

## 8.1 Assignment: Dictionaries, Tuples, and JSON Data

File: Week 8 - Dictionaries, Tuples, and JSON Data.ipynb

Name: Jonathan Lawrence

Date: 2/01/2019

Course DSC510 - Intro to Programming

Desc: This program will process a txt file, calculate the total words, and output the number of occurrences of each word in the file.

We will create a program which performs three essential operations. It will process a txt file (Gettysburg.txt[click the link to download the text file]), calculate the total words, and output the number of occurrences of each word in the file.

Open the file and process each line. Either add each word to the dictionary with a frequency of 1 or update the word's count by 1. Nicely print the output, in this case from high to low frequency. You should use string formatting for this. (See discussion 8.3). We want to achieve each major goal with a function (one function, one action). We can find four functions that need to be created.

**add\_word:** Add each word to the dictionary. Parameters are the word and a dictionary. No return value.

```
In [1]: import string
```

```
In [2]: def add_word(words, myDict):
        for word in words:
            if word not in myDict:
                myDict[word] = 1
            else:
                myDict[word] += 1
        return
```

**Process\_line:** There is some work to be done to process the line: strip off various characters, split out the words, and so on. Parameters are a line and the dictionary. It calls add word with each processed word. No return value.

```
In [3]: def Process_line(line, myDict):
        for c in string.punctuation:           # Replace punctuation with nothing
            line = line.replace(c, "")
        words = line.lower().split()           # Split the lines up
        add_word(words, myDict)
        return
```

**Pretty\_print:** Because formatted printing can be messy and often particular to each situation

(meaning that we might need to modify it later), we separated out the printing function. The parameter is a dictionary. No return value.

```
In [4]: def Pretty_print(myDict):  
        myDict_sorted = sorted(myDict.items(), key=lambda kv: kv[1], reverse = True)  
        print("\nTotal Number of Words: {}".format(sum(myDict.values())))  
        print("\nNumber of Unique Words:{}".format(len(myDict)))  
        print("\nmyDict (Unsorted):\n{}".format(myDict))  
        print("\nmyDict (Sorted Representation via LIST):\n{}".format(myDict_sorted))  
        return
```

**main:** We will use a main function as the main program. As usual, it will open the file and call process\_line on each line. When finished, it will call pretty\_print to print the dictionary.

```
In [5]: def main():  
        myDict = {}  
        with open('gettysburg.txt', 'r') as f:           # Open file  
            for line in f:  
                Process_line(line, myDict)  
        Pretty_print(myDict)
```

In [6]: `main()`

Total Number of Words: 276

Number of Unique Words:143

myDict (Unsorted):

```
{'four': 1, 'score': 1, 'and': 6, 'seven': 1, 'years': 1, 'ago': 1, 'our': 2,
'fathers': 1, 'brought': 1, 'forth': 1, 'on': 2, 'this': 4, 'continent': 1,
'a': 7, 'new': 2, 'nation': 5, 'conceived': 2, 'in': 4, 'liberty': 1, 'dedicat
ed': 4, 'to': 8, 'the': 11, 'proposition': 1, 'that': 13, 'all': 1, 'men': 2,
'are': 3, 'created': 1, 'equal': 1, 'now': 1, 'we': 10, 'engaged': 1, 'great':
3, 'civil': 1, 'war': 2, 'testing': 1, 'whether': 1, 'or': 2, 'any': 1, 'so':
3, 'can': 5, 'long': 2, 'endure': 1, 'met': 1, 'battlefield': 1, 'of': 5, 'hav
e': 5, 'come': 1, 'dedicate': 2, 'portion': 1, 'field': 1, 'as': 1, 'final':
1, 'resting': 1, 'place': 1, 'for': 5, 'those': 1, 'who': 3, 'here': 8, 'gav
e': 2, 'their': 1, 'lives': 1, 'might': 1, 'live': 1, 'it': 5, 'is': 3, 'altog
ether': 1, 'fitting': 1, 'proper': 1, 'should': 1, 'do': 1, 'but': 2, 'large
r': 1, 'sense': 1, 'not': 5, 'consecrate': 1, 'hallow': 1, 'ground': 1, 'brav
e': 1, 'living': 2, 'dead': 3, 'struggled': 1, 'consecrated': 1, 'far': 2, 'ab
ove': 1, 'poor': 1, 'power': 1, 'add': 1, 'detract': 1, 'world': 1, 'will': 1,
'little': 1, 'note': 1, 'nor': 1, 'remember': 1, 'what': 2, 'say': 1, 'never':
1, 'forget': 1, 'they': 3, 'did': 1, 'us': 3, 'rather': 2, 'be': 2, 'unfinishe
d': 1, 'work': 1, 'which': 2, 'fought': 1, 'thus': 1, 'nobly': 1, 'advanced':
1, 'task': 1, 'remaining': 1, 'before': 1, 'from': 2, 'these': 2, 'honored':
1, 'take': 1, 'increased': 1, 'devotion': 2, 'cause': 1, 'last': 1, 'full': 1,
'measure': 1, 'highly': 1, 'resolve': 1, 'shall': 3, 'died': 1, 'vain': 1, 'un
der': 1, 'god': 1, 'birth': 1, 'freedom': 1, 'government': 1, 'people': 3, 'b
y': 1, 'perish': 1, 'earth': 1, 'abraham': 1, 'lincoln': 1, 'november': 1, '1
9': 1, '1863': 1}
```

