

Practical 2

To understand the uses of strings

To study various inbuilt functions in string

String functions in Python are a set of built-in methods that can be used to manipulate strings. These functions can be used to perform a variety of tasks, such as:

Converting a string to uppercase or lowercase.

Finding the location of a substring in a string.

Replacing all occurrences of a substring in a string with another substring.

Splitting a string into a list of substrings.

Joining a list of strings into a single string.

Trimming whitespace from a string, etc.

We will discuss the following functions here:

1. title() - This in-built function is used to convert string to titular case.
2. isalnum() - This in-built function is used to check if the string has alphabetic as well as numeric characters or not.
3. find() - This in-built function is used to find a particular character in string.
4. index() - This in-built function is used to find a character at a particular index.
5. isupper() - This in-built function is used to check if the complete string is upper-case or not.

```
example = "A Trial String is this"

print(f"Original: {example}")

title = example.title()
print(f"title(): {title}")

isalnum = example.isalnum()
print(f"isalnum(): {isalnum}")

find = example.find('-')
print(f"find(): {find}")

index = example.index('is')
print(f"index(): {index}")

upper = example.isupper()
print(f"isupper(): {upper}")
```

```
Original: A Trial String is this
title(): A Trial String Is This
isalnum(): False
find(): -1
index(): 15
isupper(): False
```

Create a string that contains hyphens(-) and write a program to split the given string based on hyphens(-) and display each substring

```
example = "A Trial String is this"
split_example = example.split(sep='is')
print(f"Original: {example}")
print(f"After splitting: {split_example}")

Original: A Trial String is this
After splitting: ['A Trial String ', ' th', '']
```

Write a program to reverse a string in python using 5 different ways

```
example = "A Trial String is this"
print(f"Original String: {example}")

print(f"(Using Slicing): {example[::-1]}")

print((f"(Using 'reversed')": {''.join(reversed(example))}"))

new = ""
for _ in range(len(example) - 1, -1, -1):
    new += example[_]
print(f"(Using for loop): {new}")

new = ""
i = len(example) - 1
while i >= 0:
    new += example[i]
    i -= 1
print(f"(Using while loop): {new}")

def reverse_string(reverese_string):
    if len(reverese_string) == 0:
        return ""
    else:
        return reverese_string[-1] + reverse_string(reverese_string[:-1])

print(f"(Using a Recursive function): {reverse_string(example)}")
```

```
Original String: A Trial String is this
(Using Slicing): siht si gnirtS lairT A
(Using 'reversed'): siht si gnirtS lairT A
(Using for loop): siht si gnirtS lairT A
(Using while loop): siht si gnirtS lairT A
(Using a Recursive function): siht si gnirtS lairT A
```

Write a program to find the last position of a substring "Dev" in the given string

```
example = "Dev is a student. Dev studies CSE. Dev lives in India."

find_example = example.rfind('Dev')
print(f"Last occurrence of 'Dev' found at: {find_example}")

Last occurrence of 'Dev' found at: 35
```

Write a python program to change a given string to a newly string where the first and last characters have been exchanged

```
example = "A Trial String is this"

swap = example[-1] + example[1:-1] + example[0]

print(f"After swapping: {swap}")

After swapping: gnother StrinA
```

Write a python program that takes input from the user and displays that input back in upper and lower cases

```
example = input("Enter a string: ")
print(f"Input String: {example}")
print(f"Upper-case: {example.upper()}")
print(f"Lower-case: {example.lower()}")

Input String: A Trial String is this
Upper-case: A TRIAL STRING IS THIS
Lower-case: a trial string is this
```

Write a program to swap comma(,) and full-stop(.) in a string

```
example = "There are 26 Alphabets. Some of the characters are a, b, etc."
swap = example.replace(',', '@')
swap = swap.replace('.', ',')
swap = swap.replace('@', '.')
print(f"After swapping: {swap}")
```

After swapping: There are 26 Alphabets, Some of the characters are a. b. etc,

Create a string made up of first, middle and last character of another string

```
example = input("Enter a string with odd number of characters: ")
new = example[0] + example[(len(example) - 1)//2] + example[-1]
print(f"Input string with odd number of characters: {example}")
print(f"New string: {new}")
```

Input string with odd number of characters: A Trial String is this
New string: Ars

Given two strings S1 and S2, write a program to create a new string S3 by appending S2 in the middle of S1.

```
S1 = "This is a string"
S2 = " not"
S3 = S1[0: (len(S1) - 1)//2] + S2 + S1[(len(S1) - 1)//2:]
print(f"S1: {S1}")
print(f"S2: {S2}")
print(f"S3: {S3}")
```

S1: This is a string
S2: not
S3: This is not a string

Write a python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

```
# Strings
example = "String"
two_ex = "Not String"

# Joining the strings
swap = ' '.join([example, two_ex])

# Performing swapping of first characters of both the original string in the new string using indexing
new_swap = swap[len(example) + 1] + swap[1:len(example) + 1] + swap[0] + swap[len(example) + 2:]

# Output
print(f"Original String: {swap}")
print(f"String after Swapping: {new_swap}")

Original String: String Not String
String after Swapping: Ntring Sot String
```