```
import nltk
nltk.download('gutenberg')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
nltk.download('genesis')
nltk.download('inaugural')
nltk.download('nps_chat')
nltk.download('webtext')
nltk.download('treebank')
nltk.download('punkt')
```

```
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data]   Unzipping corpora/gutenberg.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package genesis to /root/nltk_data...
[nltk_data]   Unzipping corpora/genesis.zip.
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data]   Unzipping corpora/inaugural.zip.
[nltk_data] Downloading package nps_chat to /root/nltk_data...
[nltk_data]   Unzipping corpora/nps_chat.zip.
[nltk_data] Downloading package webtext to /root/nltk_data...
[nltk_data]   Unzipping corpora/webtext.zip.
[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data]   Unzipping corpora/treebank.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
from nltk.book import text1
text1.tokens[:20]
```

```
['[',
 'Moby',
 'Dick',
 'by',
 'Herman',
 'Melville',
 '1851',
 ']',
 'ETYMOLOGY',
 '.',
 '(',
 'Supplied',
 'by',
 'a',
```

```
 'Late',
 'Consumptive',
 'Usher',
 'to',
 'a',
 'Grammar']
```

```
text1.concordance("sea",lines=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 '
```

```
#As the text seems to be stored in a list it seems similar to the count() method for a list i
text1.count("sea")
```

```
433
```

```
#Text From Harry Potter and the Half-Blood Prince
from nltk import word_tokenize

harryPotter_Text = 'Voldemort himself created his worst enemy, just as tyrants everywhere do!
word_tokenize(harryPotter_Text)[:10]
```

```
['Voldemort',
 'himself',
 'created',
 'his',
 'worst',
 'enemy',
 ',',
 'just',
 'as',
 'tyrants']
```

```
from nltk import sent_tokenize
sent_tokenize(harryPotter_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!']
```

```
# differences between stem and lemma
#   voldemort-Voldemort
#   create-created
#   hi-his
```

```
#   enemi-enemy
#   as-a

from nltk.stem.porter import *

stemmer = PorterStemmer()
stemmed = [stemmer.stem(t) for t in word_tokenize(harryPotter_Text)]
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',
        '!']
```

```
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()
lemmatized = [wnl.lemmatize(t) for t in word_tokenize(harryPotter_Text)]
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',
        '!']
```

```
# NLTK has good functionality for parts of speech, lemma, counts for programming.
# It is good for a rules based approach or a probailistic approach to Natural Language Proces
# The code quality of NLTK is okay for doing somethings as it is pretty big.
# As we see that stem and lemma have some typos for some words like hi in stem
# I might use NLTK to write a probalistic approach to programming to see the amount of times
# that a certain word is used to see what the subject is, or getting rid of useless words tha
```