In [1]:

```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\ahmad\anaconda3\lib\site-p
ackages (3.4.5)
Requirement already satisfied: six in c:\users\ahmad\anaconda3\lib\site-pa
ckages (from nltk) (1.12.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import nltk
```

In [3]:

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

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ahmad\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[3]:

```
True
```

In [4]:

```
text=" Welcome readers. I hope you find it interesting. Please do reply."
```

In [5]:

```
from nltk.tokenize import sent_tokenize
```

In [6]:

```
sent_tokenize
```

Out[6]:

```
<function nltk.tokenize.sent_tokenize(text, language='english')>
```

In [7]:

```
sent_tokenize(text)
```

Out[7]:

```
[' Welcome readers.', 'I hope you find it interesting.', 'Please do repl
y.']
```

In [8]:

```
tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
```

In [9]:

```
text=" Welcome readers. I hope you find it interesting. Please do reply."
tokenizer.tokenize(text)
```

Out[9]:

```
[' Welcome readers.', 'I hope you find it interesting.', 'Please do repl
y.']
```

In [ ]:

```

```

In [10]:

```
Arabic_text="مرحبا بكم. نحن نتعلم اساسيات مبادئ استرجاع المعلومات."
tokenizer.tokenize(Arabic_text)
```

Out[10]:

```
['مرحبا بكم.', 'نحن نتعلم اساسيات مبادئ استرجاع المعلومات']
```

In [11]:

```
text=nltk.word_tokenize("Welcome readers. I hope you find it interesting. Please do rep
ly..»")
```

In [12]:

```
print (text)
```

```
['Welcome', 'readers', '.', 'I', 'hope', 'you', 'find', 'it', 'interestin
g', '.', 'Please', 'do', 'reply..', '»']
```

In [13]:

```
input(text)
```

Out[13]:

```
'hi'
```

In [14]:

```
import nltk
from nltk.tokenize import TreebankWordTokenizer
```

In [15]:

```
tokenizer = TreebankWordTokenizer()
tokenizer.tokenize("Have a nice day. You do great!")
```

Out[15]:

```
['Have', 'a', 'nice', 'day.', 'You', 'do', 'great', '!']
```

In [16]:

```
Arabic=input("Please write a text")
```

In [17]:

```
Arabic_tokenizer=nltk.word_tokenize(Arabic)

print (Arabic_tokenizer)
```

```
['I', 'am', 'Ahmad']
```

In [18]:

```
from nltk.tokenize import RegexpTokenizer
```

In [19]:

```
tokenizer=RegexpTokenizer("[\w]+")
```

In [20]:

```
tokenizer.tokenize("Don't hesitate to ask questions or send to me your question to mohs
arem@gmail.com")
```

Out[20]:

```
['Don',
 't',
 'hesitate',
 'to',
 'ask',
 'questions',
 'or',
 'send',
 'to',
 'me',
 'your',
 'question',
 'to',
 'mohsarem',
 'gmail',
 'com']
```

In [21]:

```
tokenizer=RegexpTokenizer("\S+@\S+")
```

In [22]:

```
tokenizer.tokenize("Don't hesitate to ask questions or send to me your question to mohs
arem@gmail.com")
```

Out[22]:

```
['mohsarem@gmail.com']
```

In [23]:

```
text=[" It is a pleasant evening.","Guests, who came from US arrived at the venue","Foo
d was tasty."]
```

In [24]:

```
from nltk.tokenize import word_tokenize
tokenized_docs=[word_tokenize(doc) for doc in text]
print(tokenized_docs)
```

```
[['It', 'is', 'a', 'pleasant', 'evening', '.'], ['Guests', ',', 'who', 'ca
me', 'from', 'US', 'arrived', 'at', 'the', 'venue'], ['Food', 'was', 'tast
y', '.']]
```

In [25]:

```
import re
import string
x=re.compile('[%s]' % re.escape(string.punctuation))
```

In [28]:

```
tokenized_docs_no_punctuation = []
for review in tokenized_docs:
 new_review = []
 for token in review:
  new_token = x.sub(u'', token)
  if not new_token == u'':
   new_review.append(new_token)
 tokenized_docs_no_punctuation.append(new_review)
print(tokenized_docs_no_punctuation)
```

```
[['It', 'is', 'a', 'pleasant', 'evening'], ['Guests', 'who', 'came', 'fro
m', 'US', 'arrived', 'at', 'the', 'venue'], ['Food', 'was', 'tasty']]
```

