Python String Functions

Python has built-in functions for almost every operation that is to be performed on a string. To make it simple, these are classified on the basis of the frequency of their use as well as their operations. They are as follows :

Python String Functions Classification

* Basic Functions
* Advanced Functions
* Miscellaneous (These functions are not specifically for strings but they can be used in string manipulation)

Basic String Functions

|  |  |  |
| --- | --- | --- |
| [capitalize()](https://www.askpython.com/python/string/python-string-capitalize-function) | It converts the first character of a string to uppercase | str\_name.capitalize() |
| casefold() | It converts any string to lower case irrespective of its case | str\_name.casefold() |
| center() | It is used to center align the string | str\_name.center(Length,character) |
| count() | It is used to count the number of times a specific value appears in the string | str\_name.count(value,start,end) |
| endswith() | It checks if the string is ending with specified value then it returns True | str\_name.endswith(value,start,end) |
| find() | It is used to find the presence of a specified value in a string | str\_name.find(value,start,end) |
| index() | It is used to find the first occurrence of a specified value in the string | str\_name.index(value,start,end) |
| isalnum() | It checks if all the characters are alphanumeric then returns True | str\_name.isalnum() |
| isalpha() | It checks if all the characters are alphabet (a-z) then returns True | str\_name.isalpha() |
| isdecimal() | It checks if all the characters are decimals (0-9) then returns True | str\_name.isdecimal() |
| isdigit() | It checks if all the characters are digits then returns True | str\_name.isdigit() |
| islower() | It checks if all the characters are in lowercase then returns True | str\_name.islower() |
| isnumeric() | It checks if all the characters are numeric (0-9) then returns True | str\_name.isnumeric() |
| isspace() | It checks if all the characters are whitespaces then returns True | str\_name.isspace() |
| isupper() | It checks if all the characters are in uppercase then returns True | str\_name.isupper() |
| lower() | It is used to convert all characters to lowercase | str\_name.lower() |
| partition() | It is used to split the string into a tuple of three elements | str\_name.partition(value) |
| replace() | It is used to replace specified word or phrase into another word or phrase in the string | str\_name.replace(oldvalue,newvalue,count) |
| split() | It is used to split a string into a list | str\_name.split(separator,maxsplit) |
| splitlines() | It is used to split the string and make a list of it. Splits at the line breaks. | str\_name.splitlines(keeplinebreaks) |
| startswith() | It checks if the string is starting with specified value then it returns True | str\_name.startswith(value,start,end) |
| strip() | It is used to remove characters specified in argument from both the ends | str\_name.strip(characters) |
| swapcase() | It is used to swap uppercase string to lowercase or vice versa | str\_name.swapcase() |
| title() | It converts initial letter of each word to uppercase | str\_name.title() |
| upper() | It is used to convert all characters in a string to uppercase | str\_name.upper() |

Advanced Python String Functions

|  |  |  |
| --- | --- | --- |
| encode() | It is used to return encoded strings | str\_name.encode(encoding=*encoding*, errors=*errors*) |
| expandtabs() | It is used to set or fix tab spaces between characters or alphabets | str\_name.expandtabs(tabsize) |
| format() | It replaces variable name written within {} with the value at execution | str\_name.format(value1,value2...) |
| format\_map() | It is used to format a string given and is returned | str\_name.format\_map(mapping) |
| isidentifier() | It checks if characters are alphanumeric letters (a-z) and (0-9), or underscores (\_) and return True | str\_name.isidentifier() |
| isprintable() | It checks if all characters are printable then returns True | str\_name.isprintable() |
| istitle() | It checks if all initial characters of words are in uppercase then returns True | str\_name.istitle() |
| join() | It accepts words as iterable and joins them into a string | str\_name.join(iterable) |
| ljust() | It returns a left-aligned string with the minimum value given as width | str\_name.ljust(length,character) |
| lstrip() | It removes characters from left end based on the given argument | str\_name.lstrip(characters) |
| maketrans() | It creates a mapped table usable for translations | str\_name.maketrans(x,y,z) |
| rsplit() | It is used to split the string from the right end | str\_name.rsplit(separator,maxsplit) |
| rfind() | It searches for a specified value and finds the position of its last value | str\_name.rfind(value,start,end) |
| rindex() | It searches for a specified value and finds the position of its last value | str\_name.rindex(value,start,end) |
| rjust() | It returns a right-aligned string with the minimum value given as width | str\_name.rjust(length,character) |
| rpartition() | It looks for the last occurrence of a specified string and splits the string into the tuple of three elements | str\_name.rpartition(value) |
| rstrip() | It removes characters from the right end based on the given argument | str\_name.rstrip(characters) |
| translate() | It is used to get a translated string | str\_name.translate(table) |
| zfill() | It returns a new string with ‘0’ characters padded to the left on the string | str\_name.zfill(len) |

Miscellaneous Functions that work on String

|  |  |  |
| --- | --- | --- |
| ascii() | It returns a string containing the printable form of an object and ignores the non-ASCII values in the string | ascii(object) |
| bool() | It returns boolean value i.e. True or False for an object | bool(value) |
| bytearray() | It returns an object containing an array of bytes provided through the input | bytearray(source,encoding,errors) |
| bytes() | It returns bytes object which cannot be modified and is a sequence of integers in the range from 0 to 255 | bytes(source,encoding,errors) |
| enumerate() | It is used to add a counter to an iterable and then returns its value | enumerate(iterable,start=0) |
| float() | It returns floating-point number from the given argument | float(argument) |
| hash() | It returns the hash value of the object, if applicable | hash(object) |
| id() | It returns the specific identity of an object which is a unique integer | id(object) |
| int() | It returns an integer object from the given input and the base of the returned object will always be 10 | int(x=0,base=10) |
| len() | It returns the length of sequence i.e. number of items in an object | len(sequence) |
| map() | It is used to apply a given function to every item of iterable which can be a tuple, list, etc. and also returns a list containing resultant values | map(function, iterable, ...) |
| ord() | It accepts a string argument of single Unicode character and returns its respect Unicode point | ord(character) |
| print() | It prints the provided object to any output device | print*(object(s)*,separator=*separator*, end=*end*,file=*file*,flush=*flush*) |
| slice() | It creates an object which represents a set of indices specified by its range(start, stop, step) | slice(stop) slice(start,stop,step) |
| type() | It returns the object’s type | type(object) type(name,bases,dict) |