lecture12_practice

February 19, 2024

1 Practice: Dictionary Counters

Now it's your turn to practice dictionary counters

1. Copy the code from the demo that counts letters in a string. Modify the string to be something else and find the letter_counts (no need to sort)

[7]:

```python
letter_counts = {}

# go through each letter in the word
for letter in "Dinosaurs":
    if letter not in letter_counts: # If there is no entry for this letter yet
        letter_counts[letter] = 1   # then make an entry set to 1
    else: # otherwise, there was an entry
        letter_counts[letter] += 1  # so add one to that entry

# we now have the total number of letters
display("total letter counts are: ")
display(letter_counts)
```

```
'total letter counts are: '
```

```
{'D': 1, 'i': 1, 'n': 1, 'o': 1, 's': 2, 'a': 1, 'u': 1, 'r': 1}
```

Now let's try this with words.

The code below makes a string, and then splits it into words by dividing it apart at each space.

[8]:

```python
# Save a poem into a string (we can use """ to make a multiline string)
# Fire and Ice BY ROBERT FROST (now public domain)
poem = """Some say the world will end in fire,
Some say in ice.
From what I've tasted of desire
I hold with those who favor fire.
But if it had to perish twice,
I think I know enough of hate
To say that for destruction ice
Is also great
And would suffice."""
```

```
# split the string (all lowercase) into words
import re # import the Regular Expressions library, to help us split words

#make the poem all lowercase
lower_case_poem = poem.lower()

# split the poem into pieces at all spaces and newlines (\s), and ,'s and .'s
poem_split_by_spaces_and_punctuation = re.split(("[\s,.]"), lower_case_poem)

# get rid of some empty strings "" that ended up in our list
split_poem = list(filter(None, poem_split_by_spaces_and_punctuation))

print(split_poem)
```

```
['some', 'say', 'the', 'world', 'will', 'end', 'in', 'fire', 'some', 'say',
'in', 'ice', 'from', 'what', 'i’ve', 'tasted', 'of', 'desire', 'i', 'hold',
'with', 'those', 'who', 'favor', 'fire', 'but', 'if', 'it', 'had', 'to',
'perish', 'twice', 'i', 'think', 'i', 'know', 'enough', 'of', 'hate', 'to',
'say', 'that', 'for', 'destruction', 'ice', 'is', 'also', 'great', 'and',
'would', 'suffice']
```

2. Make code that counts how often each word appears in the poem (it should be very similar to the code from problem 1 above

[9]:

```
word_counts = {}

for word in split_poem:
    if word not in word_counts:
        word_counts[word] = 1
    else:
        word_counts[word] += 1
display("total word counts are: ")
display(word_counts)
```

```
'total word counts are: '
```

```
{'some': 2,
 'say': 3,
 'the': 1,
 'world': 1,
 'will': 1,
 'end': 1,
 'in': 2,
 'fire': 2,
 'ice': 2,
 'from': 1,
 'what': 1,
 'i’ve': 1,
 'tasted': 1,
 'of': 2,
 'desire': 1,
 'i': 3,
 'hold': 1,
 'with': 1,
 'those': 1,
 'who': 1,
 'favor': 1,
 'but': 1,
 'if': 1,
 'it': 1,
 'had': 1,
 'to': 2,
 'perish': 1,
 'twice': 1,
 'think': 1,
 'know': 1,
 'enough': 1,
 'hate': 1,
 'that': 1,
 'for': 1,
 'destruction': 1,
 'is': 1,
 'also': 1,
 'great': 1,
 'and': 1,
 'would': 1,
 'suffice': 1}
```