



**SOMAIYA**  
**VIDYAVIHAR UNIVERSITY**

K J Somaiya Institute of Management

**Department of Data Science and Technology**

**Practical No: 08**

**Subject: Python Programming Lab**

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

**ROLL No: 09** \_\_\_\_\_

**DATE: 21/09/2023**

**FULL NAME: Atharv Ankush Desai** \_\_\_\_\_

<b>Aim:</b>	<b>Understanding Python Strings</b>
<b>Topics Covered:</b>	String & String Mutation, String Methods, String Slicing, String Iteration
<b>Problem Statement:</b>	<ol style="list-style-type: none"><li>1. WAPP to check if an entered string is a palindrome or not?</li><li>2. WAPP to split a given string into lines.</li><li>3. WAPP to remove specific characters from the string. (e.g. removing ! from the string "Hello World!")</li><li>4. WAPP to demonstrate more on string function -i) len() ii) strip() iii)rstrip() iv) lstrip() v) find() vi) rfind() vii) index() viii) rindex() ix) count() x) replace() xi) split() xii) join() xiii) upper() xiv) lower() xv) swapcase() xvi) title() xvii) capitalize() xviii) startswith() xix) endswith()</li></ol>
<b>Theory:</b>	<ol style="list-style-type: none"><li>1. len(): Returns the length (the number of characters) of a string.</li><li>2. strip(): Removes leading and trailing whitespace (or specified characters) from a string.</li><li>3. rstrip(): Removes trailing whitespace (or specified characters) from the right side of a string.</li><li>4. lstrip(): Removes leading whitespace (or specified characters) from the left side of a string.</li></ol>

	<ol style="list-style-type: none"> <li>5. find(): Searches for a substring within a string and returns the index of the first occurrence (or -1 if not found).</li> <li>6. rfind(): Searches last occurrence for a substring within a string and returns the index of the last occurrence (or -1 if not found).</li> <li>7. index(): Similar to find(), but raises an exception if the substring is not found.</li> <li>8. rindex(): Similar to rfind(), last occurrence but raises an exception if the substring is not found.</li> <li>9. count(): Counts the number of non-overlapping occurrences of a substring in a string.</li> <li>10. replace(): Replaces occurrences of a specified substring with another string.</li> <li>11. split(): Divides a string into a list of substrings based on a specified delimiter.</li> <li>12. join(): Combines a list of strings into a single string using a specified delimiter.</li> <li>13. upper(): Converts a string to uppercase.</li> <li>14. lower(): Converts a string to lowercase.</li> <li>15. swapcase(): Swaps the case (upper to lower and vice versa) of characters in a string.</li> <li>16. title(): Converts the first character of each word in a string to uppercase.</li> <li>17. capitalize(): Capitalizes the first character of a string.</li> <li>18. startswith(): Checks if a string starts with a specified prefix.</li> <li>19. endswith(): Checks if a string ends with a specified suffix.</li> </ol>
<b>Code:</b>	<pre> 1. # pal func def ispalindrome(str):     if len(str):         return "is" if str == str[::-1] else 'is not'     else:         return "Invalid"  myStr = input('Enter Your Text =&gt; ') </pre>

```
res = ispalindrome(myStr)
print(f'{myStr} {res} a Palindrome')
```

2.

```
myStr = "This is a paragraph"
myStrlist = myStr.split()
print(f'{myStr} => {myStrlist}')
```

```
for item in enumerate(myStrlist):
    print(f'{item}')
```

3.

```
myStr = "Hello World!"
# replace with no-char
myStr = myStr.replace('!', "")
print(myStr)
```

4.

```
myStr = "  This is a Paragraph  !"
myStr1 = "  Watashi no namaiwa Atharv  !"
```

```
# len
print('\nLen()\n-----')
print(f'Length of {myStr} => {len(myStr)}')
```

```
# strip
print('\nStrip()\n-----')
myStr = myStr.strip()
print(f'Strip of {myStr} => {myStr}')
```

```
# lstrip
print('\nlstrip()\n-----')
myStr = myStr.lstrip()
print(f'Strip of {myStr} => {myStr}')
```

```
# find
print('\nFind()\n-----')
res = myStr.find('This')
print(f'""This' is in {myStr} => {res}')
```

```
# rfind()
print('\nrFind()\n-----')
res = myStr.rfind('is')
print(f'""This' is in {myStr} => {res}')
```

```

# index()
print("\nIndex()\n-----')
res = myStr.index('is')
print(f"'is' is at {myStr} idx => {res}")

# rindex()
print("\nrIndex()\n-----')
res = myStr.rindex('is')
print(f"'is' is at {myStr} idx => {res}")

# count()
print("\nrCount()\n-----')
res = myStr.count('is')
print(f"'is' {myStr} count is => {res}")

# replace()
print("\nReplace()\n-----')
res = myStr.replace('is','was')
print(f"'is' {myStr} replaced => {res}")

# split()
print("\nSplit()\n-----')
res = myStr.split()
print(f"'is' {myStr} Splited => {res}")

# join()
print("\nSplit()\n-----')
res = '#'.join(myStr1.split())
print(f"'is' {myStr1} Joined => {res}")

# upper()
print("\nUpper()\n-----')
res = myStr1.upper()
print(f"'is' {myStr1} Upper => {res}")

# lower()
print("\nLower()\n-----')
res = myStr1.lower()
print(f"'is' {myStr1} Lower => {res}")

# swapcase()
print("\nSwapCase()\n-----')
res = myStr1.swapcase()
print(f"'is' {myStr1} Uppercase to Lowercase & Vice Versa => {res}")

# title()
print("\nTitle()\n-----')
res = myStr1.title()
print(f"'is' {myStr1} Using title() Titled Word => {res}")

