# Practical No: 7A

Name: Prathamesh Pawar | Roll No: B-23

In [1]:

```
#Download the required packages
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /home/student/nltk_dat
a...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/student/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /home/student/nltk_dat
a...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/student/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[1]:

```
True
```

In [2]:

```
#Initialize the text
#Sentence Tokenization
text=  "Tokenization  is  the  first  step  in  text  analytics.
The process  of  breaking  down  a  text  paragraph  into  smaller
chunks such as words or sentences is called Tokenization."
from nltk.tokenize import sent_tokenize
tokenized_text= sent_tokenize(text)
print(tokenized_text)
```

```
['Tokenization  is  the  first  step  in  text  analytics.', 'The
process  of  breaking  down  a  text  paragraph  into  smaller  ch
unks such as words or sentences is called Tokenization.']
```

In [3]:

```
#Word Tokenization
from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['Tokenization', 'is', 'the', 'first', 'step', 'in', 'text', 'anal
ytics', '.', 'The', 'process', 'of', 'breaking', 'down', 'a', 'tex
t', 'paragraph', 'into', 'smaller', 'chunks', 'such', 'as', 'words
', 'or', 'sentences', 'is', 'called', 'Tokenization', '.']
```

In [4]:
```
# print stop words of English
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{'who', 'has', 'which', 'over', 'himself', 'at', "she's", 'because
', 'won', "haven't", 'most', "don't", 'hasn', 'can', 'wouldn', "di
dn't", 'than', 'we', 'me', 'she', "doesn't", 'he', 'some', 'just',
"you'll", 'few', 'yourselves', 'from', 'where', 'about', 'both', '
being', 'very', 'been', 'but', "wasn't", 'no', 'such', "won't", 'w
asn', 'didn', 'll', 'our', 'as', "you're", 'ain', 'against', 'in',
'an', 'up', 'ma', 'was', "hadn't", 'through', 'any', 'weren', 'you
', "couldn't", 'his', 'when', "you've", 'they', 's', 'below', 'y',
'ours', 'couldn', 'isn', 'own', 'hers', "weren't", 'now', 'aren',
'theirs', 'once', "shan't", 'themselves', 'more', "isn't", 'what',
'there', 'don', 'this', 'off', 'd', 'so', "shouldn't", 'how', 'and
', 'after', "hasn't", 'yours', "mightn't", 'having', 'have', 'her
', 'your', 'while', 'herself', 'too', 'hadn', 'needn', 'i', "needn
't", 'be', 'am', 'between', 'to', 'into', 'on', 'does', 'had', "it
's", 'shouldn', 'under', 'further', 'mightn', 'a', 'then', 'shan',
'until', 'those', 'their', 'by', 'whom', 'each', 'if', 'above', 'o
urselves', 'o', 'should', "should've", 'these', 'that', 'during',
'myself', 're', 'do', 'out', 'yourself', 'only', 'same', 'not', 'n
or', 'haven', 'doing', 'here', 'all', 'the', 'him', 'of', 'my', 'd
own', 'will', 'them', 'other', 'or', 'is', 'for', "you'd", 'its',
'doesn', 'before', 'm', 've', 'mustn', "wouldn't", 'with', "mustn'
t", "aren't", 'why', "that'll", 'again', 'were', 'did', 'itself',
'are', 't', 'it'}
```

In [6]:
```
#Removing Punctuations and Stop Word
text= "How to remove stop words with NLTK library in Python?"
word_tokens= word_tokenize(text.lower())
filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print("Tokenized Sentence:",word_tokens)
print("Filterd  Sentence:",filtered_sentence)
```

```
Tokenized Sentence: ['how', 'to', 'remove', 'stop', 'words', 'with
', 'nltk', 'library', 'in', 'python', '?']
Filterd  Sentence: ['remove', 'stop', 'words', 'nltk', 'library',
'python', '?']
```

In [7]:
```
#Perform Stemming
from nltk.stem import PorterStemmer
e_words= ["wait", "waiting", "waited", "waits"]
ps =PorterStemmer()
for w in e_words:
    rootWord=ps.stem(w)
    print(rootWord)
```

```
wait
wait
wait
wait
```

In [8]:

```
#Perform Lemmatization
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
    print("Lemma for {} is {}".format(w, wordnet_lemmatizer.lemmat
ize(w)))
```

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

In [9]:

```
#Apply POS Tagging to text
from nltk.tokenize import word_tokenize
data="The pink sweater fit her perfectly"
words=word_tokenize(data)
for word in words:
    print(nltk.pos_tag([word]))
```

```
[('The', 'DT')]
[('pink', 'NN')]
[('sweater', 'NN')]
[('fit', 'NN')]
[('her', 'PRP$')]
[('perfectly', 'RB')]
```