Assignment 5

March 3, 2020

1 Assignment 7

1.1 Problem 1

Read the entire text of the file “The Raven.tx” and create a list, words_list, of the words in the file using the string method split() with the default split character. Print the number of words in the list using len().

[16]:

```
words_list = []
with open('The Raven.txt') as TheRaven:
    for line in TheRaven:
        #print(line)
        splitLine = line.split()
        for word in splitLine:
            words_list.append(word)
    #print(words_list)
    print(len(words_list))
```

```
1067
```

1.2 Problem 2

Use the set function to create a set of all of the words in the list and name the set “Universe”.

[41]:

```
Universe = set()
for word in words_list:
    Universe.add(word)
#print(Universe)
#print(len(Universe))
```

1.3 Problem 3

Using set comprehension produce a set called capwords consisting of all the words in the Universe which begin with capital letters. Use re.search() and an appropriate regex. Don’t forget to import re.

Print the set and a count of the items in the set.

[40]:

```
import re
def beginCapital(text):
    # regex
    pattern = '[A-Z]+[a-z]+$'
    if re.search(pattern, text):
        return text
        #print(text)

capwords = {word for word in Universe if beginCapital(word)}
print(capwords)
print()
print(len(capwords))
```

```
{'Back', 'Much', 'Not', 'But', 'Tell', '’Tis', '"Be', 'By', 'Soon', 'Ghastly',
'What', 'Swung', 'On', 'Then', 'Hopes', 'Heaven', 'This', 'Open', 'Plutonian',
'Nothing', 'With', 'Nameless', 'In', 'Seraphim', 'For', 'Till', 'So', 'It',
'Nightly', 'Eagerly', 'Is', 'Though', '"Other', '"Though', 'Ever', 'To',
'Thrilled', 'Hope', 'Let', 'Only', 'Straight', 'Of', 'God', 'From', 'Deep',
'She', '"’Tis', 'Caught', 'That', 'Whether', 'Over', 'Some', 'Perched', 'While',
'Take', 'And', 'Once', 'Startled', 'Leave', 'Bird', 'Meant', 'Darkness', 'As',
'Disaster', 'Tempter', 'Quoth', 'Desolate', '"Get', 'Clasp', 'Pallas',
'Presently', 'Shall', 'Merely', 'Horror', 'Followed', 'Fancy', 'Raven'}

77
```

1.4 Problem 4

Using set comprehension, create sets ewords and awords. An eword contains a lower case “e” or an uppercase “E”. The set awords has an obvious analogous definition. Print these sets and tbeir counts. Note that the membership conditions can the “in” operator. A regular expression is not necessary.

