	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

GITHUB

Aim: Write a program to create, concatenate and print a string and accessing substring from a given

string.

IDE:

Slicing and indexing are two fundamental concepts in Python. They help us access specific elements in a sequence, such as a string or (list and tuple).

Indexing in Python

Indexing is the process of accessing an element in a sequence using its position in the sequence (its index). In Python, indexing starts from 0, which means the first element in a sequence is at position 0, the second element is at position 1, and so on. To access an element in a sequence, you can use square brackets [] with the index of the element you want to access.

Let's consider the following example:

```
# create a string using double quotes
string1 = "ICT Department"
print(string1)
# create a string using single quotes
string1 = ' ICT Department '
print(string1)
```

Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
1   string1 = "ICT Department"
2   print(string1)
3   string1 = ' ICT Department '
4   print(string1)
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
ICT Department
 ICT Department
PS E:\SEM 3\PWP>
```

Access String Characters in Python
string2 = '3EK1'
# access 1st index element
print(string2 [1])
Output:

```
6   string2 = '3EK1'
7   # access 1st index element
8   print(string2 [1])
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
E
PS E:\SEM 3\PWP>
```

Negative Indexing:
Python allows negative indexing for its strings. For example,
string3 = 'ICT Department'
# access 4th last element
print(string3 [-4])
output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
10    string3 = 'ICT Department'
11    # access 4th last element
12    print(string3 [-4])
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
m
PS E:\SEM 3\PWP>
```

Slicing in Python

Slicing is the process of accessing a sub-sequence of a sequence by specifying a starting and ending index. In Python, you perform slicing using the colon: operator. The syntax for slicing is as follows:

Example:

sequence[start_index:end_index]

where start_index is the index of the first element in the sub-sequence and end_index is the index of the last element in the sub-sequence (excluding the element at the end_index). To slice a sequence, you can use square brackets [] with the start and end indices separated by a colon.

For example,
string4 = 'ICT Department'
# access character from 1st index to 3rd index
print(string4[1:4])
Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
14    string4 = 'ICT Department'
15    # access character from 1st index to 3rd index
16    print(string4[1:4])
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
CT
PS E:\SEM 3\PWP>
```

You can also omit either the start_index or the end_index in a slice to get all the elements from the beginning or end of the sequence. For example:
print(string4[:2])
print(string4[2:])
output:

```
14    string4 = 'ICT Department'
15    print(string4[:2])
16    print(string4[2:])
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
IC
T Department
PS E:\SEM 3\PWP>
```

In the first line of the above code, we have used slicing to get all the elements from the beginning of string4 up to (but not including) the element at index 2. In the second line, we have used slicing to get all the elements from index 2 to the end of string4.

Python Strings are Immutable

In Python, strings are immutable. That means the characters of a string cannot be changed. For example,
message = 'ICT Department'
message[0] = 'H'

print(message)
Output:
Type error (Strings are immutable)

However, we can assign the variable name to a new string. For example,

```
message = 'ICT'
# assign new string to message variable
message = 'ICT Department'
print(message)
```

Output:

```
22    message = 'ICT'
23    # assign new string to message variable
24    message = 'ICT Department'
25    print(message)
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
ICT Department
PS E:\SEM 3\PWP>
```

Python Multiline String

We can also create a multiline string in Python. For this, we use triple double quotes """ or triple single quotes '''.
For example,

```
# multiline string
message = """
ICT
Department
3EK1
"""
print(message)
```

Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
27  message = """
28  ICT
29  Department
30  3EK1
31  """
32  print(message)
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"

ICT
Department
3EK1

PS E:\SEM 3\PWP>
```

Python String Operations

Many operations can be performed with strings, which makes it one of the most used data types in Python.

1. **Compare Two Strings**
For example,

```
str1 = "ICT"
str2 = "Department"
str3 = "3EK1"
# compare str1 and str2
print(str1 == str2)
# compare str1 and str3
print(str1 == str3)
```

Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
34    str1 = "ICT"
35    str2 = "Department"
36    str3 = "3EK1"
37    # compare str1 and str2
38    print(str1 == str2)
39    # compare str1 and str3
40    print(str1 == str3)
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
False
False
PS E:\SEM 3\PWP>
```

2. **Join Two or More Strings**

In Python, we can join (concatenate) two or more strings using the + operator.

```
greet = "ICT"
name = "Department"
# using + operator
result = greet + name
print(result)
```

Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
42    greet = "ICT"
43    name = "Department"
44    # using + operator
45    result = greet + name
46    print(result)
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
ICTDepartment
PS E:\SEM 3\PWP>
```

Python String Length

In Python, we use the len() method to find the length of a string. For example,

```
greet = 'ICT'
# count length of greet string
print(len(greet))
```

Output:

```
48    greet = 'ICT'
49    # count length of greet string
50    print(len(greet))
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
3
PS E:\SEM 3\PWP>
```

String Membership Test

We can test if a substring exists within a string or not, using the keyword in.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

print('a' in 'program')
print('at' not in 'battle')
Output:

```
52  print('a' in 'program')
53  print('at' not in 'battle')
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_3.py"
True
False
PS E:\SEM 3\PWP>
```

Methods of Python String

Python String upper()
The upper() method converts all lowercase characters in a string into uppercase characters and returns it.
message = 'python is fun'
# convert message to uppercase
print(message.upper())
Output:

```
67  message = 'python is fun'
68  print(message.upper())
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
PYTHON IS FUN
```

