

The `tokens()` method returns the output of a word tokenizer that has been run on `text1`. Also, the tokens have not been preprocessed in any sort of way, which explains why things like punctuation marks are still included as tokens.

In [1]:

```
import nltk
from nltk.book import *
text1.tokens[0:20]
```

```
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

Out[1]:

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

In [2]:

```
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 '
```

This method counts the number of occurrences of a word in the text. It is slightly different from Python's built-in `count` method because this method utilizes the `tokens` attribute of text objects.

In [3]:

```
print(text1.count('sea'))
print(text1.tokens.count('sea'))
```

433
433

Source: Moby Dick (<https://www.gutenberg.org/files/2701/2701-h/2701-h.htm>)

```
In [4]: raw_text = 'Call me Ishmael. Some years ago—never mind how long precisely—having little  
tokens = nltk.word_tokenize(raw_text)  
tokens[0:10]
```

```
Out[4]: ['Call',  
         'me',  
         'Ishmael',  
         '.',  
         'Some',  
         'years',  
         'ago—never',  
         'mind',  
         'how',  
         'long']
```

```
In [5]: nltk.sent_tokenize(raw_text)
```

```
Out[5]: ['Call me Ishmael.',  
         'Some years ago—never mind how long precisely—having little or no money in my purse, an  
         d nothing particular to interest me on shore, I thought I would sail about a little and  
         see the watery part of the world.',  
         'It is a way I have of driving off the spleen and regulating the circulation.',  
         'Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly No  
         vember in my soul; whenever I find myself involuntarily pausing before coffin warehouse  
         s, and bringing up the rear of every funeral I meet; and especially whenever my hypos ge  
         t such an upper hand of me, that it requires a strong moral principle to prevent me from  
         deliberately stepping into the street, and methodically knocking people’s hats off—then,  
         I account it high time to get to sea as soon as I can.',  
         'This is my substitute for pistol and ball.',  
         'With a philosophical flourish Cato throws himself upon his sword; I quietly take to th  
         e ship.',  
         'There is nothing surprising in this.',  
         'If they but knew it, almost all men in their degree, some time or other, cherish very  
         nearly the same feelings towards the ocean with me.']
```

```
In [6]: stemmer = nltk.PorterStemmer()  
stemmed = [stemmer.stem(t) for t in tokens]  
stemmed
```

```
Out[6]: ['call',  
         'me',  
         'ishmael',  
         '.',  
         'some',  
         'year',  
         'ago—nev',  
         'mind',  
         'how',  
         'long',  
         'precisely-hav',  
         'littl',  
         'or',
```

'no',
'money',
'in',
'my',
'purs',
'',
'and',
'noth',
'particular',
'to',
'interest',
'me',
'on',
'shore',
'',
'i',
'thought',
'i',
'would',
'sail',
'about',
'a',
'littl',
'and',
'see',
'the',
'wateri',
'part',
'of',
'the',
'world',
'.',
'it',
'is',
'a',
'way',
'i',
'have',
'of',
'drive',
'off',
'the',
'spleen',
'and',
'regul',
'the',
'circul',
'.',
'whenev',
'i',
'find',
'myself',
'grow',
'grim',
'about',
'the',
'mouth',
';',
'whenev',
'it',

'is',
'a',
'damp',
'',
'drizzli',
'novemb',
'in',
'my',
'soul',
';',
'whenev',
'i',
'find',
'myself',
'involuntarili',
'paus',
'befor',
'coffin',
'warehous',
'',
'and',
'bring',
'up',
'the',
'rear',
'of',
'everi',
'funer',
'i',
'meet',
';',
'and',
'especi',
'whenev',
'my',
'hypo',
'get',
'such',
'an',
'upper',
'hand',
'of',
'me',
'',
'that',
'it',
'requir',
'a',
'strong',
'moral',
'princip'l',
'to',
'prevent',
'me',
'from',
'deliber',
'step',
'into',
'the',
'street',

','
'and',
'method',
'knock',
'peopl',
'',
's',
'hat',
'off-then',
'',
'i',
'account',
'it',
'high',
'time',
'to',
'get',
'to',
'sea',
'as',
'soon',
'as',
'i',
'can',
'',
'thi',
'is',
'my',
'substitut',
'for',
'pistol',
'and',
'ball',
'.',
'with',
'a',
'philosoph',
'flourish',
'cato',
'throw',
'himself',
'upon',
'hi',
'sword',
';',
'i',
'quietli',
'take',
'to',
'the',
'ship',
'.',
'there',
'is',
'noth',
'surpris',
'in',
'thi',
'.',
'if',

