

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>
<code>find(sub[, start[, end]])</code>	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.	<pre>>>>str1='mammals' >>>str1.find('ma') 0 <i>On omitting the start parameters, the function starts the search from the beginning.</i> >>>str1.find('ma',2) 3 >>>str1.find('ma',2,4) -1 <i>Displays -1 because the substring could not be found between the index 2 and 4-1.</i> >>>str1.find('ma',2,5)</pre>
<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 The function returns False as space is an alphanumeric character. >>>'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()) True</pre>
<code>isupper()</code>	Returns True if the string is in uppercase.	<pre>>>> str1='welcome' >>>print (str1.isupper()) False</pre>
<code>upper()</code>	Returns the exact copy of the string with all letters in uppercase.	<pre>>>> str1='welcome' >>>print (str.upper()) WELCOME</pre>
<code>title()</code>	Returns a string in title case	<pre>>>> str1='education department' >>> print(str1.title()) Education Department</pre>
<code>swapcase()</code>	It will change case of every character to its opposite case vice-versa.	<pre>>>> str1="tAmiL NaDu" >>> print(str1.swapcase()) TaMIL nAdU</pre>

<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))</pre>

		65 >>> print(<i>ord('B')</i>) 66
<i>chr(ASII)</i>	Returns the character represented by a ASCII.	>>> ch=97 >>> print(chr(ch)) <i>a</i> >>> print(chr(87)) <i>W</i>