String Functions and Methods

Function/ Method	Description	Example	Output
`len(s)`	Returns length of string *s*	`string1 = 'Python'` `print(len(string1))`	`6`
`max(s)`	Returns the maximum(ASCII) alphabetical character from string *s*	`string1 = 'Python'` `print(max(string1))`	`y`
`min(s)`	Returns the minimum(ASCII) alphabetical character from string *s*	`string1 = 'Python'` `print(min(string1))`	`P`
`s.capitalize()`	Capitalizes first letter of string *s*	`string1 = 'python program'` `print(string1.capitalize())`	`Python program`
`s.lstrip()`	Removes all leading whitespace from string *s*	`string1 = '  Python'` `print(string1.lstrip())`	`Python`
`s.rstrip()`	Removes all trailing whitespace from string *s*	`string1 = 'Python  '` `print(string1.rstrip())`	`Python`
`s.swapcase()`	Inverts case for all letters in string *s*	`string1 = 'Python'` `print(string1.swapcase())`	`pYTHON`
`s.title()`	Returns title cased version of string *s*	`string1 = 'python program'` `print(string1.title())`	`Python Program`
`s.zfill(width)`	Returns string *s* left padded with zeros to a total of width	`string1 = 'python'` `print(string1.zfill(10))`	`0000python`
`s.center(width, fillchar)`	Returns a string padded with fillchar with the original string *s* centered to a total of width	`string1 = 'python'` `print(string1.center(10, '*'))`	`**python**`
`s.ljust(width, fillchar)`	Returns a space padded string with string *s* left-justified to a total of width	`string1 = 'python'` `print(string1.ljust(12, '-'))`	`python------`
`s.rjust(width, fillchar)`	Returns a space padded string with string *s* right justified to a total of width	`string1 = 'python'` `print(string1.rjust(12, '-'))`	`------python`
`s.index(str, beg, end)`	Same as *find()*, but raises an exception if *str* is not found in string *s*	`string1 = 'python'` `print(string1.index('thon'))`	`2`

`s.rfind(str, beg,end)`	Same as the *find()*, but search backwards in string *s*	`string1 = 'python'`
		`print(string1.rfind('thon'))`
		`print(string1.rfind('no'))`
			`2`
			`-1`

`s.rindex(str, beg, end)`	Same as the *index()*, but search backward in string *s*	`string1 = 'python'` `print(string1.rindex('thon'))`	`2`

`s.startswith(prefix,beg, end)`	Returns *True* if a string or substring of string *s* (if starting index *beg* and ending index *end* are given) starts with a prefix and *False* otherwise	`string1 = 'python'`
		`print(string1.startswith('py', 0, 6))`
		`print(string1.startswith('thon', 0, 6))`
			`True`
			`False`

`s.endswith(suffix,beg, end)`	Returns *True* if a string or a substring of string *s* (if starting index *beg* and ending index *end* are given) ends with a suffix and *False* otherwise	`string1 = 'python'`
		`print(string1.endswith('py', 0, 6))`
		`print(string1.endswith('thon', 0, 6))`
			`False`
			`True`

`s.isalnum()`	Returns *True* if string *s* has at	`string1 = 'python'`	`True`

Function/ Method	Description	Example	Output
	least 1 character and all characters are alphanumeric and *False* otherwise	`print(string1.isalnum())`	
`s.isalpha()`	Returns *True* if string *s* has at least 1 character and all characters are alphabetic and *False* otherwise	`string1 = 'python'` `print(string1.isalpha())`	`True`
`s.isdigit()`	Returns *True* if string *s* contains only digits and *False* otherwise	`string1 = 'python'` `print(string1.isdigit())`	`False`
`s.islower()`	Returns *True* if string *s* has at least 1 cased character and all cased characters are in lowercase and *False* otherwise	`string1 = 'python'` `print(string1.islower())`	`True`
`s.isnumeric()`	Returns *True* if unicode string *s* contains only numeric characters and *False* otherwise	`string1 = 'python'` `print(string1.isnumeric())`	`False`
`s.isdecimal()`	Returns *True* if unicode string *s* contains only decimal characters and *False* otherwise	`string1 = 'python'` `print(string1.isdecimal())`	`False`
`s.isspace()`	Returns *True* if string *s* contains only whitespace characters and *False* otherwise	`string1 = 'python'` `print(string1.isspace())`	`False`
`s.istitle()`	Returns *True* if string *s* is properly title cased and *False* otherwise	`string1 = 'python'` `print(string1.istitle())`	`False`
`s.isupper()`	Returns *True* if string *s* has at least one cased character and all cased characters are in uppercase and *False* otherwise	`string1 = 'python'` `print(string1.isupper())`	`False`