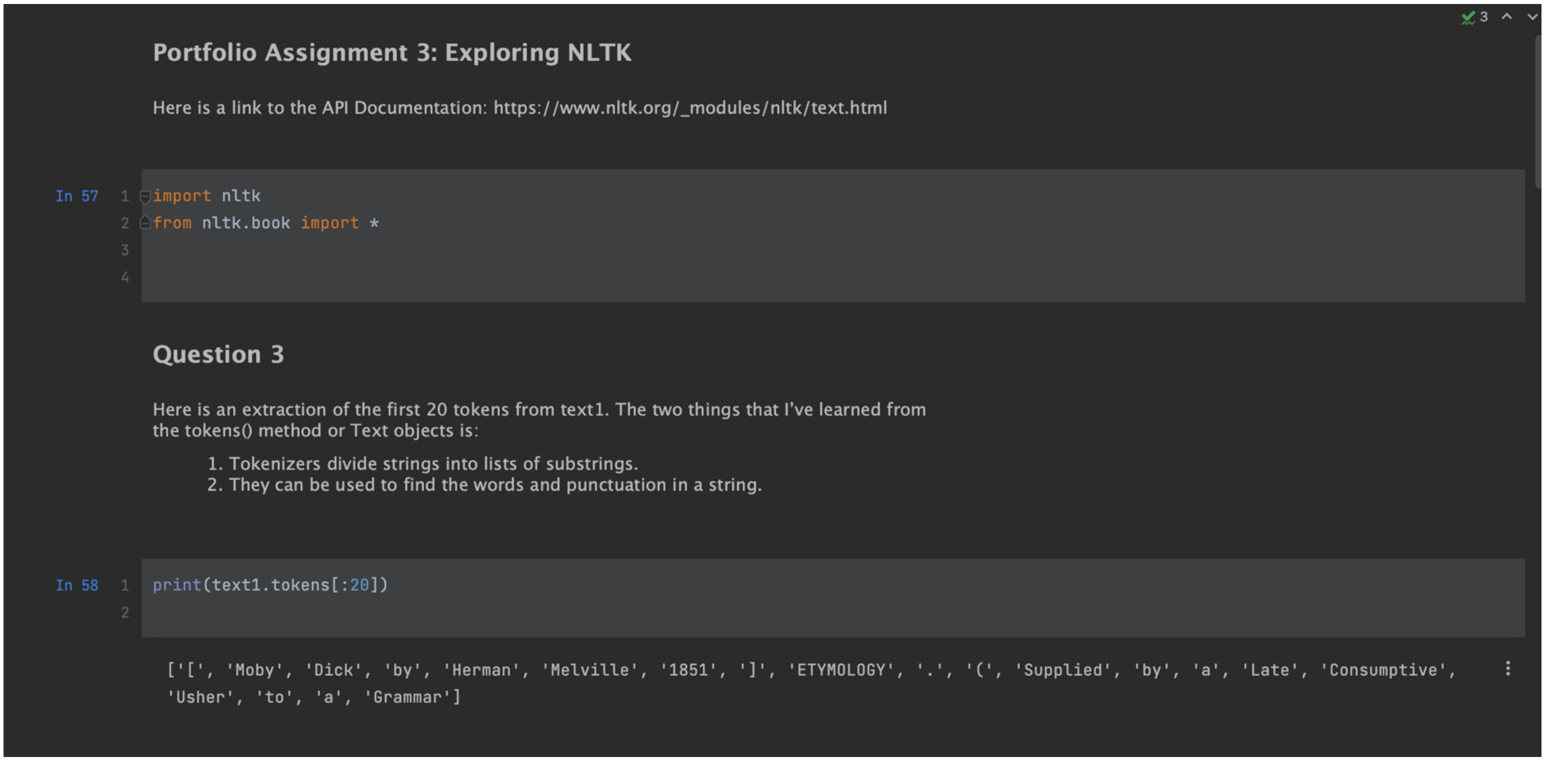


Portfolio Assignment 3: Exploring NLTK

Here is a link to the API Documentation: https://www.nltk.org/_modules/nltk/text.html

```python
import nltk
from nltk.book import *