```

	<pre> # capitalize() print("\nCapitalize()\n-----') res = myStr1.strip() res = res.capitalize() print(f"'is' {myStr1} Capitalized =&gt; {res}")  # startswith() myStr1 = myStr1.strip() print("\nStarts With()\n-----') res = myStr1.endswith('!') print(f"Ends With {myStr1} =&gt; {res}") # endswith() print("\nEnds With()\n-----') res = myStr1.startswith('Wat') print(f"Starts with {myStr1} =&gt; {res}") </pre>
<b>Screenshot of Output:</b>	<p>1.</p>  <pre>malayalam is a Palindrome</pre> <p>2.</p>  <pre>This is a paragraph =&gt; ['This', 'is', 'a', 'paragraph'] (0, 'This') (1, 'is') (2, 'a') (3, 'paragraph')</pre> <p>3.</p>  <pre>Hello World</pre> <p>4.</p>

```
Len()
-----
Length of   This is a Paragraph   ! => 26
```

```
Strip()
-----
Strip of This is a Paragraph   ! => This is a Paragraph   !
```

```
lstrip()
-----
Strip of This is a Paragraph   ! => This is a Paragraph   !
```

```
Find()
-----
'This' is in This is a Paragraph   ! => 0
```

```
rFind()
-----
'This' is in This is a Paragraph   ! => 5
```

```
Index()
-----
'is' is at This is a Paragraph   ! idx => 2
```

```
rIndex()
-----
'is' is at This is a Paragraph   ! idx => 5
```

```
rCount()
-----
'is'   This is a Paragraph   ! count is => 2
```

```
Replace()
-----
'is'   This is a Paragraph   ! replaced => Thwas was a Paragraph   !
```

```
Split()
-----
'is'   This is a Paragraph   ! Splited => ['This', 'is', 'a', 'Paragraph', '!']
```

```
Split()
-----
'is'       Watashi no namaiwa Atharv   ! Joined => Watashi#no#namaiwa#Atharv#!
```

```
Upper()
-----
'is'       Watashi no namaiwa Atharv   ! Upper =>   WATASHI NO NAMAIWA ATHARV   !
```

```
Lower()
-----
'is'       Watashi no namaiwa Atharv   ! Lower =>   watashi no namaiwa atharv   !
```

```
SwapCase()
-----
'is'       Watashi no namaiwa Atharv   ! Uppercase to Lowercase & Vice Versa =>   wATASHI NO NAMAIWA aTHARV   !
```

```
Title()
```

	<pre> Title() ----- 'is'    Watashi no namaiwa Atharv  ! Using title() Titled Word =&gt;   Watashi No Namaiwa Atharv   !  Capitalize() ----- 'is'    Watashi no namaiwa Atharv  ! Capitalized =&gt; Watashi no namaiwa atharv   !  Starts With() ----- Ends With Watashi no namaiwa Atharv  ! =&gt; True  Ends With() ----- Starts with  Watashi no namaiwa Atharv  !  =&gt; True </pre>
<b>Observations:</b>	<ol style="list-style-type: none"> <li>1. Strings in Python are immutable, which means they remain unchanged after they are created.</li> <li>2. Python strings are typically enclosed in single ' ' or double " " quotation marks</li> <li>3. Python Str() class has a large collection of built-in string manipulation methods that help with tasks like searching, replacing, splitting, and formatting.</li> </ol>
<b>Conclusion:</b>	<ol style="list-style-type: none"> <li>1. 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.</li> <li>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.</li> <li>3. Categorizing news articles into predefined topics or classes in order to facilitate efficient content organization and retrieval.</li> </ol>

**Subject-In-Charge:**

**Sign:** \_\_\_\_\_

**Prof. Mayura Nagar**