



Vinayakar Matric Hr.Sec.School

STRING
FUNCTIONS

23.08.2021

12 - A - CS - 29

8.9 Built-in String functions

Python supports the following built-in functions to manipulate string.

Syntax	Description	Example
<code>len(str)</code>	Returns the length (no of characters) of the string.	<pre>>>> A="Corporation" >>> print(len(A)) 11</pre>
<code>capitalize()</code>	Used to capitalize the first character of the string	<pre>>>> city="chennai" >>> print(city.capitalize()) Chennai</pre>
<code>center(width, fillchar)</code>	Returns a string with the original string centered to a total of width columns and filled with fillchar in columns that do not have characters	<pre>>>> str1="Welcome" >>> print(str1.center(15,'*')) ****Welcome****</pre>

`find(sub[, start[, end]])`

The function is used to search the first occurrence of the sub string in the given string. It returns the index at which the substring starts. It returns -1 if the substring does not occur in the string.

```
>>>str1='mammals'
```

```
>>>str1.find('ma')
```

0

On omitting the start parameters, the function starts the search from the beginning.

```
>>>str1.find('ma',2)
```

3

```
>>>str1.find('ma',2,4)
```

-1

Displays -1 because the substring could not be found between the index 2 and 4-1.

```
>>>str1.find('ma',2,5)
```

3

Syntax	Description	Example
<code>isalnum()</code>	Returns True if the string contains only letters and digit. It returns False. If the string contains any special character like <code>_</code> , <code>@</code> , <code>#</code> , <code>*</code> , etc.	<pre>>>>str1='Save Earth' >>>str1.isalnum() False</pre> The function returns False as space is an alphanumeric character. <pre>>>>'Save1Earth'.isalnum() True</pre>
<code>isalpha()</code>	Returns True if the string contains only letters. Otherwise return False.	<pre>>>>'Click123'.isalpha() False >>>'python'.isalpha() True</pre>
<code>isdigit()</code>	Returns True if the string contains only numbers. Otherwise it returns False.	<pre>>>> str1='Save Earth' >>>print(str1.isdigit()) False</pre>
<code>lower()</code>	Returns the exact copy of the string with all the letters in lowercase.	<pre>>>>str1='SAVE EARTH' >>>print(str1.lower()) save earth</pre>

<code>islower()</code>	Returns True if the string is in lowercase.	<pre>>>> str1='welcome' >>> print (str1.islower())</pre> <i>True</i>
<code>isupper()</code>	Returns True if the string is in uppercase.	<pre>>>> str1='welcome' >>> print (str1.isupper())</pre> <i>False</i>
<code>upper()</code>	Returns the exact copy of the string with all letters in uppercase.	<pre>>>> str1='welcome' >>> print (str.upper())</pre> <i>WELCOME</i>
<code>title()</code>	Returns a string in title case	<pre>>>> str1='education department' >>> print(str1.title())</pre> <i>Education Department</i>
<code>swapcase()</code>	It will change case of every character to its opposite case vice-versa.	<pre>>>> str1="tAmiL NaDu" >>> print(str1.swapcase())</pre> <i>TaMIL nAdU</i>

Syntax	Description	Example
<code>count(str, beg, end)</code>	Returns the number of substrings occurs within the given range. Remember that substring may be a single character. Range (beg and end) arguments are optional. If it is not given, python searched in whole string. Search is case sensitive.	<pre>>>> str1="Raja Raja Chozhan" >>> print(str1.count('Raja')) 2 >>> print(str1.count('r')) 0 >>> print(str1.count('R')) 2 >>> print(str1.count('a')) 5 >>> print(str1.count('a',0,5)) 2 >>> print(str1.count('a',11)) 1</pre>
<code>ord(char)</code>	Returns the ASCII code of the character.	<pre>>>> ch = 'A' >>> print(ord(ch)) 65 >>> print(ord('B')) 66</pre>

`chr(ASII)`

Returns the character represented by a ASCII.

```
>>> ch=97
>>> print(chr(ch))
a
>>> print(chr(87))
W
```

Example 8.11 Programs using Strings :

Example 8.11.1 : Program to check whether the given string is palindrome or not

```
str1 = input ("Enter a string: ")
str2 = ''
index=-1
for i in str1:
    str2 += str1[index]
    index -= 1
print ("The given string = { } \n The Reversed string = { }".format(str1, str2))
if (str1==str2):
    print ("Hence, the given string is Palindrome")
else:
    print ("Hence, the given is not a palindrome")
```

Output : 1

```
Enter a string: malayalam
The given string = malayalam
The Reversed string = malayalam
Hence, the given string is Palindrome
```

Output : 2

```
Enter a string: welcome
The given string = welcome
The Reversed string = emoclew
Hence, the given string is not a palindrome
```

Example 8.11.2 : Program to display the following pattern

```
*  
* *  
* * *  
* * * *  
* * * * *
```

```
str1='* '
```

```
i=1
```

```
while i<=5:
```

```
    print (str1*i)
```

```
    i+=1
```

Output

```
*  
* *  
* * *  
* * * *  
* * * * *
```


Example 8.11.4 : Program to create an Abecedarian series. (Abecedarian refers list of elements appear in alphabetical order)

```
str1="ABCDEFGH"  
str2="ate"  
for i in str1:  
    print ((i+str2),end='\t')
```

Output

Aate Bate Cate Date Eate Fate Gate Hate



**THANK
YOU!!**