

A string is considered to be in title case if each word in the string is either (a) capitalised (that is, only the first letter of the word is in upper case) or (b) considered to be an exception and put entirely into lower case unless it is the first word, which is always capitalised.

Write a function that will convert a string into title case, given an optional list of exceptions (minor words). The list of minor words will be given as a string with each word separated by a space. Your function should ignore the case of the minor words string -- it should behave in the same way even if the case of the minor word string is changed.

Arguments (Haskell)

- **First argument:** space-delimited list of minor words that must always be lowercase except for the first word in the string.
- **Second argument:** the original string to be converted.

Arguments (Other languages)

- **First argument (required):** the original string to be converted.
- **Second argument (optional):** space-delimited list of minor words that must always be lowercase except for the first word in the string. The JavaScript/CoffeeScript tests will pass `undefined` when this argument is unused.

Example

```
title_case('a clash of KINGS', 'a an the of') # should return: 'A Clash of Kings'
title_case('THE WIND IN THE WILLOWS', 'The In') # should return: 'The Wind in the Willows'
title_case('the quick brown fox') # should return: 'The Quick Brown Fox'
```

My Solution:

```
def title_case(title, minor_words=''):
    new_title = [title.split(" ")[0].title()]
    minor_words = [value.lower() for value in minor_words.split(" ")]
```

```
for word in title.split(" ")[1:]:
    if word.lower() in minor_words:
        new_title.append(word.lower())
    else:
        new_title.append(word.title())

return " ".join([value for value in new_title])
```