```
'they',  
'but',  
'knew',  
'it',  
,,  
'almost',  
'all',  
'men',  
'in',  
'their',  
'degre',  
,,  
'some',  
'time',  
'or',  
'other',  
,,  
'cherish',  
'veri',  
'nearli',  
'the',  
'same',  
'feel',  
'toward',  
'the',  
'ocean',  
'with',  
'me',  
'.'
```

voldermort-Voldermort

creat-created

hi-his

enemi-enemy

as-a

everywher-everywhere

have-Have

ani-any

people-people

all-All

```
In [7]: wn1 = nltk.WordNetLemmatizer()  
lemmas = [wn1.lemmatize(t) for t in tokens]  
lemmas
```

```
Out[7]: ['Call',  
'me',  
'Ishmael',
```

'.' ,
'Some' ,
'year' ,
'ago-never' ,
'mind' ,
'how' ,
'long' ,
'precisely-having' ,
'little' ,
'or' ,
'no' ,
'money' ,
'in' ,
'my' ,
'purse' ,
' , ' ,
'and' ,
'nothing' ,
'particular' ,
'to' ,
'interest' ,
'me' ,
'on' ,
'shore' ,
' , ' ,
'I' ,
'thought' ,
'I' ,
'would' ,
'sail' ,
'about' ,
'a' ,
'little' ,
'and' ,
'see' ,
'the' ,
'watery' ,
'part' ,
'of' ,
'the' ,
'world' ,
'.' ,
'It' ,
'is' ,
'a' ,
'way' ,
'I' ,
'have' ,
'of' ,
'driving' ,
'off' ,
'the' ,
'spleen' ,
'and' ,
'regulating' ,
'the' ,
'circulation' ,
'.' ,
'Whenever' ,
'I' ,

'find',
'myself',
'growing',
'grim',
'about',
'the',
'mouth',
';',
'whenever',
'it',
'is',
'a',
'damp',
'',
'drizzly',
'November',
'in',
'my',
'soul',
';',
'whenever',
'I',
'find',
'myself',
'involuntarily',
'pausing',
'before',
'coffin',
'warehouse',
'',
'and',
'bringing',
'up',
'the',
'rear',
'of',
'every',
'funeral',
'I',
'meet',
';',
'and',
'especially',
'whenever',
'my',
'hypo',
'get',
'such',
'an',
'upper',
'hand',
'of',
'me',
'',
'that',
'it',
'requires',
'a',
'strong',
'moral',

'principle',
'to',
'prevent',
'me',
'from',
'deliberately',
'stepping',
'into',
'the',
'street',
,',',
'and',
'methodically',
'knocking',
'people',
,',',
's',
'hat',
'off-then',
,',',
'I',
'account',
'it',
'high',
'time',
'to',
'get',
'to',
'sea',
'a',
'soon',
'a',
'I',
'can',
,',',
'This',
'is',
'my',
'substitute',
'for',
'pistol',
'and',
'ball',
,',',
'With',
'a',
'philosophical',
'flourish',
'Cato',
'throw',
'himself',
'upon',
'his',
'sword',
,',',
'I',
'quietly',
'take',
'to',
'the',

```
'ship',  
'.'  
'There',  
'is',  
'nothing',  
'surprising',  
'in',  
'this',  
'.'  
'If',  
'they',  
'but',  
'knew',  
'it',  
'',  
'almost',  
'all',  
'men',  
'in',  
'their',  
'degree',  
'',  
'some',  
'time',  
'or',  
'other',  
'',  
'cherish',  
'very',  
'nearly',  
'the',  
'same',  
'feeling',  
'towards',  
'the',  
'ocean',  
'with',  
'me',  
'.'
```

The NLTK library functions quite well when it comes to demonstrating concepts such as stemming and lemmatization. The code quality is also very easy to understand and implement. In the future, I can implement NLTK's functions in much larger NLP projects in order to save time.