myDict (Sorted Representation via LIST):

```
[('that', 13), ('the', 11), ('we', 10), ('to', 8), ('here', 8), ('a', 7), ('an
d', 6), ('nation', 5), ('can', 5), ('of', 5), ('have', 5), ('for', 5), ('it',
5), ('not', 5), ('this', 4), ('in', 4), ('dedicated', 4), ('are', 3), ('grea
t', 3), ('so', 3), ('who', 3), ('is', 3), ('dead', 3), ('they', 3), ('us', 3),
('shall', 3), ('people', 3), ('our', 2), ('on', 2), ('new', 2), ('conceived',
2), ('men', 2), ('war', 2), ('or', 2), ('long', 2), ('dedicate', 2), ('gave',
2), ('but', 2), ('living', 2), ('far', 2), ('what', 2), ('rather', 2), ('be',
2), ('which', 2), ('from', 2), ('these', 2), ('devotion', 2), ('four', 1), ('s
core', 1), ('seven', 1), ('years', 1), ('ago', 1), ('fathers', 1), ('brought',
1), ('forth', 1), ('continent', 1), ('liberty', 1), ('proposition', 1), ('al
l', 1), ('created', 1), ('equal', 1), ('now', 1), ('engaged', 1), ('civil',
1), ('testing', 1), ('whether', 1), ('any', 1), ('endure', 1), ('met', 1), ('b
attlefield', 1), ('come', 1), ('portion', 1), ('field', 1), ('as', 1), ('fina
l', 1), ('resting', 1), ('place', 1), ('those', 1), ('their', 1), ('lives',
1), ('might', 1), ('live', 1), ('altogether', 1), ('fitting', 1), ('proper',
1), ('should', 1), ('do', 1), ('larger', 1), ('sense', 1), ('consecrate', 1),
('hallow', 1), ('ground', 1), ('brave', 1), ('struggled', 1), ('consecrated',
1), ('above', 1), ('poor', 1), ('power', 1), ('add', 1), ('detract', 1), ('wor
ld', 1), ('will', 1), ('little', 1), ('note', 1), ('nor', 1), ('remember', 1),
('say', 1), ('never', 1), ('forget', 1), ('did', 1), ('unfinished', 1), ('wor
k', 1), ('fought', 1), ('thus', 1), ('nobly', 1), ('advanced', 1), ('task',
1), ('remaining', 1), ('before', 1), ('honored', 1), ('take', 1), ('increase
d', 1), ('cause', 1), ('last', 1), ('full', 1), ('measure', 1), ('highly', 1),
('resolve', 1), ('died', 1), ('vain', 1), ('under', 1), ('god', 1), ('birth',
```

```
1), ('freedom', 1), ('government', 1), ('by', 1), ('perish', 1), ('earth', 1),  
('abraham', 1), ('lincoln', 1), ('november', 1), ('19', 1), ('1863', 1)]
```