
stringutils

Release 0.3.0

Feb 04, 2018

Contents

1	Features	3
2	Installation	5
3	Usage	7
4	Contribute	9
5	License	11
5.1	Function reference	11

A functional string utility library for Python 2 and 3. Closely inspired by implementations in Haskell and PHP.
For more documentation, please see <http://stringutils.readthedocs.io>.

Note: This package is still in an early development stage, and it is possible that the naming and API of some functions will change.

CHAPTER 1

Features

The main design goal of this library is to provide helpful string functions which complement what can already be done with *str*, *str:format* and *textwrap*. As such, it includes:

- A selective set of additional string helper functions.
- Extended versions of existing *str* methods as functions where useful.

CHAPTER 2

Installation

Install the latest release from [PyPI](#):

```
pip install stringutils
```


CHAPTER 3

Usage

All functions are available directly off the `stringutils` package. You may choose to import individual functions by name, or import all.

```
from stringutils import reverse, unwords, words

def reverse_words(string):
    return unwords(map(reverse, words(string)))
```


CHAPTER 4

Contribute

- Source code: <https://github.com/huntie/stringutils>
- Issue tracker: <https://github.com/huntie/stringutils/issues>

The project is licensed under the MIT license.

5.1 Function reference

5.1.1 Assertions

`stringutils.contains (string, matches)`

Determine if a string contains any of the given values. *matches* may be a single string, or a list of strings.

`stringutils.contains_all (string, matches)`

Determine if a string contains all of the given values.

`stringutils.is_whitespace (string)`

Determine if a string contains only whitespace characters or is empty.

5.1.2 Case conversion

All functions which convert to programming case formats accept strings in any of the other formats.

`stringutils.camel_case (string)`

Convert a string identifier to `camelCase`.

`stringutils.dashed_case (string)`

Convert a string identifier to dashed-case. If the string is in `snake_case`, capitalization of words will be preserved.

`stringutils.lcfirst (string)`

Convert the first character of a string to lowercase.

`stringutils.pascal_case (string)`

Convert a string identifier to `PascalCase`.

`stringutils.snake_case(string)`

Convert a string identifier to `snake_case`. If the string is in `dashed-case`, capitalization of words will be preserved.

`stringutils.title_case(string)`

Convert a string identifier to `Title Case`.

`stringutils.ucfirst(string)`

Convert the first character of a string to uppercase.

5.1.3 Lists

`stringutils.concat(strings)`

Concatenate a list of strings into a single string.

`stringutils.join(strings, sep=', ', insertend=False)`

Concatenate a list of strings into a single string by a separating delimiter. If *insertend* is given and true, the delimiter is also included at the end of the string.

`stringutils.lines(string, keepends=False)`

Split a string into a list of strings at newline characters. Unless *keepends* is given and true, the resulting strings do not have newlines included.

`stringutils.split_identifier(string)`

Split a string identifier into a list of its subparts.

`stringutils.words(string)`

Split a string into a list of words, which were delimited by one or more whitespace characters.

`stringutils.unlines(lines, newline='\n')`

Join a list of lines into a single string after appending a terminating newline character to each.

`stringutils.unwords(words)`

Join a list of words into a single string with separating spaces.

5.1.4 Transformations

`stringutils.reverse(string)`

Reverse the order of the characters in a string.

C

`camel_case()` (in module `stringutils`), 11
`concat()` (in module `stringutils`), 12
`contains()` (in module `stringutils`), 11
`contains_all()` (in module `stringutils`), 11

D

`dashed_case()` (in module `stringutils`), 11

I

`is_whitespace()` (in module `stringutils`), 11

J

`join()` (in module `stringutils`), 12

L

`lcfirst()` (in module `stringutils`), 11
`lines()` (in module `stringutils`), 12

P

`pascal_case()` (in module `stringutils`), 11

R

`reverse()` (in module `stringutils`), 12

S

`snake_case()` (in module `stringutils`), 11
`split_identifier()` (in module `stringutils`), 12

T

`title_case()` (in module `stringutils`), 12

U

`ucfirst()` (in module `stringutils`), 12
`unlines()` (in module `stringutils`), 12
`unwords()` (in module `stringutils`), 12

W

`words()` (in module `stringutils`), 12