In [29]:

```
print(text[0].upper())
```

```
 IT IS A PLEASANT EVENING.
```

In [30]:

```
print(text[0].lower())
```

```
 it is a pleasant evening.
```

In [31]:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stops=set(stopwords.words('english'))
words=["Don't",'hesitate','to','ask','questions']
[word for word in words if word not in stops]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ahmad\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[31]:

```
["Don't", 'hesitate', 'ask', 'questions']
```

In [ ]:

```
Text= "NLTK allows you to convert Text into Lowercase and uppercase. Don't hesitate to
 askquestions"
```

In [32]:

```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "yo
u're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselv
es', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'hersel
f', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'the
mselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'thes
e', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'hav
e', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'th
e', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'a
t', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through',
'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'on
ce', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'no
t', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can',
'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll',
'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't",
'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn',
"mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't",
'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "would
n't"]
```

In [33]:

```
print(stopwords.words('Arabic'))
```

```
['إذ', 'إذا', 'إذما', 'إذن', 'أف', 'أقل', 'أكثر', 'ألا', 'إلا', 'التي', 'الذي', 'الذين',
'اللاتي', 'اللائي', 'اللتان', 'اللتيا', 'اللتين', 'اللذان', 'اللذين', 'اللواتي', 'إلى', 'إليك',
م', 'إليكما', 'إليكن', 'أم', 'أما', 'أما', 'إما', 'أن', 'إن', 'إنا', 'أنا', 'أنت', 'أنتم',
'أنتما', 'أنتن', 'إنما', 'إنه', 'أنى', 'أنى', 'آه', 'آها', 'أو', 'أولاء', 'أولئك', 'أوه', 'آ
ي', 'أي', 'أيها', 'إي', 'أين', 'أين', 'أينما', 'إيه', 'بخ', 'بس', 'بعد', 'بعض', 'ب
ك', 'بكم', 'بكم', 'بكما', 'بكن', 'بل', 'بلى', 'بما', 'بماذا', 'بمن', 'بنا', 'به', 'بها',
'بهم', 'بهما', 'بهن', 'بي', 'بين', 'بيد', 'تلك', 'تلكم', 'تلكما', 'ته', 'تي', 'تين', 'تين
ك', 'ثم', 'ثمة', 'حاشا', 'حبذا', 'حتى', 'حيث', 'حيثما', 'حين', 'خلا', 'دون', 'ذا',
'ذات', 'ذاك', 'ذان', 'ذانك', 'ذلك', 'ذلكم', 'ذلكما', 'ذلكن', 'ذه', 'ذو', 'ذوا', 'ذوات
ا', 'ذواتي', 'ذي', 'ذين', 'ذينك', 'ريث', 'سوف', 'سوى', 'شتان', 'عدا', 'عسى', 'ع
ل', 'على', 'عليك', 'عليه', 'عما', 'عن', 'عند', 'غير', 'فإذا', 'فإن', 'فلا', 'فمن',
'في', 'فيم', 'فيما', 'فيه', 'فيها', 'قد', 'كأن', 'كأنما', 'كأي', 'كأين', 'كذا', 'كذلك',
'كل', 'كلا', 'كلاهما', 'كلتا', 'كلما', 'كليكما', 'كليهما', 'كم', 'كم', 'كما', 'كي', 'كي
ت', 'كيف', 'كيفما', 'لا', 'لاسيما', 'لدى', 'لست', 'لستم', 'لستما', 'لستن', 'لسن', 'لسن
ا', 'لعل', 'لك', 'لكم', 'لكما', 'لكن', 'لكنما', 'لكي', 'لكيلا', 'لم', 'لما', 'لن', 'لنا',
'له', 'لها', 'لهم', 'لهما', 'لهن', 'لو', 'لولا', 'لوما', 'لي', 'لئن', 'ليت', 'ليس', 'ليس
', 'ليست', 'ليستا', 'ليسوا', 'ما', 'ماذا', 'متى', 'مذ', 'مع', 'مما', 'ممن', 'من', 'من
ه', 'منها', 'منذ', 'مه', 'مهما', 'نحن', 'نحو', 'نعم', 'ها', 'هاتان', 'هاته', 'هاتي',
'هاتين', 'هاك', 'هاهنا', 'هذا', 'هذان', 'هذه', 'هذي', 'هذين', 'هكذا', 'هل', 'هلا', 'ه
م', 'هما', 'هن', 'هنا', 'هناك', 'هنالك', 'هو', 'هؤلاء', 'هي', 'هيا', 'هيت', 'هيها
ت', 'والذي', 'والذين', 'وإذ', 'وإذا', 'وإن', 'ولا', 'ولكن', 'ولو', 'وما', 'ومن', 'وه
يا', 'و']
```

In [ ]: