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: 92510133028**

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

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
ICT Department
 ICT Department
PS E:\PWP>
```

Access String Characters in Python

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

Output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
E
PS E:\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: 92510133028**

Negative Indexing:
Python allows negative indexing for its strings. For example,

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

output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
m
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
CT
PS E:\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:])
```

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: 92510133028**

output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
CT
IC
T Department
PS E:\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

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)
```

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
ICT Department
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py

ICT
Department
3EK1

PS E:\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: 92510133028**

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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
False
False
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
ICTDepartment
PS E:\PWP>
```

Python String Length

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

```
greet = 'ICT'
```

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: 92510133028**

# count length of greet string
print(len(greet))
Output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
3
```

String Membership Test

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

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

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
3
True
False
```

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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.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'
# convert message to lowercase
print(message.lower())
Output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
python is fun
PS E:\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: 92510133028**

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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
ICT Department
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
12
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Python Programming
PS E:\PWP>
```

Python String split()

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

```
text = 'Python is fun'
# split the text from space
```

print(text.split())

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
['Python', 'is', 'fun']
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
True
PS E:\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
print(pin.isnumeric())
```

Output:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
True
PS E:\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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
7
PS E:\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: 92510133028**

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:

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Cathy is from UK
PS E:\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

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
This is a
 normal string example
This is a \n raw string example
PS E:\PWP>
```

Post Lab Exercise:

a. Write a Python program to reverse a string.

```
s = input("Enter a string: ")
reversed_string = s[::-1]
print("Reversed string:", reversed_string)
```

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: 92510133028**

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Enter a string: red
Reversed string: der
PS E:\PWP>
```

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

```
s = input("Enter a string: ")
if s == s[::-1]:
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Enter a string: black
The string is not a palindrome.
PS E:\PWP>
```

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

```
s = input("Enter a string: ")
if s.isdigit():
print("The string contains only digits.")
else:
print("The string does not contain only digits.")
```

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Enter a string: blue
The string does not contain only digits.
PS E:\PWP>
```

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

```
sentence = input("Enter a sentence: ")
words = sentence.split()
longest_word = max(words, key=len)
print("The longest word is:", longest_word)
```

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Enter a sentence: she is a girl
The longest word is: girl
PS E:\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: 92510133028**

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

```
sentence = input("Enter a sentence: ").strip()
words = sentence.split()
if words:
    last_word = words[-1]
    print("Length of the last word:", len(last_word))
else:
    print("No words in the input.")
```

```
PS E:\PWP> & C:/Users/keshv/AppData/Local/Programs/Python/Python313/python.exe e:/PWP/harikeshsirexperiment/exp1.3.py
Enter a sentence: i learn python
Length of the last word: 6
PS E:\PWP>
```