```
import nltk
from nltk.book import text1
from nltk import word_tokenize
from nltk import sent_tokenize
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
True
```

3. 1)I learned that the return type of the function tokens() is a list. 2) I learned that the function tokens() breaks down a string into words as well as punctation marks.

```
# 3
print(text1.tokens[:20])
```

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

```
# 4
newText = text1.concordance('sea', 80, 5)
print(type(newText))
```

```
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 '
<class 'NoneType'>
```

5. The count method returns the number of times the given word appears in the text. Python’s count() method, on the other hand, returns the number of elements with the given value that appear in the given list.link text

```
# 6
raw_text = 'The past year had taken a heavy toll on him, but he didnt appreciate seei
tokens = word_tokenize(raw_text)[:10]
print(tokens)
# print(tokens)
```

```
['The', 'past', 'year', 'had', 'taken', 'a', 'heavy', 'toll', 'on', 'him']
```

```
# 7
sentences = sent_tokenize(raw_text)
print(sentences)
```

```
['The past year had taken a heavy toll on him, but he didnt appreciate seeing pr
```

```
# 8
stemmer = PorterStemmer()
stemmed = [stemmer.stem(t) for t in tokens]
print(stemmed)
```

```
['the', 'past', 'year', 'had', 'taken', 'a', 'heavi', 'toll', 'on', 'him']
```

```
# 9
wnl = WordNetLemmatizer()
lemmas = [wnl.lemmatize(t) for t in tokens]
print(lemmas)
```

```
['The', 'past', 'year', 'had', 'taken', 'a', 'heavy', 'toll', 'on', 'him']
```

10.

a) I do like the functionality of the NLTK library. It has very good functions that I could use for my projects in the future. b) I do like the quality of the NLTK library. It’s well documented as well as proper names are given to variables. c) I could scrape reviews from Amazon and to find out if reviews are mostly positive or negative. Another idea is to scrape reviews from YouTube to figure out if a video is good.

Colab paid products - Cancel contracts here

✓ 1s completed at 2:46 PM