

How to convert string to Title Case in Python?

Asked 12 years, 4 months ago Modified 2 years ago Viewed 248k times



Example:

164



```
HILO -> Hilo  
new york -> New York  
SAN FRANCISCO -> San Francisco
```



Is there a library or standard way to perform this task?



python

Share Improve this question

Follow

edited Apr 15, 2019 at 20:25



Jonathan Benn

3,230 ● 4 ● 24 ● 30

asked Dec 1, 2011 at 19:21



daydreamer

89.4k ● 198 ● 461 ● 735

19 That isn't "CamelCase", that's "Capitalize"; which do you want? – Andrew Marshall Dec 1, 2011 at 19:23

14 camelCase isLike this. – Jonathan M Dec 1, 2011 at 19:27

6 Your examples use PascalCase. – David Betz Feb 10, 2016 at 20:53

I used this one and it worked pretty well. stackoverflow.com/a/74912560/17598016 – Procodedev Feb 25, 2023 at 13:55

10 Answers

Sorted by: Highest score (default)



Why not use [title](#) Right from the docs:

354



```
>>> "they're bill's friends from the UK".title()  
"They'Re Bill'S Friends From The Uk"
```

If you really wanted PascalCase you can use this:



```
>>> ''.join(x for x in 'make IT pascal CaSe'.title() if not x.isspace())  
'MakeItPascalCase'
```



Share Improve this answer Follow

edited Sep 13, 2019 at 17:32



Aidan H

122 ● 9

answered Dec 1, 2011 at 19:33



Facundo Casco

10.3k ● 8 ● 43 ● 64

- 7 I think the 'r' in "They're" ought to be lower case. And the 's' in "Bill's" definitely must be lower case. – [Daniel Fischer](#) Dec 1, 2011 at 19:55
- 4 @Daniel - That issue is noted in the docs for `title`: "The algorithm uses a simple language-independent definition of a word as groups of consecutive letters. The definition works in many contexts but it means that apostrophes in contractions and possessives form word boundaries, which may not be the desired result". One possible solution would be to use Laurence's answer with the regex `r"['\w]+"` so apostrophes wouldn't end a match (additional punctuation could be added as needed). – [Andrew Clark](#) Dec 1, 2011 at 20:10
- 27 For the record, a neater way to do the last CamelCase example is as `'make IT camel CaSe'.title().replace(' ', '')`. – [Henry Gomersall](#) Oct 7, 2013 at 12:46
- 24 If anyone else feels like they're on crazy pills -- this is PascalCase, not camelCase. – [bertday](#) Mar 15, 2016 at 17:40
- 6 Nice code but camelCase doesn't start with a CAPITAL. Try this:

```
def toCamel(s):    ret =
''.join(x for x in s.title() if not x.isspace())    return ret[0].lower() +
ret[1:]
```

 Usage: `toCamel("WRITE this in camelcase")` `'writeThisInCamelcase'` – [Ron Kalian](#) Nov 22, 2017 at 11:46



This one would always start with lowercase, and also strip non alphanumeric characters:

34

```
def camelCase(st):
    output = ''.join(x for x in st.title() if x.isalnum())
    return output[0].lower() + output[1:]
```



Share Improve this answer Follow edited Jun 28, 2018 at 10:32



answered Dec 23, 2013 at 13:51



[Ivan Chaer](#)

7,020 ● 1 ● 39 ● 49



```
def capitalizeWords(s):
    return re.sub(r'\w+', lambda m:m.group(0).capitalize(), s)
```

15



`re.sub` can take a function for the "replacement" (rather than just a string, which is the usage most people seem to be familiar with). This `repl` function will be called with an `re.Match` object for each match of the pattern, and the result (which should be a string) will be used as a replacement for that match.



A longer version of the same thing:

```
WORD_RE = re.compile(r'\w+')

def capitalizeMatch(m):
    return m.group(0).capitalize()

def capitalizeWords(s):
    return WORD_RE.sub(capitalizeMatch, s)
```

This pre-compiles the pattern (generally considered good form) and uses a named function instead of a lambda.

Share Improve this answer Follow

edited Dec 1, 2011 at 22:37

answered Dec 1, 2011 at 19:29



Laurence Gonsalves

140k ● 35 ● 252 ● 303

this one is pretty neat, I am trying to get my head around lambda functions, thank you for helping

– [daydreamer](#) Dec 1, 2011 at 19:36

- 1 @JohnMachin I just asked because I thought adding some explanation would make your answer more complete and better. – [N.N.](#) Dec 1, 2011 at 21:35

@Laurence Gonsalves what is lambda doing here? – [Zion](#) Aug 17, 2015 at 21:07 ✎

what is lambda doing here? from wha I can decipher and from you explanation. this is what I understood. when you use a function in re.sub each `match` will be passed to the function? and since `matches` in regular expressions have groups. thats why this line exists `Lambda m:m.group(0).capitalize()` ? – [Zion](#) Aug 17, 2015 at 21:21

@Zion yes. When `re.sub` is given a callable (eg: a function) as the "replacement", it passes the match object to that callable, and expects to get back a string which is what it actually uses as the replacement. If you find lambdas confusing, the "longer version" does exactly the same thing in a more verbose way. – [Laurence Gonsalves](#) Aug 18, 2015 at 18:37



Potential library: <https://pypi.org/project/stringcase/>

8

Example:



```
import stringcase
stringcase.camelcase('foo_bar_baz') # => "fooBarBaz"
```



Though it's questionable whether it will leave spaces in. (Examples show it removing space, but there is a bug tracker issue noting that it leaves them in.)

