```
import nltk
from nltk.tokenize import PunktSentenceTokenizer
```

```
!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) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.12.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.2)
```

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

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
text = 'Moses supposes his toeses are roses but moses supposes erroneously '
tokens = nltk.word_tokenize(text)
print(tokens)
```

```
['Moses', 'supposes', 'his', 'toeses', 'are', 'roses', 'but', 'moses', 'supposes', 'erroneously']
```

```
import spacy
```

```
import nltk
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
for w , m in nltk.pos_tag(nltk.word_tokenize(text)):
    print(f'word : ({w}), type : ({m}) , means : ({spacy.explain(m)})')
```

```
word : (Moses), type : (NNS) , means : (noun, plural)
word : (supposes), type : (VBZ) , means : (verb, 3rd person singular present)
word : (his), type : (PRP$) , means : (pronoun, possessive)
word : (toeses), type : (NNS) , means : (noun, plural)
word : (are), type : (VBP) , means : (verb, non-3rd person singular present)
word : (roses), type : (NNS) , means : (noun, plural)
word : (but), type : (CC) , means : (conjunction, coordinating)
word : (moses), type : (VBZ) , means : (verb, 3rd person singular present)
word : (supposes), type : (NNS) , means : (noun, plural)
word : (erroneously), type : (RB) , means : (adverb)
```

```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
```

```
from nltk.stem.porter import * # import everything in porter
p_stemmer = PorterStemmer()    # object of porter stemmer
```

```
words = ['book','booking','booked','books','booker','bookstore']
```

```
for w in words:
    print(f"Word : '{w}' has porter stemming: {p_stemmer.stem(w)}")
```

```
Word : 'book' has porter stemming: book
Word : 'booking' has porter stemming: book
Word : 'booked' has porter stemming: book
Word : 'books' has porter stemming: book
Word : 'booker' has porter stemming: booker
Word : 'bookstore' has porter stemming: bookstor
```

```
import spacy
nlp = spacy.load('en_core_web_sm')
```

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
nltk.download('wordnet')
words = ["feet","radii","men","children","carpenter","fighter"]

for word in words :
    print(lemmatizer.lemmatize(word,'n'))
```

```
 [nltk_data] Downloading package wordnet to /root/nltk_data...
 foot
 radius
 men
 child
 carpenter
 fighter
```