[39]:

```
ewords = {word for word in Universe if "E" in word or "e" in word}
awords = {word for word in Universe if "A" in word or "a" in word}
print(ewords)
print()
print(len(ewords))
print()
print()
print(awords)
print()
print(len(awords))
```

```
{'core;', 'more.', 'Respite-respite', 'above', 'friends', 'little', 'scarcely',
'Aidenn,', 'Tell', 'laden', 'before;', 'censer', 'then,', 'wheeled', 'the',
'bore;', 'beak', 'burned', 'heart', 'see,', 'Seraphim', 'Lenore-', 'farther',
'fiend!"', 'wide', 'nodded,', 'craven,', '"Lenore!"-', 'forget', 'stern',
'made', 'somewhat', 'spoke', 'shore!"', 'yore-', 'louder', 'leave', 'feather',
'ebony', 'Then,', 'pondered,', 'beating', 'thee', 'thee-by', 'streaming',
'lent', 'shutter,', 'perfumed', 'She', 'yore', 'velvet-violet', 'Over',
'minute', 'dreaming', 'Whether', 'murmured', 'Once', 'implore-', 'Startled',
'Leave', 'rare', 'Tempter', 'forgiveness', 'sure', 'sent,', 'you"-here', 'sent',
'heard', 'press,', 'then', 'demon’s', 'myself', 'there-is', 'perched',
'discourse', 'relevancy', 'explore-', 'have', 'token,', 'unseen', 'late',
'here', 'reclining', 'more', 'followed', 'grew', '"Be', 'denser,', 'shore-',
'whispered,', 'flutter,', 'home', 'shrieked,', 'purple', 'he;', 'cushioned',
'crest', 'Heaven', 'Open', 'engaged', 'when,', 'being', 'tinkled', 'gently',
'explore;-', 'lonely', 'scarce', 'o’er', 'forgotten', 'store', 'wished',
'"Other', 'countenance', 'fiery', 'unbroken!-quit', 'mien', 'stopped',
'surcease', 'stillness', 'expressing', 'hesitating', 'Perched,', 'angels',
'stepped', 'silence', 'yet', 'one', 'reply', 'unmerciful', 'shore!', 'there',
'nevermore!', 'lies', 'Take', 'suddenly', 'velvet', 'ancient', 'beguiling',
'these', 'hear', 'least', 'muttered', 'before-', 'plume', 'memories', 'answer',
'implore;', 'sculptured', 'uttered-not', '"surely', 'Merely', 'wore,', 'token',
'undaunted,', 'guessing,', '"Doubtless,"', 'melancholy', 'take', 'tossed',
'evil!-prophet', 'weak', 'silken,', 'gloated', 'Then', 'darkness', 'loneliness',
'more."', 'ember', 'methought,', 'stronger;', 'bends', 'decorum', 'bore',
'stately', 'he', 'dreaming,', 'name', 'peering,', 'eyes', 'Hope', 'chamber',
'‘Never-nevermore’."', 'lattice;', 'Deep', 'while', 'seeming', 'came', 'be',
'wondering,', 'repeating', 'Some', 'devil!', 'While', 'lifted-nevermore!',
'spoken,', 'wandering', 'dreams', 'something', 'Meant', 'Disaster', 'maiden',
'Lenore;', 'before."', 'yore;', 'devil!-', 'beast', 'master', 'o’er,', 'seat',
'more!"', 'whether', 'Presently', 'whispered', 'muttered,', 'weary,', 'me,',
'"Wretch,"', 'entreating', 'burden', 'desert', 'dream', 'obeisance', 'broken',
'me-filled', 'sainted', 'never', 'blessed', 'each', 'head', 'gave', 'faster',
'Hopes', 'Gilead?-tell', 'dirges', 'me-tell', 'spoken', 'cried,', 'ashore,',
'echo', 'dared', 'Nameless', 'spoken!', 'dreary,', '"Lenore?"', 'we', 'some',
'seeing', 'nearly', 'syllable', 'shaven,', 'grave', 'terrors', '"Surely,"',
'me', 'Lenore."', 'meaning-little', 'Eagerly', 'uncertain', 'implore!"', 'Ever',
'Thrilled', 'unbroken,', 'Let', 'before.', 'tempest', 'whose', 'help',
'remember', 'fearing,', 'bore-', 'opened', 'enchanted-', 'adore-', 'December;',
'marvelled', 'utters', 'thereat', 'heart,', 'Perched', 'mystery', 'fluttered-',
'lore-', 'tufted', '"Nevermore."', 'betook', 'entrance', 'Darkness',
'haunted-tell', '"Prophet!"', 'Desolate', 'felt', 'Raven,', '"Get', 'separate',
'stayed', 'bleak', 'Lenore!"', 'evermore.', 'longer,', 'moment', 'ease', 'ever',
'Followed', 'lie', 'nepenthe', 'Raven', 'agreeing', 'volume'}

308


{'mortal', 'Back', 'an', 'sad', 'Madam,', 'above', 'scarcely', 'Aidenn,', 'sat',
'laden', 'beak', 'all', 'heart', 'human', 'gaunt,', 'Seraphim', '"art',
'placid', 'fact', 'farther', 'craven,', 'somewhat', 'made', 'fancy', 'leave',
'feather', 'beating', 'sat,', 'streaming', 'black', 'dreaming', 'land', 'and',
'radiant', 'Leave', 'Startled', 'rare', 'tapping', 'heard', 'Shall', 'Fancy',
'than', 'what', 'relevancy', 'have', 'late', 'unhappy', 'days', 'What', 'had',
'Heaven', 'engaged', 'gloating', 'hath', 'foot-falls', 'scarce', 'was',
'countenance', 'as', 'Straight', 'surcease', 'balm', 'hesitating', 'angels',
'That', 'fast', 'Take', 'napping,', 'ancient', 'hear', 'least', 'Clasp', 'many',
'ghastly,', 'ungainly', 'answer', 'tapping,', 'curtain', 'undaunted,',
'melancholy', 'clasp', 'said,', 'air', 'Quaff,', 'take', 'saintly', 'weak',
'shall', 'Ghastly', 'gloated', 'darkness', 'Plutonian', 'sad,', 'ungainly,',
'shadow', 'stately', 'dreaming,', 'name', 'chamber', 'lattice;', '"what', 'at',
'again', 'came', 'fancy,', 'said', 'ah,', 'repeating', 'a', 'faintly',
'wandering', 'dreams', 'Ah,', 'Meant', 'Disaster', 'maiden', 'morrow;-vainly',
'beast', 'master', 'Pallas', 'seat', 'weary,', 'dream', 'entreating',
'obeisance', 'lady,', 'sainted', 'plainly,', 'back', 'each', 'head', 'rapping,',
'gave', 'faster', 'Gilead?-tell', 'ashore,', 'dared', 'Nameless', 'parting,',
'dreary,', 'that', 'aptly', 'quaff', 'distant', 'nearly', 'shaven,', 'grave',
'cannot', 'syllable', 'meaning-little', 'fantastic', 'Eagerly', 'uncertain',
'fearing,', '"tapping', 'quaint', 'adore-', 'croaking', 'marvelled', 'Caught',
'enchanted-', 'thereat', 'heart,', 'And', 'As', 'entrance', 'Darkness',
'haunted-tell', 'rapping', 'Desolate', 'upstarting-', 'Raven,', 'stayed',
'separate', 'bleak', 'pallid', 'ease', 'floating', 'lamp-light', 'Raven',
'agreeing'}

181
```

1.5 Problem 5

Verify that intersection is an assoicative operation with these particular sets. The associative law states that for all sets A, B and C.

$$A \cap (B \cap C) = (A \cap B) \cap C$$

You have three subsets of the Universe to make your example. Use them in any order. Note that only a single line of code is necessary.

[43]:

```
print((capwords and (ewords and awords)) == ((capwords and ewords) and awords))
```

```
True
```

[0]:

```
```