



SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya Institute of Management

Department of Data Science and Technology

Practical No: 09

Subject: Python Programming Lab

MCA / Sem I / Python Programming [Course Code : 217P09L102]

ROLL No: 09 _____

DATE: 22/09/2023

FULL NAME: Atharv Ankush Desai _____

Aim:	Understanding Python Strings
Topics Covered:	String & String Mutation, String Methods, String Slicing, String Iteration
Problem Statement:	<p>1. Write a Python program that takes two strings as input and concatenates them. Ensure that the final string is in uppercase. Calculate and print the length of a user-entered string. Remove all leading and trailing spaces from a given string.</p> <p>2. a) Write a program to find and print the index of the first occurrence of a substring in each string. b) Create a Python function that takes a sentence as input and replaces all occurrences of a specified word with a user-defined word.</p> <p>3. a) Write a Python function that extracts the domain name from an email address using string slicing. b) Given a list of words separated by spaces, write a program to split the input into individual words and count the number of words.</p> <p>4. Create a Python program that accepts a user's name and age, and then prints a formatted message like "My name is [name] and I am [age] years old". Format a given string to title case,</p>

	<p>and then capitalize the first character of the result.</p> <p>5. Write a Python program to count the number of times a specified character appears in a given string. Implement a function that searches for all occurrences of a specified word in a paragraph and counts them.</p> <p>6. a) Develop a program that reverses a given string without using string slicing. b) Create a Python function that accepts a sentence and removes all punctuation marks from it.</p> <p>7. a) Write a Python program that takes a list of words and joins them into a single string with a comma and space as the delimiter. b) Develop a program that reads a paragraph of text and splits it into sentences based on period (.) as the delimiter.</p> <p>8. a) Create a Python function that accepts a string and checks whether it starts with an uppercase letter. b) Implement a program that checks if a given string ends with a specific suffix.</p> <p>9. a) Write a program that removes all leading and trailing zeros from a user-entered string of numbers. b) Develop a function that strips a given string of a specific character from both ends.</p> <p>10. Compare two strings entered by the user and determine if they are equal. Find the index of the first occurrence of a user-specified character in a given string.</p>
Theory:	<p>1. len(): Returns the length (the number of characters) of a string.</p> <p>2. strip(): Removes leading and trailing whitespace (or specified characters) from a string.</p> <p>3. rstrip(): Removes trailing whitespace (or specified characters) from the right side of a string.</p> <p>4. lstrip(): Removes leading whitespace (or specified characters) from the left side of a string.</p> <p>5. find(): Searches for a substring within a string and returns the index of the first occurrence (or -1 if not found).</p> <p>6. rfind(): Searches last occurrence for a substring within a string and returns the index of the last occurrence (or -1 if not found).</p> <p>7. index(): Similar to find(), but raises an exception if the substring is not found.</p>

	<p>8. <code>rindex()</code>: Similar to <code>rfind()</code>, last occurrence but raises an exception if the substring is not found.</p> <p>9. <code>count()</code>: Counts the number of non-overlapping occurrences of a substring in a string.</p> <p>10. <code>replace()</code>: Replaces occurrences of a specified substring with another string.</p> <p>11. <code>split()</code>: Divides a string into a list of substrings based on a specified.</p> <p>12. <code>join()</code>: Combines a list of strings into a single string using a specified</p> <p>13. <code>upper()</code>: Converts a string to uppercase.</p> <p>14. <code>lower()</code>: Converts a string to lowercase.</p> <p>15. <code>swapcase()</code>: Swaps the case (upper to lower and vice versa) of characters in a string.</p> <p>16. <code>title()</code>: Converts the first character of each word in a string to uppercase.</p> <p>17. <code>capitalize()</code>: Capitalizes the first character of a string.</p> <p>18. <code>startswith()</code>: Checks if a string starts with a specified prefix.</p> <p>19. <code>endswith()</code>: Checks if a string ends with a specified suffix.</p>
Code:	<pre> 1. def cusConcat(str1, str2): return str1.upper() + str2.upper() myText = " Watashi no namaiwa Atharv !" res = cusConcat(myText, "LOL") print(f'Concat => {res}') def cusLen(text): return len(text.strip()) myText = " Watashi no namaiwa Atharv !" res = cusLen(myText) print(f'Custom Length => {res}') 2. def cusReplace(text): myDic = {'paragraph':'Line','Nice':"LOM"} for word in text.split(): if word in [*myDic.keys()]: text = text.replace(word,myDic.get(word)) return text </pre>

```
myText = "this is a paragraph paragraph Nice"
res = cusReplace(myText)
print(f'OG => {myText} Replaced => {res}')
```

```
myWord = "Line"
res = myText.find('Nice') # or find()
print(f'First hit indx of {myWord} => {res}')
```

```
3.
def cusSplit(text):
    if text.count('@') > 0:
        return text.split('@')
    return 'Not Valid Mail'
```

```
# * var
myEmail = 'atharvdesai2002@gmail.com'
```

```
def countWords(text):
    splText,splWords = text.split(),None
    splWords = [*splText]
    for word in splWords:
        print(f'{word} => {splText.count(word)}')
```

```
myText = " Watashi no namaiwa Atharv Atharv !"
username, domain = cusSplit(myEmail)
print(f'Username => {username} Domain => {domain}')
```

```
print(f'\n-----\n")
print('Word Counter')
countWords(myText)
```

```
4.
name,age = input(f'Your Name =>') , input(f'Your Age =>')
res = f'my name is {name} and i am {age} years old'
```

```
print(f'Title => {res.title()}')
print(f'Capital => {res.capitalize()}')
```

```
5.
def countChars(text):
    print(f'{text} has :')
    text = text.replace(' ','')
    splText = [*text]
    splWords = [*splText]
    for word in enumerate(splWords,1):
        print(f'{word} => {splText.count(word[1])}')
```

```
myText = "This is a Paragraph"
```