```

Question 3

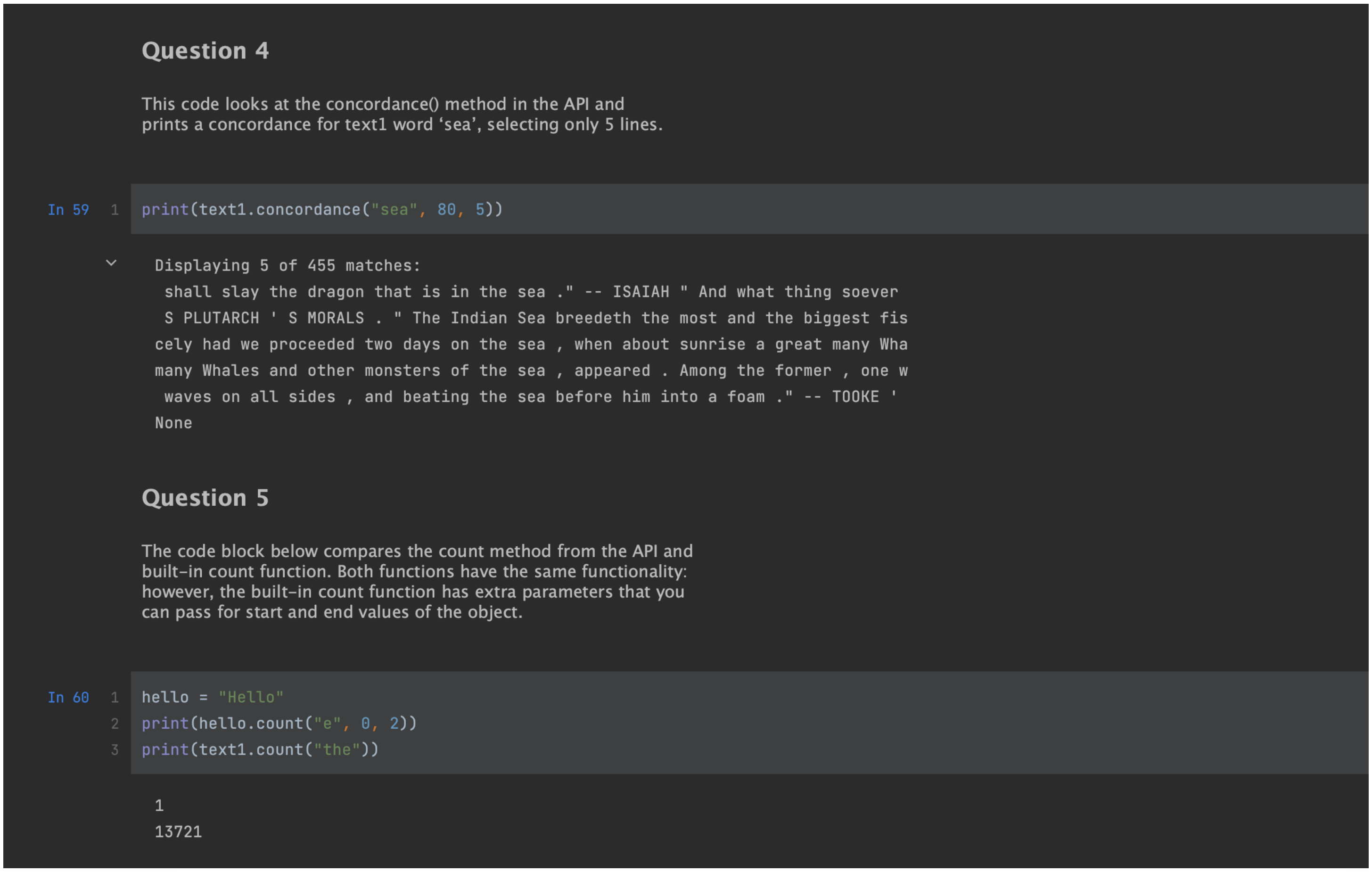
Here is an extraction of the first 20 tokens from text1. The two things that I've learned from the tokens() method or Text objects is:

1. Tokenizers divide strings into lists of substrings.
2. They can be used to find the words and punctuation in a string.

```python
print(text1.tokens[:20])

```

```
['[', 'Moby', 'Dick', 'by', 'Herman', 'Melville', '1851', ']', 'ETYMOLOGY', '.', '(', 'Supplied', 'by', 'a', 'Late', 'Consumptive',
'Usher', 'to', 'a', 'Grammar']
```



Question 4

This code looks at the concordance() method in the API and prints a concordance for text1 word 'sea', selecting only 5 lines.

```python
print(text1.concordance("sea", 80, 5))
```

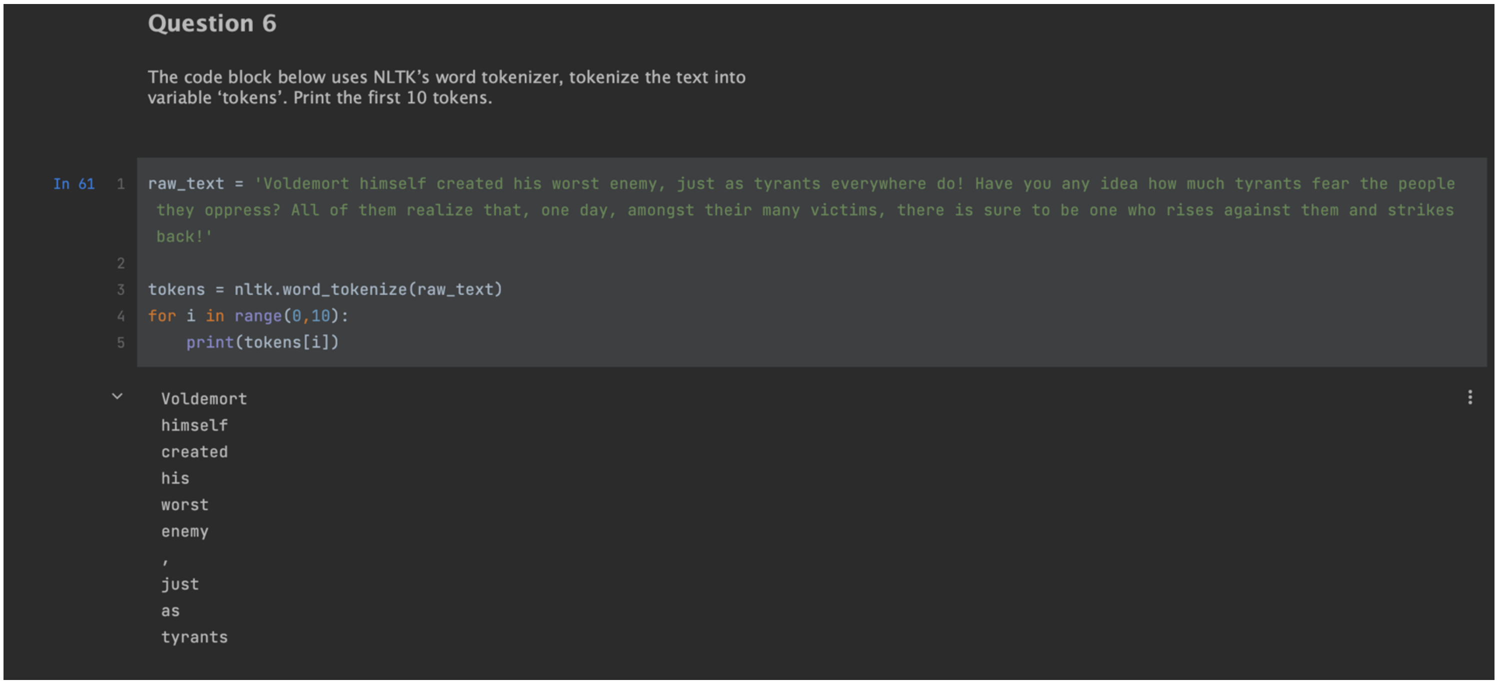
```
Displaying 5 of 455 matches:
 shall slay the dragon that is in the sea ." -- ISAIAH " And what thing soever
 S PLUTARCH ' S MORALS . " The Indian Sea breedeth the most and the biggest fis
cely had we proceeded two days on the sea , when about sunrise a great many Wha
many Whales and other monsters of the sea , appeared . Among the former , one w
 waves on all sides , and beating the sea before him into a foam ." -- TOOKE '
None
```

Question 5

The code block below compares the count method from the API and built-in count function. Both functions have the same functionality: however, the built-in count function has extra parameters that you can pass for start and end values of the object.

```python
hello = "Hello"
print(hello.count("e", 0, 2))
print(text1.count("the"))
```

```
1
13721
```



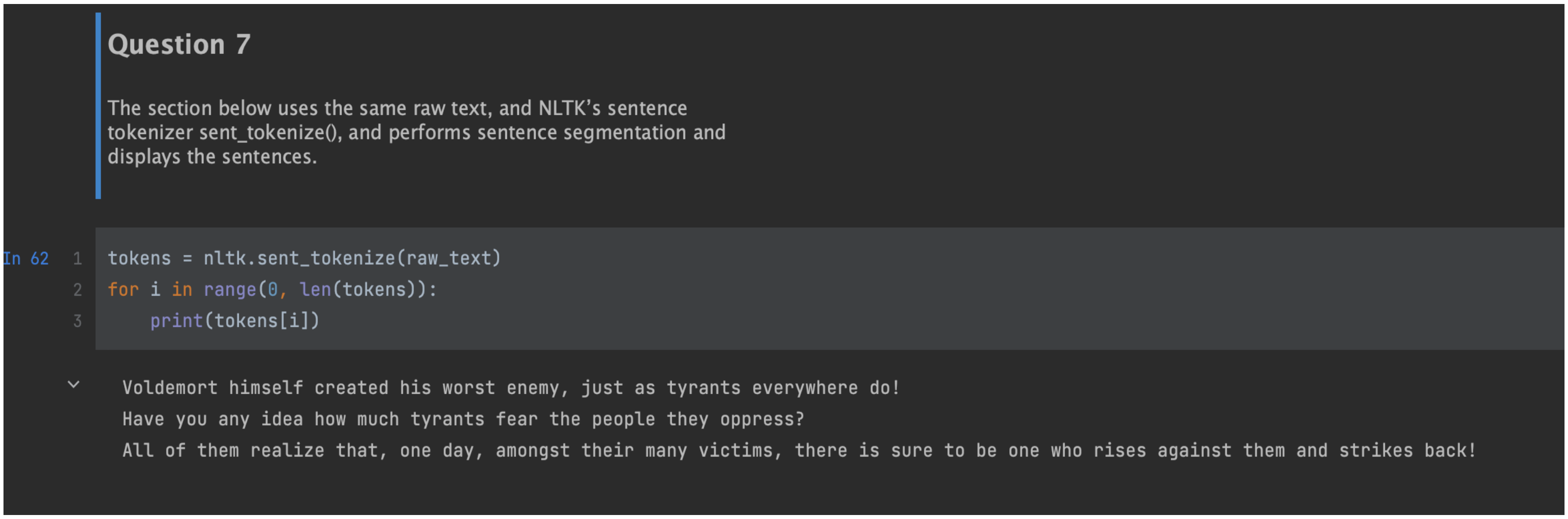
Question 6

The code block below uses NLTK's word tokenizer, tokenize the text into variable 'tokens'. Print the first 10 tokens.

```python
raw_text = 'Voldemort himself created his worst enemy, just as tyrants everywhere do! Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one who rises against them and strikes back!'

tokens = nltk.word_tokenize(raw_text)
for i in range(0,10):
    print(tokens[i])
```

```
Voldemort
himself
created
his
worst
enemy
,
just
as
tyrants
```