Python String lower()
The lower() method converts all uppercase characters in a string into lowercase characters and returns it.
message = 'PYTHON IS FUN'
print(message.lower())
Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
70  message = 'PYTHON IS FUN'
71  print(message.lower())
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
python is fun
PS E:\SEM 3\PWP>
```

Python String replace()

The replace() method replaces each matching occurrence of a substring with another string.

text = 'CE Department'
replaced_text = text.replace('CE', 'ICT')
print(replaced_text)

Output:

```
73  text = 'CE Department'
74  replaced_text = text.replace('CE', 'ICT')
75  print(replaced_text)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
ICT Department
PS E:\SEM 3\PWP>
```

Python String find()

The find() method returns the index of first occurrence of the substring (if found). If not found, it returns -1.

message = 'Python is a fun programming language'
# check the index of 'fun'
print(message.find('fun'))

Output:

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
77    message = 'Python is a fun programming language'
78    print(message.find('fun'))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
12
PS E:\SEM 3\PWP>
```

Python String rstrip()

The rstrip() method returns a copy of the string with trailing characters removed (based on the string argument passed).

```
title = 'Python Programming   '
result = title.rstrip()
print(result)
```

Output:

```
81    title = 'Python Programming    '
82    result = title.rstrip()
83    print(result)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
Python Programming
PS E:\SEM 3\PWP>
```

Python String split()

The split() method breaks down a string into a list of substrings using a chosen separator.

```
text = 'Python is fun'
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

# split the text from space
print(text.split())
Output:

```
85  text = 'Python is fun'
86  print(text.split())
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
['Python', 'is', 'fun']
PS E:\SEM 3\PWP>
```

Python String startswith()
The startswith() method returns True if a string starts with the specified prefix(string). If not, it returns False.
message = 'Python is fun'
# check if the message starts with Python
print(message.startswith('Python'))
Output:

```
88  message = 'Python is fun'
89  print(message.startswith('Python'))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
True
PS E:\SEM 3\PWP>
```

Python String isnumeric()
The isnumeric() method checks if all the characters in the string are numeric.
pin = "523"
# checks if every character of pin is numeric

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

print(pin.isnumeric())
Output:

```
91    pin = "523"
92    print(pin.isnumeric())
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
True
PS E:\SEM 3\PWP>
```

Python String index()

The index() method returns the index of a substring inside the string (if found). If the substring is not found, it raises an exception.

text = 'Python is fun'
# find the index of is
result = text.index('is')
print(result)
Output:

```
94    text = 'Python is fun'
95    result = text.index('is')
96    print(result)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
7
PS E:\SEM 3\PWP>
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

Python String Formatting (f-Strings)

Python f-Strings makes it easy to print values and variables. For example,

name = 'Cathy'
country = 'UK'
print(f'{name} is from {country}')

Output:

```
 99    name = 'Cathy'
100    country = 'UK'
101    print(f'{name} is from {country}')

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
Cathy is from UK
PS E:\SEM 3\PWP>
```

Python Raw String

Python strings become raw strings when they are prefixed with r or R, such as r'...' and R'...'. Raw strings treat backslashes () as literal characters. Raw strings are useful for strings with a lot of backslashes, like regular expressions or directory paths.

str = "This is a \n normal string example"
print(str)
raw_str = r"This is a \n raw string example"
print(raw_str)

Output

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
104    str = "This is a \n normal string example"
105    print(str)
106    raw_str = r"This is a \n raw string example"
107    print(raw_str)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\lab_2.py"
This is a
 normal string example
This is a \n raw string example
PS E:\SEM 3\PWP>
```

Post Lab Exercise:

a. Write a Python program to reverse a string.

```
ex-3.py > ...
1    user= input("Enter String: ")
2    rev = user[::-1]
3    print("Reversed String: ", rev)
```

```
DEBUG CONSOLE    TERMINAL    PORTS

TERMINAL                                    powers

PS E:\SEM 3\PWP> python .\ex-3.py
Enter String: hello I am Ruhaan
Reversed String:  naahuR ma I olleh
PS E:\SEM 3\PWP>
```

b. Write a Python program to check if a string is a palindrome.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
#Question 2
user = input("Enter a string: ")
rev = "".join(reversed(user))
if user == rev:
    print("Palindrome")
else:
    print("Not Palindrome")
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\post_lab_3.py"
Enter a string: mom
Palindrome
PS E:\SEM 3\PWP>
```

c. Write a Python program to check if a string contains only digits.

```
#Question 3
user = input("Enter a string: ")
if user.isdigit():
    print("Only digits")
else:
    print("It has other characters too")

```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\post_lab_3.py"
Enter a string: 123
Only digits
PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\post_lab_3.py"
Enter a string: i am 19 years old
It has other characters too
PS E:\SEM 3\PWP>
```

d. Write a Python program to find the longest word in a sentence.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a program to create, concatenate and print a string and accessing substring from a given string.	
Experiment No: 03	**Date:**	**Enrollment No: 92400133055**

```
21  #Question 4
22  user = input("Enter a sentence: ")
23  words = user.split()
24  longest = max(words, key=len)
25  print("Longest word:", longest)
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\post_lab_3.py"
Enter a sentence: Feburary is the shortest month
Longest word: Feburary
PS E:\SEM 3\PWP>
```

e. Write a Python program to find the length of the last word in a sentence.

```
27  #Question 5
28  user = input("Enter a sentence: ")
29  words = user.split()
30  last_word = words[-1]
31  print("Length of last word:", len(last_word))
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\SEM 3\PWP> python -u "e:\SEM 3\PWP\Class Tutorials\post_lab_3.py"
Enter a sentence: this is august month
Length of last word: 5
PS E:\SEM 3\PWP>
```