	<pre> countChars(myText) 6. def cusReverse(text): return text[::-1] def puncRemover(text): punctuations = "!@#\$%^&*()_+=[',./\]" for punc in punctuations: text = text.replace(punc,"") print(text) myText = "This is, a paragraph" res = cusReverse(myText) print(f"{myText}" revered is => '{res}') print(f"\n-----\n") puncRemover(myText) 7. def cusJoiner(*word): return ','.join([*word]) def cusSenspliter(text): text = text.split('.') print(text) print(f'Line of Para are :') for sen in enumerate(text,1): print(f'{sen[0]} - {sen[1]}') res = cusJoiner("THIS","IS","A","Parafgrapg") print(f'Result of Join() => {res}') print(f"\n-----\n") myPara = "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum" cusSenspliter(myPara) 8. def cusUcheck(text): return True if text[0].isupper() else False myText = "This is a Para" res = cusUcheck(myText) print(f'{myText} has the first letter is capital => {res}') </pre>
--	--

	<pre> print("\n-----\n') mySuffix = "lol" # 'Para','ara','ra' print(f'{myText} ends with Suffix {mySuffix} => {myText.endswith(mySuffix)}') 9. def cusRemover0(text): return text.strip('0') def cusStrip(text,char): return text.strip(char) myText = input(f'Enter You Text => ') res= cusRemover0(myText) print(f'{myText} stripper of zeros => {res}') print("\n-----\n') res= cusStrip(res,"9") print(f'{myText} specific caracter stip from both end => {res}') 10. def areSame(text1,text2): return True if text1 == text2 else False def charOccur(text,char): return f'is at index {text.index(char)}' if text.find(char) != -1 else f'Not in {text}' myText = 'This is a Paragraph' inp1,inp2 = input(f'Enter Your Input 1 =>'),input(f'Enter Your Input 2 =>') print(f'{inp1} {inp2} are same => {areSame(inp1,inp2)}') chartoSearch = input(f'Char to Search =>') print("\n-----\n') print(myText) res = charOccur(myText,chartoSearch) print(f'"In {myText} {chartoSearch} ' => {res}') </pre>
Screenshot of Output:	1.

```
Concat =>      WATASHI NO NAMAIWA ATHARV      !LOL
Length After Strip => 29
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

2.

```
OG => this is a paragraph paragraph Nice Replaced => this is a Line Line LOM
First hit indx of Line => 30
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

3.

```
Username => atharvdesai2002 Domain => gmail.com
```

```
-----

Word Counter
Atharv => 2
Watashi => 1
namaiwa => 1
no => 1
! => 1
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

4.

```
Your Name =>Atharv
Your Age =>21
Title => My Name Is Atharv And I Am 21 Years Old
Capital => My name is atharv and i am 21 years old
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

5.

```
This is a Paragraph has :
```

```
(1, 'r') => 2
```

```
(2, 'g') => 1
```

```
(3, 'h') => 2
```

```
(4, 'p') => 1
```

```
(5, 'T') => 1
```

```
(6, 'i') => 2
```

```
(7, 'a') => 4
```

```
(8, 's') => 2
```

```
(9, 'P') => 1
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.□
```

6.

```
'This is, a paragraph'' revered is => 'hpargarap a ,si sihT'
```

```
-----
```

```
This is a paragraph
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.□
```

7.


```
Result of Join() => THIS,IS,A,Parafgrapg
```

```
-----
```

```
['Lorem Ipsum is simply dummy text of the printing and typesetting industry', " Lorem Ipsum has been the ind  
xt ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type speci  
rised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently w  
ftware like Aldus PageMaker including versions of Lorem Ipsum']
```

```
Line of Para are :
```

```
1 - Lorem Ipsum is simply dummy text of the printing and typesetting industry
```

```
2 - Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer t  
scrambled it to make a type specimen book
```

```
3 - It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, an  
top publishing software like Aldus PageMaker including versions of Lorem Ipsum
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.[]
```

8.

```
This is a Para has the first letter is capital => True
```

```
-----
```

```
This is a Para ends with Suffix lol => False
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.[]
```

9.

```
Enter You Text => 000009999Athrav99400000004
```

```
000009999Athrav99400000004 stipper of zeros => 9999Athrav99400000004
```

```
-----
```

```
000009999Athrav99400000004 specific caracter stip from both end => Athrav99400000004
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.[]
```

10.

	<pre> Enter Your Input 1 =>Atharv Enter Your Input 2 =>Nidhi Atharv Nidhi are same => False Char to Search =>P ----- This is a Paragraph 'In This is a Paragraph P ' => is at index 10 ...Program finished with exit code 0 Press ENTER to exit console. </pre>
Observations:	<ol style="list-style-type: none"> 1. Strings in Python are immutable, which means they remain unchanged after they are created. 2. Python strings are typically enclosed in single ' ' or double " " quotation marks 3. Python Str() class has a large collection of built-in string manipulation methods that help with tasks like searching, replacing, splitting, and formatting.
Conclusion:	<p>Text preprocessing in natural language processing (NLP) in which raw text data is cleaned, tokenized, and transformed into a format suitable for machine learning models.</p> <ol style="list-style-type: none"> 2. Used in Sentiment Analysis determining the emotional tone or sentiment expressed in text, which is frequently used to gauge public opinion or customer feedback. 3. Categorizing news articles into predefined topics or classes in order to facilitate efficient content organization and retrieval.

Subject-In-Charge:

Sign: _____

Prof. Mayura Nagar

