirements.
- A segment of string is called slice.
- **Syntax: string_variablename [start:end]**

String Slice example:
s="Hello"

```
>>> s="hello"
>>> s[1:4]
'ell'
>>> s[1:]
'ello'
>>> s[:]
'hello'
>>> s[1:100]
'ello'
>>> s[-1]
'o'
>>> s[::]
'hello'
>>> s[:-3]
'he'
```

Strings are immutable

- Strings are immutable character sets.
- Once a string is generated, you cannot change any character within the string.

```
>>> a="python program"
>>> a[0]
'p'
>>> a[0]="b"
Traceback (most recent call last):
  File "<pyshell#16>", line 1, in <module>
    a[0]="b"
TypeError: 'str' object does not support item assignment
>>> a[0]
'p'
```

String Comparision

- We can compare two strings using comparision operators such as ==, !=, <,<=,>, >=
- Python compares strings based on their corresponding ASCII values.

Example of string comparision

```
str1="green"
str2="black"
print("Is both Equal:", str1==str2)
print("Is str1> str2:", str1>str2)
print("Is str1< str2:", str1<str2)
```

String formatting operator

- •String formatting operator % is unique to strings.
- **•Example:**
- print(**"My name is %s and i secured %d marks in python" % ("Arba",92))**
- **•Output:**
- My name is Arbaz and i secured 92 marks in python

i) Converting string functions

captitalize()	Only First character capitalized
lower()	All character converted to lowercase
upper()	All character converted to uppercase
title()	First character capitalized in each word
swapcase()	Lower case letters are converted to Uppercase and Uppercase letters are converted to Lowercase
replace(old,new)	Replaces old string with nre string

Program:

```
str=input("Enter any string:")
print("String Capitalized:", str.capitalize())
print("String lower case:", str.lower())
print("String upper case:", str.upper())
print("String title case:", str.title())
print("String swap case:", str.swapcase())
print("String replace case:",str.replace("python","python
programming"))
```

ii)Formatting String functions

center(width)	Returns a string centered in a field of given width
ljust(width)	Returns a string left justified in a field of given width
rjust(width)	Returns a string right justified in a field of given width
format(items)	Formats a string

Program:

```
a=input("Enter any string:")
print("Center alignment:", a.center(20))
print("Left alignment:", a.ljust(20))
print("Right alignment:", a.rjust(20))
```

```
Enter any string:welcome
Center alignment:       welcome
Left alignment: welcome
Right alignment:                welcome
```

Testing String/Character

isalnum()	Returns true if all characters in string are alphanumeric and there is atleast one character
isalpha()	Returns true if all characters in string are alphabetic
isdigit()	Returns true if string contains only number character
islower()	Returns true if all characters in string are lowercase letters
isupper()	Returns true if all characters in string are uppercase letters
isspace()	Returns true if string contains only whitespace characters.

Program:

```
a=input("Enter any string:")
print("Alphanumeric:",a.isalnum())
print("Alphabetic:",a.isalpha())
print("Digits:",a.isdigit())
print("Lowecase:",a.islower())
print("Upper:",a.isupper())
```

```
Enter any string:python
Alphanumeric: True
Alphabetic: True
Digits: False
Lowecase: True
Upper: False
```

Searching for substring

Endswith()	Returns true if the strings ends with the substring
Startswith()	Returns true if the strings starts with the substring
Find()	Returns the lowest index or -1 if substring not found
Count()	Returns the number of occurrences of substring

Program

```
a=input("Enter any string:")
print("Is string ends with thon:", a.endswith("thon"))
print("Is string starts with good:", a.startswith("good"))
print("Find:", a.find("ython"))
print("Count:", a.count(“o"))
```

String Modules

- String module contains a number of functions to process standard Python strings
- **Mostly used string modules:**

string.upper()

string.upper()

string.split()

string.join()

string.replace()

string.find()

string.count()

Example

```
import string
text="Monty Python Flying Circus"
print("Upper:",  string.upper(text))
print("Lower:",   string.lower(text))
print("Split:", string.split(text))
print("Join:", string.join(string.split(test),"+"))
print("Replace:", string.replace(text,"Python", "Java"))
print("Find:",  string.find(text,"Python"))
print("Count", string.count(text,"n"))
```