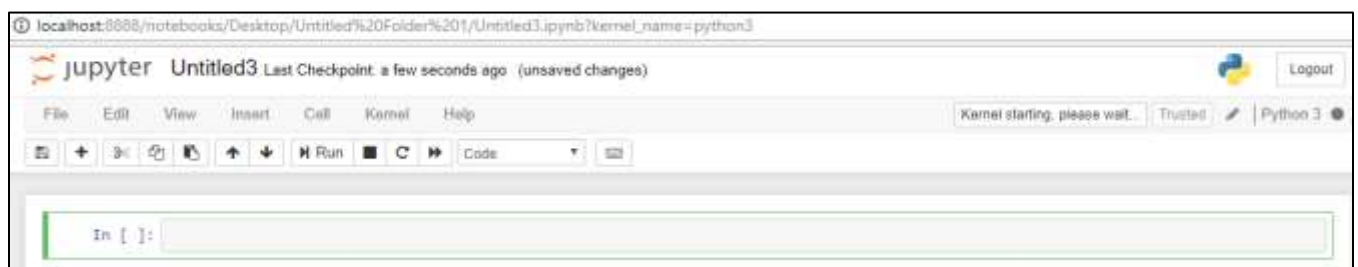


For this part, we need to process and cleanse the data before converting to sentiment or WordCloud.

	1. What did you learn from the offense you committed and its consequence/s?
1	I learned that committing an offense have its own.
2	I learned a lot and I will not do it again
3	I learn that once you violate there is a sanction so I must be responsible next time
4	I will always bring my ID to school
5	Its really important to obey rules and to not commit violations
6	I need to focus on myself about the rules


The two Raw Data from the reflection Form was merge and transferred into a text file. Click the “New” drop-down list the select Python 3.



Now first we need to import packages that we’re going to use

```
In [1]: import os
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from collections import Counter
import re
```



Shift + Enter or click  for next command line

After that, we need to open the raw file `rawData.txt` and place it into a variable

```
In [2]: os.chdir("C://Users/AikaS/Desktop")
text = open('rawData.txt', "r")
data = text.read()
```

This will allow the system to locate the directory of the text file [rawData.txt](#) and `text.read()` to transform into readable content.

Let’s check if it contains correct data.

```
In [3]: print(data)

I learned that committing an offense have its own.
I learned a lot and I will not do it again
I learn that once you violate there is a sanction so I must be responsible next ti
I will always bring my ID to school
Its really important to obey rules and to not commit violations
I need to focus on myself about the rules
Abide the rules
I learned about my mistakes
To observe all the rules of TIP and to exert more effort in keeping them
I learned punctuality and patience
To cooperate
I learn that people should be responsible
```

The Raw Data contains words that are not useful or needs to be ignored. Therefore, the NLTK has a list of stopwords that we can use to filter and remove the useless data.

Let's call stopwords for English words.

```
In [4]: ensw = stopwords.words('english')
```

Now let's check if the variable contains list of stop words

```
In [5]: print(ensw)

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you",
'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'h
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'h
'n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', '
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below'
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", '
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't"
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'ne
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wo
```

The list in the stopwords contains lowercase words and this may not identify words that are capitalized.

So in our Raw Data, we need to lowercase the words and tokenize them.

```
In [32]: textArray = word_tokenize(data.lower())
```

Next we need to remove the stop words in our data.

```
: filterArray = [item for item in textArray if item not in ensw]
```

Let's check if it works.

```
In [34]: print(filterArray)

['learned', 'committing', 'offense', '.', 'learned', 'lot', 'learn', 'viola',
'ime', 'always', 'bring', 'id', 'school', 'really', 'important', 'obey', 'ru',
'les', 'abide', 'rules', 'learned', 'mistakes', 'observe', 'rules', 'tip', 't',
'ty', 'patience', 'cooperate', 'learn', 'people', 'responsible', 'learned',
"'s", 'major', 'minor', 'offense', 'learn', 'form', 'offense', 'committed',
```

The stopwords have been removed, next transform list into text String.

```
In [41]: stringFilter = ','.join(filterArray)
```

```
In [42]: print(stringFilter)
```

learned,committing,offense,.,learned,lot,learn,violate,sanction,must,respon  
rtant,obey,rules,commit,violations,need,focus,rules,abide,rules,learned,mis  
ed,punctuality,patience,cooperate,learn,people,responsible,learned,responsi  
se,learn,form,offense,committed,never,try,violate,mandatory,12,hour,service  
student,going,way,tip,wants,.,follow,school,regulation,learn,n't,misplaced,

After removing useless words, the processed data has values that are not known to be words.

Something like this

follow, school, regulation, learn, n't, misplaced,

To clean this part, we need to check the valid English words by using nltk words.

```
In [43]: words = set(nltk.corpus.words.words())
```

Now the variable “words” has been set for the list of valid English words.

Now let's create another variable that will contain valid English words from the processed data

Remove the symbols in the list of processed data then export to text file [newList.txt](#)

```
In [34]: stringString = re.sub('\W+', '\n ', stringFilter)

# print(stringString)
f = open('newList.txt', "w")
f.write(stringString)
f.close()
```

Now we need to read each line of the created text file and check if the word is valid in NLTK English words.

```
In [50]: with open ('newList.txt') as fp:
          line = fp.readline()
          cnt = 1
          f = open('finalList.txt', "w")
          while line:
              data = line.strip()
              if data in words:
                  print(data)
                  f.write(data + "\n")
              line = fp.readline()
              cnt += 1
```

```
learned
offense
learned
lot
learn
violate
sanction
must
responsible
next
time
always
bring
id
school
```

This will also print all the valid results in [finalList.txt](#)

For the count of similar words in the final data, we need to use counter.

read the finalist text then tokenize words to list.

```
In [52]: finalText = open('finalList.txt', "r")
          finalData = finalText.read()

          tokenwords = word_tokenize(finalData)
          countWords = Counter(tokenwords)
```

Let's check for the countWords.

```
In [53]: print(countWords)
```

```
Counter({'responsible': 136, 'school': 86, 'learned': 85, 'discipline': 76, 'student': 71, 'feel': 69,
6, 'follow': 52, 'learn': 47, 'person': 46, 's': 45, 'offense': 39, 'good': 34, 'sanction': 31, 'think
'us': 29, 'time': 28, 'always': 27, 'mature': 27, 'id': 25, 'need': 25, 'make': 25, 'better': 23, 'vio
'felt': 21, 'must': 20, 'service': 20, 'mistake': 20, 'obey': 19, 'commit': 19, 'wear': 18, 'proper':
17, 'bad': 17, 'following': 17, 'order': 17, 'm': 16, 'like': 16, 'uniform': 15, 'avoid': 15, 'people'
'self': 14, 'hard': 14, 'every': 14, 'aware': 14, 'bring': 13, 'tip': 13, 'lesson': 13, 'first': 13, '
12, 'given': 12, 'rule': 12, 'never': 11, 'mandatory': 11, 'wrong': 11, 'wearing': 11, 'done': 11, 'fa
iolate': 10, 'obedient': 10, 'community': 10, 'everything': 10, 'sad': 10, 'lot': 9, 'important': 9, '
ould': 9, 'nothing': 9, 'one': 9, 'especially': 9, 'realize': 8, 'punishment': 8, 'get': 8, 'inside':
'improve': 8, 'irresponsible': 7, 'action': 7, 'break': 7, 'kind': 7, 'guilty': 7, 'able': 7, 'yes': 7
6, 'enough': 6, 'specially': 6, 'even': 6, 'hair': 6, 'careful': 6, 'want': 6, 'duty': 6, 'fine': 6, '
5, 'major': 5, 'minor': 5, 'big': 5, 'face': 5, 'days': 5, 'respect': 5, 'policy': 5, 'problem': 5, 's
ste': 5, 'campus': 5, 'individual': 5, 'many': 5, 'experience': 5, 'value': 5, 'much': 5, 'fault': 5,
ty': 5, 'peace': 5, 'maintain': 5, 'sa': 5, 'model': 5, 'simply': 5, 'great': 4, 'accept': 4, 'anythin
'well': 4, 'manual': 4, 'thing': 4, 'e': 4, 'pay': 4, 'something': 4, 'sorry': 4, 'bit': 4, 'develop:
'difficult': 4, 'fun': 4, 'assistant': 4, 'already': 4, 'place': 4, 'regret': 4, 'give': 4, 'apply': 4
s': 4, 'everyday': 4, 'act': 4, 'professional': 4, 'times': 4, 'work': 4, 'decision': 4, 'long': 4, 'n
ardless': 3, 'going': 3, 'lead': 3, 'without': 3, 'small': 3, 'importance': 3, 'serious': 3, 'another'
mply': 3, 'pants': 3, 'leave': 3, 'home': 3, 'forget': 3, 'easily': 3, 'control': 3, 'temper': 3, 'don
```

For more readable result we'll transform this list into String.

```
In [55]: countList = list(Counter((countWords)).items())
newList = '\n'.join([str(i) for i in countList])
```

```
In [56]: print(newList)
```

```
( 'learned', 85)
( 'offense', 39)
( 'lot', 9)
( 'learn', 47)
( 'violate', 10)
( 'sanction', 31)
( 'must', 20)
( 'responsible', 136)
( 'next', 3)
( 'time', 28)
( 'always', 27)
( 'bring', 13)
( 'id', 25)
( 'school', 86)
( 'really', 5)
( 'important', 9)
( 'obey', 19)
( 'commit', 19)
( 'need', 25)
( 'focus', 2)
```

We can also print this into CSV file.

```
In [60]: finalData = re.sub("[!@#'$()]", "", newList)
f = open('FrequencyCount.csv', "w")
f.write(finalData)
f.close()
```

[FrequencyCount.csv](#)

	A	B	C
1	learned	85	
2	offense	39	
3	lot	9	
4	learn	47	
5	violate	10	
6	sanction	31	
7	must	20	
8	responsib	136	
9	next	3	
10	time	28	
11	always	27	
12	bring	13	
13	id	25	
14	school	86	