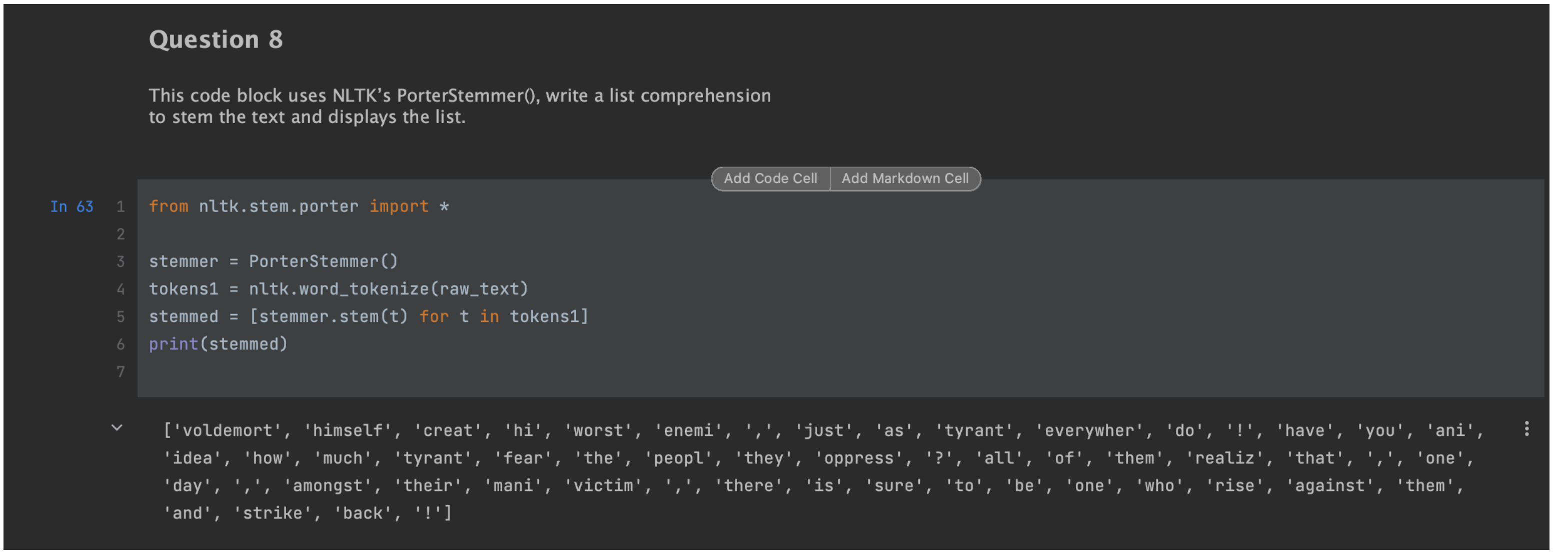


Question 7

The section below uses the same raw text, and NLTK's sentence tokenizer sent_tokenize(), and performs sentence segmentation and displays the sentences.

```python
tokens = nltk.sent_tokenize(raw_text)
for i in range(0, len(tokens)):
    print(tokens[i])
```

```
Voldemort himself created his worst enemy, just as tyrants everywhere do!
Have you any idea how much tyrants fear the people they oppress?
All of them realize that, one day, amongst their many victims, there is sure to be one who rises against them and strikes back!
```



Question 8

This code block uses NLTK's PorterStemmer(), write a list comprehension
to stem the text and displays the list.

