Ch.12_Ex_8_wordtools

December 3, 2020

0.1 Ch. 12

```
8. Create a module named wordtools.py with our test scaffolding in place.

Now add functions to these tests pass:

test(cleanword("what?") == "what")
test(cleanword("'now!'") == "now")
test(cleanword("?+='w-o-r-d!,@$()'") == "word")

test(has_dashdash("distance--but"))
test(not has_dashdash("several"))
test(has_dashdash("spoke--"))
test(has_dashdash("distance--but"))
test(not has_dashdash("-yo-yo-"))

test(extract_words("Now is the time! 'Now', is the time? Yes, now.") ==
['now','is','the','time','now','is','the','time','yes','now'])
test(extract_words("she tried to curtsey as she spoke--fancy") ==
['she','tried','to','curtsey','as','she','spoke','fancy'])

test(wordcount("now", ["now","is","time","is","now","is","is"]) == 2)
test(wordcount("is", ["now","is","time","is","now","the","is"]) == 3)
test(wordcount("time", ["now","is","time","is","now","is","is"]) == 1)
test(wordcount("frog", ["now","is","time","is","now","is","is"]) == 0)
test(wordset(["now", "is", "time", "is", "now", "is", "is"]) ==
["is", "now", "time"])

test(wordset(["I", "a", "a", "is", "a", "is", "I", "am"]) ==
["I", "a", "am", "is"])
test(wordset(["or", "a", "am", "is", "are", "be", "but", "am"]) ==
["a", "am", "are", "be", "but", "is", "or"])

test(longestword(["a", "apple", "pear", "grape"]) == 5)
test(longestword(["a", "am", "I", "be"]) == 2)
test(longestword(["this","supercalifragilisticexpialidocious"]) == 34)
test(longestword([ ]) == 0)
```

Save this module so you can use the tools it contains in future programs.

[2]:
```
from unit_tester import test
import string
```

Function 1

[3]:
```
def cleanword(s):
    s_without_punct = ""
    for letter in s:
        if letter not in string.punctuation:   #the assignment is automatic␣
↪from an already created module
            s_without_punct += letter #creates a string
    split = s_without_punct.split()
    return s_without_punct
```

Single Example

[6]:
```
print(cleanword("?+='w-o-r-d!,@$()'"))
```

```
word
```

Tests

[5]:
```
test(cleanword("what?") == "what")
test(cleanword("'now!'") == "now")
test(cleanword("?+='w-o-r-d!,@$()'") == "word")
```

```
Test at line 1 ok.
Test at line 2 ok.
Test at line 3 ok.
```

Function 2 A program that determines if a string has a dash ("-") in it.

[7]:
```
def has_dashdash(text):
    return "--" in text
```

[8]:
```
test(has_dashdash("distance--but"))
test(not has_dashdash("several"))
test(has_dashdash("spoke--"))
test(has_dashdash("distance--but"))
test(not has_dashdash("-yo-yo-"))
```

```
Test at line 1 ok.
Test at line 2 ok.
Test at line 3 ok.
Test at line 4 ok.
Test at line 5 ok.
```

Function 3 A program that extracts the words from a text.

[11]:
```
def extract_words(s):
    result = s.split("--")
    result = " ".join(result)
    result = result.lower()
    result = cleanword(result)
    result = result.split(" ")
    return result
```

[12]:
```
test(extract_words("Now is the time! 'Now', is the time? Yes, now.") ==␣
↪['now','is','the','time','now','is','the','time','yes','now'])
test(extract_words("she tried to curtsey as she spoke--fancy") ==␣
↪['she','tried','to','curtsey','as','she','spoke','fancy'])
```

```
Test at line 1 ok.
Test at line 2 ok.
```

Function 4 Counts the occurance of a specific word in a text

[15]:
```
def wordcount(word, list):
    count = 0
    for s in list:
        if word == s:
            count += 1
    print(count)
    return count
```

[14]:
```
test(wordcount("now", ["now","is","time","is","now","is","is"]) == 2)
test(wordcount("is", ["now","is","time","is","now","the","is"]) == 3)
test(wordcount("time", ["now","is","time","is","now","is","is"]) == 1)
test(wordcount("frog", ["now","is","time","is","now","is","is"]) == 0)
```

```
2
Test at line 1 ok.
3
Test at line 2 ok.
1
Test at line 3 ok.
0
Test at line 4 ok.
```

Function 5 Sorts the list of words and remove duplicates

[16]:
```
def wordset(list):
    new_list = []
    for i in sorted(list):
        new_elem = i
```

```
        if new_elem not in new_list:
            new_list.append(new_elem)
    print(new_list)
    return new_list
```

[17]:
```
test(wordset(["now", "is", "time", "is", "now", "is", "is"]) == ["is", "now",␣
↪"time"])
test(wordset(["I", "a", "a", "is", "a", "is", "I", "am"]) == ["I", "a", "am",␣
↪"is"])
test(wordset(["or", "a", "am", "is", "are", "be", "but", "am"]) == ["a", "am",␣
↪"are", "be", "but", "is", "or"])
```

```
['is', 'now', 'time']
Test at line 1 ok.
['I', 'a', 'am', 'is']
Test at line 2 ok.
['a', 'am', 'are', 'be', 'but', 'is', 'or']
Test at line 3 ok.
```

Function 6 Returns the count of letters in the longest word

[18]:
```
def longestword(list):
    greatest = 0
    for i in list:
        if len(i) >= greatest:
            greatest = len(i)
    print(greatest)
    return greatest
```

[19]:
```
test(longestword(["grape", "apple", "pear", "a"]) == 5)
test(longestword(["a", "am", "I", "be"]) == 2)
test(longestword(["this","supercalifragilisticexpialidocious"]) == 34)
test(longestword([ ]) == 0)
```

```
5
Test at line 1 ok.
2
Test at line 2 ok.
34
Test at line 3 ok.
0
Test at line 4 ok.
```

[ ]: