UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 09/03/2021
S03018

Ex. No.: 11 & 12

Files:

1. Counting in a File

Aim

To create a file, write content in it, then count the number of characters, words and lines in it.

Algorithm

```
Input: Contents of a file
Output: The number of characters, words and lines in the file

Step 1: Start
Step 2: Check if the user wants to enter content into the file.
Step 3: If yes, open the file in write mode.
    • Get a line as input from the user and write it into the file.
    • Write a new line character into the file after every entry.
    • Repeat the above steps until the user does not want to enter into the file.
    • Close the file.
Step 4: Open the file in read mode.
Step 5: Set the counters as 0 for the number of characters, words and lines.
Step 6: Read every character in the file and do the following:
    • If the character is a new line character, add 1 to the line counter.
    • If the character is a space or a full stop, add 1 to the word counter.
    • If the character is not a new line character, add 1 to the character counter.
Step 7: Close the file.
Step 8: Print the value of each counter.
Step 9: End
```

Program

```
# Python program to create a file, write content in it, then count
the number of characters, words and lines in a file.

ch=input('Do you want to write in the file? (y/n): ')
#if the user wants to start with a blank file and write in it
if ch=='y':
  file=open('notes.txt','w')
  while(ch=='y'):
    file.write(input("Enter line: "))
    file.write('\n')
    ch=input('Do you want to write in the file? (y/n): ')
  file.close()

#if ch=='n', the user can read from the pre-existing file

#the required old or new file is opened to work with
file=open('notes.txt','r')
count,ctword,ctline=0,0,0
for i in file.read():
  if i!='\n':
    count+=1
  if i==" " or i=='.':
    ctword+=1
  if i=='\n':
    ctline+=1

print("No. of characters: ",count)
print("No. of words: ",ctword)
print("No. of lines: ",ctline)
file.close()
```

Output

```
Do you want to write in the file? (y/n): y
Enter line: This is line one.
Do you want to write in the file? (y/n): y
Enter line: This is line two.
Do you want to write in the file? (y/n): y
Enter line: This is line three.
Do you want to write in the file? (y/n): n
No. of characters:  53
No. of words:  12
No. of lines:  3
```

Results / Inferences

Program for creating a file, writing some content in it, then counting the number of characters, words and lines in it, is written and executed.

2. Frequency in a File

Aim

To count the number of occurrences of each word in a file, store in a dictionary and display the frequency of each word, then find the most frequent word.

Algorithm

Input: A file
Output: The frequency of each word in the file and the most frequent word

Pre-condition: The existence of a file

```
Step 1: Start
Step 2: Open the required file in read mode.
Step 3: Create an empty string and an empty dictionary.
Step 4: Read every character in the file and do the following:
    • If the character is alphanumeric, add it to the string.
    • If not, if the string so far is not empty, then convert
      the string to lower case.
    • Check if the string is present in the dictionary and if
      not, add it as a key.
    • For every occurrence of the same string, add 1 to the
      value of the string in the dictionary.
    • Set the string as empty again and move to the next
      character in the file.
Step 5: Print the dictionary containing the frequency of each
word.
Step 6: Initialise the maximum value to be 0. Run a loop
through the values in the dictionary and if any of them are
```

greater than the maximum, then set that value as the maximum value.
Step 7: Print the key with the maximum value and the maximum value.
Step 8: End

Program

```
# Python program to count the frequency of each word in a file and to
find the most frequent word in the file.
#to display the text in the file
f=open('crystals.txt','r')
print("TEXT FROM FILE:",'\n')
print(f.read(),'\n')
f.close()
#closing and re-opening to move the file pointer to the beginning

#to find the occurence of each word
f=open('crystals.txt','r')
print("OCCURENCE OF EACH WORD:",'\n')
s=''
freq={}
for i in f.read():
  if i.isalnum():
    s+=i
  else:
    if s=='':
      continue
    s=s.lower()
    if s not in freq.keys():
      freq[s]=1
    else:
      freq[s]+=1
    s=''
print(freq)
```

```
#to find the most frequent word and its frequency
max=0
for k,v in freq.items():
  if v>max:
    max=v
    m=k
print()
print("Most frequent word: ",m)
print("Frequency: ",max)
f.close()
```

Output

```
TEXT FROM FILE:

Crystals and Crystal Systems are defined based on
Symmetry, not based on the geometry.

APF for simple cubic structure is only 52%, which means
that 48% is vacant. Therefore, this arrangement is very
rare and Polonium is the only element with this
structure.

Diamond has only 36% packing factor, but it is extremely
hard. Why? - 'coz the bond strength is really high.

OCCURENCE OF EACH WORD:

{'crystals': 1, 'and': 2, 'crystal': 1, 'systems': 1,
'are': 1, 'defined': 1, 'based': 2, 'on': 2, 'symmetry':
1, 'not': 1, 'the': 3, 'geometry': 1, 'apf': 1, 'for': 1,
'simple': 1, 'cubic': 1, 'structure': 2, 'is': 6, 'only':
3, '52': 1, 'which': 1, 'means': 1, 'that': 1, '48': 1,
'vacant': 1, 'therefore': 1, 'this': 2, 'arrangement': 1,
'very': 1, 'rare': 1, 'polonium': 1, 'element': 1,
'with': 1, 'diamond': 1, 'has': 1, '36': 1, 'packing': 1,
'factor': 1, 'but': 1, 'it': 1, 'extremely': 1, 'hard':
1, 'why': 1, 'coz': 1, 'bond': 1, 'strength': 1,
'really': 1, 'high': 1}

Most frequent word:  is
Frequency:  6
```

Results / Inferences

Program for counting the frequency of each word in a file and finding the most frequent word is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 09/03/2021
S03018

Ex. No.: 11 & 12

Command Line Arguments:

1. Format of Arguments

Aim

To print the file name, the arguments and the number of arguments given as command line arguments.

Algorithm

```
Step 1: Start
Step 2: Import the sys module.
Step 3: Print the first element in the argument list as the file name.
Step 4: Print the other elements as the arguments.
Step 5: Print the length of the list as the number of arguments.
Step 6: End
```

Program

```python
# Python program to print the file name, the arguments and the number of arguments.

import sys
print(sys.argv)
print("File name:",sys.argv[0])
for i in range(1,len(sys.argv)):
  print("Argument:",sys.argv[i])
print("No. of arguments: ",len(sys.argv))
```

Output

Command line:

```
python3 3arg.py arg1 arg2
```

Output:

```
['3arg.py', 'arg1', 'arg2']
File name: 3arg.py
Argument: arg1
Argument: arg2
No. of arguments:  3
```

Results / Inferences

Program for printing the file name, the arguments and the number of arguments in the command line is written and executed.

2. Words in a Sentence

Aim

To take a sentence as a command line argument and to count the number of words in the sentence.

Algorithm

```
Step 1: Start
Step 2: Import the sys module.
Step 3: Split the second argument in the argument list into words.
Step 4: Increment the counter by 1 for every word.
Step 5: Print the counter.
Step 6: End
```

Program

```
# Python program to take a sentence as a command line argument and
count the number of words in it.

import sys
print(sys.argv)
count=0
for i in sys.argv[1].split():
  count+=1
print("No. of words:",count)
```

Output

Command line:

```
python3 4sentcount.py 'This is my name.'
```

Output:

```
['4sentcount.py', 'This is my name.']
```

Results / Inferences

Program for taking a sentence as a command line argument and counting the number of words in it is written and executed.

3. Simple Calculator

Aim

To perform a simple calculator by taking two numbers and the operator as command line arguments.

Algorithm

```
Step 1: Start
Step 2: Import the sys module.
Step 3: Define an empty string.
Step 4: For every element in the argument list, add the element to the string.
Step 5: Use the eval() function on the string to evaluate it and print the result of the expression.
Step 6: End
```

Program

```
# Python program to execute a simple calculator by taking two numbers and an operator as command line arguments

import sys
c=''
for i in range(1,4):
  c+=sys.argv[i]
print(eval(c))
```

Output

```
python3 5calc.py 32 + 64
```

```
96

python3 5calc.py 40 // 5
8

python3 5calc.py 45 - 13
32

python3 5calc.py 21 / 4
5.25
```

Results / Inferences

Program for executing a simple calculator using command line arguments is written and executed.