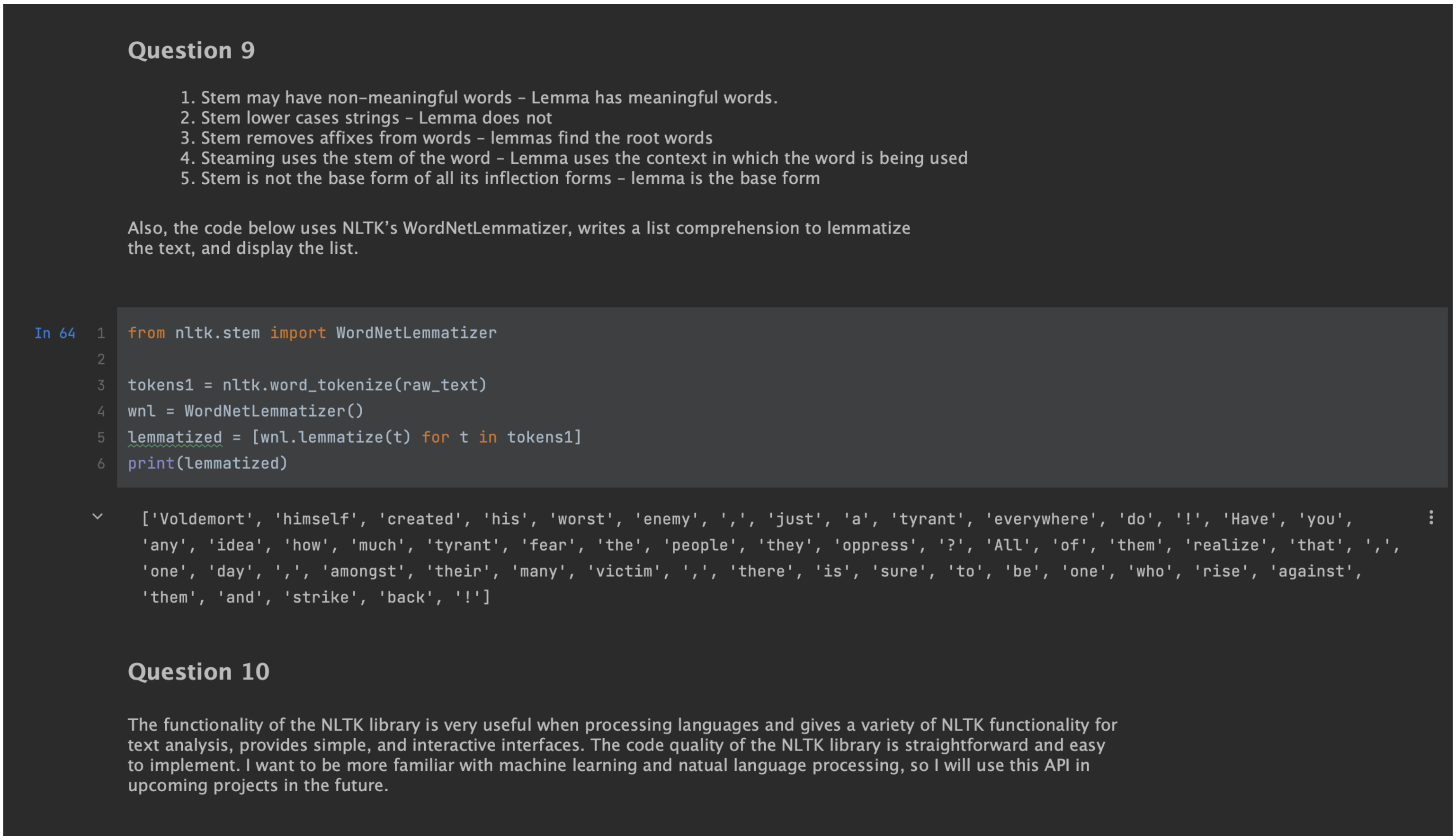
In 63

```python
from nltk.stem.porter import *

stemmer = PorterStemmer()
tokens1 = nltk.word_tokenize(raw_text)
stemmed = [stemmer.stem(t) for t in tokens1]
print(stemmed)

```

```
['voldemort', 'himself', 'creat', 'hi', 'worst', 'enemi', ',', 'just', 'as', 'tyrant', 'everywher', 'do', '!', 'have', 'you', 'ani',
'idea', 'how', 'much', 'tyrant', 'fear', 'the', 'peopl', 'they', 'oppress', '?', 'all', 'of', 'them', 'realiz', 'that', ',', 'one',
'day', ',', 'amongst', 'their', 'mani', 'victim', ',', 'there', 'is', 'sure', 'to', 'be', 'one', 'who', 'rise', 'against', 'them',
'and', 'strike', 'back', '!']
```



Question 9

1. Stem may have non-meaningful words - Lemma has meaningful words.
2. Stem lower cases strings - Lemma does not
3. Stem removes affixes from words - lemmas find the root words
4. Steaming uses the stem of the word - Lemma uses the context in which the word is being used
5. Stem is not the base form of all its inflection forms - lemma is the base form

Also, the code below uses NLTK's WordNetLemmatizer, writes a list comprehension to lemmatize
the text, and display the list.

In 64

```python
from nltk.stem import WordNetLemmatizer

tokens1 = nltk.word_tokenize(raw_text)
wnl = WordNetLemmatizer()
lemmatized = [wnl.lemmatize(t) for t in tokens1]
print(lemmatized)
```

```
['Voldemort', 'himself', 'created', 'his', 'worst', 'enemy', ',', 'just', 'a', 'tyrant', 'everywhere', 'do', '!', 'Have', 'you',
'any', 'idea', 'how', 'much', 'tyrant', 'fear', 'the', 'people', 'they', 'oppress', '?', 'All', 'of', 'them', 'realize', 'that', ',',
'one', 'day', ',', 'amongst', 'their', 'many', 'victim', ',', 'there', 'is', 'sure', 'to', 'be', 'one', 'who', 'rise', 'against',
'them', 'and', 'strike', 'back', '!']
```

Question 10

The functionality of the NLTK library is very useful when processing languages and gives a variety of NLTK functionality for text analysis, provides simple, and interactive interfaces. The code quality of the NLTK library is straightforward and easy to implement. I want to be more familiar with machine learning and natual language processing, so I will use this API in upcoming projects in the future.