Experiment No: 6

Aim: Program to Implement String Manipulations

Procedure:

How to create a string in Python?

```
# defining strings in Python
# all of the following are equivalent
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

my_string = '''Hello'''
print(my_string)

# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
           the world of Python"""
print(my_string)
```

How to access characters in a string?

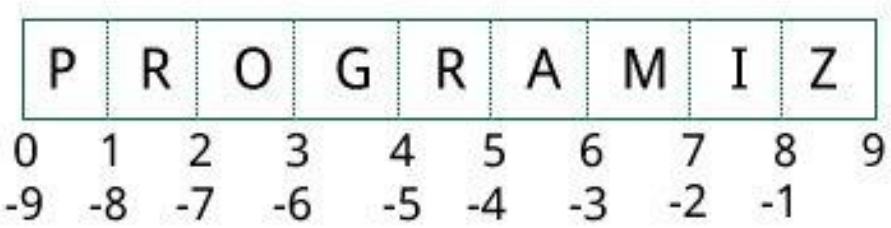
```
#Accessing string characters in Python
str = 'programiz'
print('str = ', str)

#first character
print('str[0] = ', str[0])

#last character
print('str[-1] = ', str[-1])

#slicing 2nd to 5th character
print('str[1:5] = ', str[1:5])
```

```
#slicing 6th to 2nd last character
print('str[5:-2] = ', str[5:-2])
```

String Slicing:

Slicing can be best visualized by considering the index to be between the elements as shown below.

If we want to access a range, we need the index that will slice the portion from the string.



Python String Operations

Concatenation of Two or More Strings

```
# Python String Operations
str1 = 'Hello'
str2 ='World!'

# using +
print('str1 + str2 = ', str1 + str2)

# using *
print('str1 * 3 =', str1 * 3)
```

If we want to concatenate strings in different lines, we can use parentheses.

```
>>> # two string literals together
>>> 'Hello ''World!'
```

```
'Hello World!'

>>> # using parentheses
>>> s = ('Hello '
...      'World')
>>> s
'Hello World'
```

Example to count the total no of L ‘s in the string.

```
# Iterating through a string
count = 0
for letter in 'Hello World':
    if(letter == 'l'):
        count += 1
print(count,'letters found')
```

Built-in functions to Work with Python

```
str = 'cold'

# enumerate()
list_enumerate = list(enumerate(str))
print('list(enumerate(str) = ', list_enumerate)

#character count
print('len(str) = ', len(str))
```

Accessing String Elements

str='Computer Sciene'

print('str-', str)

print('str[0]-', str[0])

print('str[1:4]-', str[1:4])

print('str[2:]-', str[2:])

```
print('str *2-', str *2 )
print("str +'yes'-", str +'yes')
```

```
str='Computer Sciene‘
for i in str:
print(i)
```

String Manipulation

String functions and methods

Method	Result
str.capitalize()	To capitalize the string
str.find(sub)	**To find the substring position**
str.isalnum()	String consists of only alphanumeric characters (no symbols)
str.isalpha()	String consists of only alphabetic characters (no symbols)
str.islower()	String's alphabetic characters are all lower case
str.isnumeric()	String consists of only numeric characters
str.isspace()	String consists of only whitespace characters
str.istitle()	String is in title case
str.isupper()	String's alphabetic characters are all upper case
str.lstrip(char) str.rstrip(char)	Returns a copy of the string with leading/trailing characters

Note: Try to implement atleast one example from the above list of methods.