```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.6)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)
```

```
# Word Tokenization
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize
data="All work and no play makes Jack a dull boy"
print(word_tokenize(data))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
['All', 'work', 'and', 'no', 'play', 'makes', 'Jack', 'a', 'dull', 'boy']
```

```
# Sentence Tokenization
from nltk.tokenize import sent_tokenize, word_tokenize
data="All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy."
print(sent_tokenize (data))
```

```
['All work and no play makes Jack a dull boy.', 'All work and no play makes Jack a dull boy.']
```

```
# Storing Words and Sentences in lists
from nltk.tokenize import sent_tokenize, word_tokenize
data="All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy."
phrases=sent_tokenize(data)
words=word_tokenize(data)
print(phrases)
print(words)
```

```
['All work and no play makes Jack a dull boy.', 'All work and no play makes Jack a dull boy.']
['All', 'work', 'and', 'no', 'play', 'makes', 'Jack', 'a', 'dull', 'boy', '.', 'All', 'work', 'and', 'no', 'play', 'make
```

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.6)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)
```

```
# Stop Word Removal
from nltk.tokenize import sent_tokenize, word_tokenize
nltk.download('stopwords')
from nltk.corpus import stopwords
data  = "Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units,
stopWords = set(stopwords.words('english'))
words = word_tokenize(data.lower())
wordsFiltered = []
for w in words:
  if w not in stopWords:
        wordsFiltered.append(w)

print("Stop Words:", stopWords)
print("FilteredWords:", wordsFiltered)
```

```
Stop Words: {'there', 'own', 'have', 'such', 'into', 'myself', 'mustn', 'up', "mustn't", 'yourself', "weren't", 'has', '
FilteredWords: ['tokenization', 'essentially', 'splitting', 'phrase', ',', 'sentence', ',', 'paragraph', ',', 'entire',
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
words=["game","gaming","gamed","games"]
ps=PorterStemmer()
for word in words:
  print(ps.stem(word))

sentence="I do gaming, the gamers play games"
words=word_tokenize(sentence)
ps=PorterStemmer()
for word in words:
  print(word+":"+ps.stem(word))
```

```
game
game
game
game
I:i
do:do
gaming:game
,:,
the:the
gamers:gamer
play:play
games:game
```

```
# Regular Expressions
# re.match() function matches a pattern at the beginning of a string
import re
result=re.match('Analytics',r'Analytics Vidhya is the largest data science community in India')
print(result)
```

```
<re.Match object; span=(0, 9), match='Analytics'>
```

```
# re.search() function matches the first occurence of a pattern at the beginning of a string
import re
r=re.search('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(r)
```

```
<re.Match object; span=(10, 17), match='founded'>
```

```
# re.findall() function matches all the occurences of the given pattern in the given raw string
import re
result2=re.findall('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(result2)
```

```
['founded', 'founded']
```