Share Improve this answer Follow

edited Oct 17, 2018 at 16:26

answered Oct 16, 2018 at 21:24



gdw2

7,788 ● 4 ● 46 ● 49



Lol

950 ● 8 ● 6

- 1 Heck yes. Was looking for a package. This package has snakecase and other conversion functions as well. – [s2t2](#) Jul 19, 2019 at 0:20



just use `.title()`, and it will convert first letter of every word in capital, rest in small:

7



```
>>> a='mohs shahid ss'
>>> a.title()
'Mohs Shahid Ss'
>>> a='TRUE'
>>> b=a.title()
>>> b
```





```
'True'
>>> eval(b)
True
```

Share Improve this answer Follow

answered Feb 24, 2016 at 5:15



Mohammad Shahid Siddiqui

3,960 ● 2 ● 27 ● 13



Why not write one? Something like this may satisfy your requirements:

6

```
def FixCase(st):
    return ' '.join(' '.join([w[0].upper(), w[1:].lower()]) for w in st.split())
```



Share Improve this answer Follow

answered Dec 1, 2011 at 19:25



multipleinterfaces

9,013 ● 4 ● 31 ● 34



thanks, that helped completely. My bad, I didn't think to write one in first place – [daydreamer](#) Dec 1, 2011 at 19:31



6

Note: Why am I providing yet another answer? *This answer is based on the title of the question and the notion that camelcase is defined as: a series of words that have been concatenated (no spaces!) such that each of the original words start with a capital letter (the rest being lowercase) excepting the first word of the series (which is completely lowercase). Also it is assumed that "all strings" refers to ASCII character set; unicode would not work with this solution).*



simple

Given the above definition, this function

```
import re
word_regex_pattern = re.compile("[^A-Za-z]+")

def camel(chars):
    words = word_regex_pattern.split(chars)
    return " ".join(w.lower() if i is 0 else w.title() for i, w in enumerate(words))
```

, when called, would result in this manner

```
camel("San Francisco") # sanFrancisco
camel("SAN-FRANCISCO") # sanFrancisco
camel("san_francisco") # sanFrancisco
```

less simple

Note that it fails when presented with an already camel cased string!

```
camel("sanFrancisco") # sanfrancisco <-- noted limitation
```

even less simple

Note that it fails with many unicode strings

```
camel("México City") # mXicoCity <-- can't handle unicode
```

I don't have a solution for these cases(or other ones that could be introduced with some creativity). So, as in all things that have to do with strings, cover your own edge cases and good luck with unicode!

Share Improve this answer Follow

edited Jul 28, 2017 at 6:47

answered Nov 21, 2016 at 22:01



Patrice Bernassola
14.3k ● 6 ● 48 ● 60



Marc
5,170 ● 3 ● 42 ● 38

How can you determine that a string is Camel case without knowing the meaning of the sentence? In your "less simple" example, "sanfRancisco" is Camel case as well as "itSnotcaMelcAse".

– Patrice Bernassola Jul 25, 2017 at 15:42

I'm guessing your input had apostrophes or other punctuation? I should note other failed input. Definitely a good catch. What was the input you provided? – Marc Jul 25, 2017 at 17:30

- 1 I mean than a sequence of characters without space must be consider as 1 word. You can not extract work from it without knowing the meaning of the sentence. Put "sanfRancisco" or "itSnotcaMelcAse" as input of camello() and you will see that the output will be the same. – Patrice Bernassola Jul 27, 2017 at 12:44

Oh I see - yeah, I think you're right. I overfit the solution. I'll update it. – Marc Jul 27, 2017 at 16:34



2



Share Improve this answer Follow

answered May 18, 2020 at 9:56



Aishwarya Skandamani
191 ● 1 ● 5



I would like to add my little contribution to this post:

0



```
def to_camelcase(str):  
    return ' '.join([t.title() for t in str.split()])
```

Share Improve this answer Follow

answered Mar 12, 2019 at 19:07



Evhz

9,028 ● 9 ● 52 ● 70

Actually, str.title() is the same and you are saving up computational cost. – [JuanB](#) May 4, 2020 at 11:04



0



From code wars - Write simple .camelCase method in Python for strings. All words must have their first letter capitalized without spaces. camelcase("hello case") => HelloCase
camelcase("camel case word") => CamelCaseWord

```
def camel_case(string):  
    titled_string = string.title()  
    space_joined_string = titled_string.replace(' ', '')  
    return space_joined_string
```

Share Improve this answer Follow

answered Mar 30, 2022 at 5:46



Dancan Chibole

151 ● 1 ● 4