

# Assignment 5

March 1, 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()`.

```
[24]: fHandle = open('The Raven.txt', "r")

words_list = []

for lines in fHandle:
    lines = lines.split() ## using default split()
    for words in lines:
        words_list.append(words)

print(type(words_list))
print(len(words_list))
## print(words_list)
```

```
<class '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”.

```
[30]: Universe = set() ## or just Universe = set(words_list)

for item in words_list:
    Universe.add(item)

print(len(Universe)) ## removes duplicate words from the list
```

```
553
```

### 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.

```
[50]: import re

capwords = set()

patternCap = '[A-Z]+[a-z]+$'

for item in Universe:
    if re.search(patternCap, item):
        capwords.add(item)

print(capwords)
print(len(capwords))
```

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

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 their counts. Note that the membership conditions can use the “in” operator. A regular expression is not necessary.

```
[63]: ewords = {n for n in Universe if 'E' in n or 'e' in n}

awords = {n for n in Universe if 'A' in n or 'a' in n}

print(ewords)
print(len(ewords))

print(awords)
```

```
print(len(awords))
```

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

308

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

181

## 1.5 Problem 5

Verify that intersection is an associative 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.

```
[62]: capwords & (awords & ewords) == (capwords & awords) & ewords
```

```
[62]: True
```

```
[0]:
```