

```
"""
```

```
##Step 2: This will import the necessary nltk items
```

```
"""
```

```
import nltk
```

```
nltk.download('stopwords')
```

```
nltk.download('wordnet')
```

```
nltk.download('punkt')
```

```
nltk.download('omw-1.4')
```

```
nltk.download('book')
```

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

```
[nltk_data] | Package reuters is already up-to-date!
[nltk_data] | Downloading package senseval to /root/nltk_data...
[nltk_data] | Package senseval is already up-to-date!
[nltk_data] | Downloading package state_union to /root/nltk_data...
[nltk_data] | Package state_union is already up-to-date!
[nltk_data] | Downloading package stopwords to /root/nltk_data...
[nltk_data] | Package stopwords is already up-to-date!
[nltk_data] | Downloading package swadesh to /root/nltk_data...
[nltk_data] | Package swadesh is already up-to-date!
[nltk_data] | Downloading package timit to /root/nltk_data...
[nltk_data] | Package timit is already up-to-date!
[nltk_data] | Downloading package treebank to /root/nltk_data...
[nltk_data] | Package treebank is already up-to-date!
[nltk_data] | Downloading package toolbox to /root/nltk_data...
[nltk_data] | Package toolbox is already up-to-date!
[nltk_data] | Downloading package udhr to /root/nltk_data...
[nltk_data] | Package udhr is already up-to-date!
[nltk_data] | Downloading package udhr2 to /root/nltk_data...
[nltk_data] | Package udhr2 is already up-to-date!
[nltk_data] | Downloading package unicode_samples to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package unicode_samples is already up-to-date!
[nltk_data] | Downloading package webtext to /root/nltk_data...
[nltk_data] | Package webtext 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 wordnet_ic to /root/nltk_data...
[nltk_data] | Package wordnet_ic is already up-to-date!
[nltk_data] | Downloading package words to /root/nltk_data...
[nltk_data] | Package words is already up-to-date!
[nltk_data] | Downloading package maxent_treebank_pos_tagger to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package maxent_treebank_pos_tagger is already up-
[nltk_data] | to-date!
[nltk_data] | Downloading package maxent_ne_chunker to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package maxent_ne_chunker is already up-to-date!
[nltk_data] | Downloading package universal_tagset to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package universal_tagset 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 book_grammars to
[nltk_data] | /root/nltk_data...
```

```

[nltk_data] | /root/nltk_data...
[nltk_data] | Package book_grammars is already up-to-date!
[nltk_data] | Downloading package city_database to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package city_database is already up-to-date!
[nltk_data] | Downloading package tagsets to /root/nltk_data...
[nltk_data] | Package tagsets is already up-to-date!
[nltk_data] | Downloading package panlex_swadesh to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package panlex_swadesh is already up-to-date!
[nltk_data] | Downloading package averaged_perceptron_tagger to
[nltk_data] | /root/nltk_data...
[nltk_data] | Package averaged_perceptron_tagger is already up-
[nltk_data] | to-date!
[nltk_data] |

```

```

"""

```

```

##Step 3: Investigate the tokens method of Text objects

```

```

{textname}.tokens returns a list containing all tokens contained in {textname}.
The Text object contains methods to perform multiple analyses on the text it contains.

```

```

"""

```

```

token = text1.tokens
token[:20]

```

```

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

```

```

"""

```

```

##Step 4: Investigate the concordance method of Text objects

```

```

"""

```

```

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 '

```
"""
```

```
##Step 5: Experiment with count methods
```

The count method in the API counts the amount of tokens that equal  
the specified string, while python's count method searches for the amount  
of times that the specified string appears anywhere in the text,  
even if it is just part of a word.

```
"""
```

```
words = text1.tokens  
print(text1.count("sea"))  
print(words.count("Indian"))
```

```
433
```

```
57
```

```
"""
```

```
##Step 6: Experiment with word_tokenize
```

Text source: lipsum.com randomly generated lorem ipsum text

```
"""
```

```
from nltk import word_tokenize  
raw_text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ullamcorper id  
words = word_tokenize(raw_text)  
words[:10]
```

```
['Lorem',  
 'ipsum',  
 'dolor',  
 'sit',  
 'amet',  
 ',',  
 'consectetur',  
 'adipiscing',  
 'elit',  
 '.']
```

```
"""
```

```
##Step 7: Experiment with sent_tokenize
```

```
"""
```

```
from nltk import sent_tokenize  
raw_text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ullamcorper id  
sentences = sent_tokenize(raw_text)  
sentences
```

```
['Lorem ipsum dolor sit amet, consectetur adipiscing elit.',
```

```
'Praesent ullamcorper id sapien vel placerat.',
'Nulla interdum maximus erat, in blandit nisl condimentum nec.',
'Curabitur et lorem sapien.',
'Sed a dolor non lacus condimentum pellentesque vel at orci.',
'Duis ultrices auctor ex, suscipit aliquet sem pharetra at.',
'In vestibulum neque ac tincidunt congue.',
'Sed cursus nisi non risus facilisis, molestie dictum turpis congue.',
'Suspendisse nec massa ac mauris pulvinar posuere ac vel ex.',
'Curabitur commodo lacus lorem, a cursus lacus luctus eget.',
'Nam tristique scelerisque suscipit.']
```

```
"""
```

```
##Step 8: Experiment with PorterStemmer
```

```
"""
```

```
from nltk import PorterStemmer
stemmer = PorterStemmer()
raw_text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ullamcorper id
words = word_tokenize(raw_text)
stemmed = [stemmer.stem(word) for word in words]
stemmed
```

```
['lorem',
'ipsum',
'dolor',
'sit',
'amet',
'',
'',
'consectetur',
'adipisc',
'elit',
'.',
'praesent',
'ullamcorp',
'id',
'sapien',
'vel',
'placerat',
'.',
'nulla',
'interdum',
'maximu',
'erat',
'',
'',
'in',
'blandit',
'nisl',
'condimentum',
'nec',
'.']
```

```
"""
```

```
##Step 9: Experiment with WordNetLemmatizer
```

```

stem-lemma
adipisc-adipiscing
ullamcorp-ullamcorper
maximu-maximus
"""
from nltk import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
raw_text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ullamcorper id
words = word_tokenize(raw_text)
lemmas = [lemmatizer.lemmatize(word) for word in words]
lemmas

```

```

['Lorem',
 'ipsum',
 'dolor',
 'sit',
 'amet',
 ',',
 'consectetur',
 'adipiscing',
 'elit',
 '.',
 'Praesent',
 'ullamcorper',
 'id',
 'sapien',
 'vel',
 'placerat',
 '.',
 'Nulla',
 'interdum',
 'maximus',
 'erat',
 ',',
 'in',
 'blandit',
 'nisl',
 'condimentum',
 'nec',
 '.']

```

```

"""

```

## ##Step 10: Question

The functionality of NLTK is great. It accomplished all tasks quickly and easily. The quality of the code is also great. It is very well documented and cleanly written. NLTK will be very useful in future projects where large texts must be parsed and manipulated in different ways.

```

"""

```

↳ '\n##Step 10: Question\n\nThe functionality of NLTK is great. It accomplished all tasks quickly and easily.\n\nThe quality of the code is also great. It is very well documented and cleanly \nwritten. NLTK will be very useful in future projects where large texts mu

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 9:20 PM

