```
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
nltk.download('book')


from nltk.book import *
```

From the code cell below, I learned that Text object is a wrapper for a sequence of string tokens to support the initial exploration of texts. Furthermore, Tokens API for Text object create square brackets [] tokens to outline the text's title or name.

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

text1.concordance('sea', 40, 5)
```

```
['[', 'Moby', 'Dick', 'by', 'Herman', 'Melville', '1851', ']', 'ETYMOLOGY', '.', '(', 'Supplied',
Displaying 5 of 455 matches:
on that is in the sea ." -- ISAIAH " An
LS . " The Indian Sea breedeth the most
d two days on the sea , when about sunr
r monsters of the sea , appeared . Amon
, and beating the sea before him into a
```

The count method in the API returns the number of times the word passed in as the parameter appears in the text. It is almost the same as Python built in count function with the exception that you cannot pass in a range to count for the API's count function like you could with Python count function.

```
print(text1.count('sea'))
string = 'Hi, my name is Q, nice to meet you!'
print(string.count('m', 0, 20))
```

```
433
2
```

Source of raw text: https://iamafoodblog.com/corn-cheese/

```
from nltk import word_tokenize
raw_text = "Move over pumpkin spice, now that the weather has cooled down and it’s time to start making
tokens = word_tokenize(raw_text)
print(tokens[:10])
```

```
['Move', 'over', 'pumpkin', 'spice', ',', 'now', 'that', 'the', 'weather', 'has']
```

```
from nltk import sent_tokenize
tokens = sent_tokenize(raw_text)
print(tokens)
```

```
['Move over pumpkin spice, now that the weather has cooled down and it’s time to start making all
```

```
list = word_tokenize(raw_text)
```

```
from nltk.stem.porter import *
stemmer = PorterStemmer()
stemmedList = [stemmer.stem(x) for x in list]
print(stemmedList)
```

```
['move', 'over', 'pumpkin', 'spice', ',', 'now', 'that', 'the', 'weather', 'ha', 'cool', 'down',
```

Stem vs Lemma

cool vs cooled

combin vs combination

korean vs Korean

make vs making

cheesi vs cheesy

whi vs why

```
from nltk.stem import WordNetLemmatizer
lemmer = WordNetLemmatizer()
lemmeredList = [lemmer.lemmatize(x) for x in list]
print(lemmeredList)
```

```
['Move', 'over', 'pumpkin', 'spice', ',', 'now', 'that', 'the', 'weather', 'ha', 'cooled', 'down'
```

y, functionality-wise, I believe it does a good job of breaking down data or text into tokens to be pro
ion making it easy understand the API's code. The code does what it is expected to do while following a
ate a better data set for feeding into a bot that is trying to analyze the positive/negative sentiment

Colab paid products - Cancel contracts here