In [1]:

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

In [2]:

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

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

Out[2]:

```
True
```

In [3]:

```
text="India is a unique country with diversity. Unity is diversity is the main slogan of
```

In [4]:

```
print(sent_tokenize(text))
```

```
['India is a unique country with diversity.', 'Unity is diversity is the m
ain slogan of the country.']
```

In [5]:

```
print(f'whitespace tokenization = {text.split()}')
```

```
whitespace tokenization = ['India', 'is', 'a', 'unique', 'country', 'wit
h', 'diversity.', 'Unity', 'is', 'diversity', 'is', 'the', 'main', 'sloga
n', 'of', 'the', 'country.']
```

In [6]:

```
from nltk.tokenize import wordpunct_tokenize
```

In [7]:

```
print(f'Punctuation-based tokenization = {wordpunct_tokenize(text)}')
```

```
Punctuation-based tokenization = ['India', 'is', 'a', 'unique', 'country',
'with', 'diversity', '.', 'Unity', 'is', 'diversity', 'is', 'the', 'main',
'slogan', 'of', 'the', 'country', '.']
```

In [8]:

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

In [9]:

```
sentence="What's your name?"
```

In [10]:

```
tokenizer = TreebankWordTokenizer()
print(f'Default/Treebank tokenization = {tokenizer.tokenize(sentence) }')
```

```
Default/Treebank tokenization = ['What', "'s", 'your', 'name', '?']
```

In [11]:

```
pip install emoji --upgrade
```

```
Defaulting to user installation because normal site-packages is not writea
bleNote: you may need to restart the kernel to use updated packages.
Requirement already satisfied: emoji in c:\users\sarvesh\appdata\roaming\p
ython\python39\site-packages (2.2.0)
```

In [12]:

```
import emoji
```

In [13]:

```
print(emoji.emojize('Hi Everyone! :grinning_face:'))
```

```
Hi Everyone! 😀
```

In [14]:

```
sentence1= emoji.emojize('Hi Everyone! :grinning face:')
```

In [15]:

```
from nltk.tokenize import TweetTokenizer
```

In [16]:

```
tokenizer = TweetTokenizer()
print(f'Tweet-rules based tokenization = {tokenizer.tokenize(sentence1)}')
```

```
Tweet-rules based tokenization = ['Hi', 'Everyone', '!', ':', 'grinning',
'face', ':']
```

In [17]:

```
sentence2="Hope, is the only thing stronger than fear! Hunger Games"
```

In [18]:

```
from nltk.corpus.reader.tagged import word_tokenize
print(word_tokenize(sentence2))
```

```
['Hope', ',', 'is', 'the', 'only', 'thing', 'stronger', 'than', 'fear',
'!', 'Hunger', 'Games']
```

In [19]:

```
from nltk.tokenize import MWETokenizer
```

In [20]:

```
tokenizer = MWETokenizer()
tokenizer.add_mwe(('Hunger', 'Games'))
print(f'Multi-word expression (MWE) tokenization = {tokenizer.tokenize(word_tokenize(sent
```

```
Multi-word expression (MWE) tokenization = ['Hope', ',', 'is', 'the', 'onl
y', 'thing', 'stronger', 'than', 'fear', '!', 'Hunger_Games']
```

In [21]:

```
# Import the toolkit and the full Porter Stemmer library
import nltk
from nltk.stem.porter import *
p_stemmer = PorterStemmer()
words = ['run','runner','running','ran','runs','easily','fairly']
for word in words:
  print(word+' --> '+p_stemmer.stem(word))
```

```
run --> run
runner --> runner
running --> run
ran --> ran
runs --> run
easily --> easili
fairly --> fairli
```

In [22]:

```
from nltk.stem.snowball import SnowballStemmer
# The Snowball Stemmer requires that you pass a language parameter
s_stemmer = SnowballStemmer(language='english')
words = ['run','runner','running','ran','runs','easily','fairly']
for word in words:
  print(word+' --> '+s_stemmer.stem(word))
```

```
run --> run
runner --> runner
running --> run
ran --> ran
runs --> run
easily --> easili
fairly --> fair
```

In [23]:

```
#Perform standard imports:
import spacy
# Load English tokenizer, tagger, parser and NER
nlp = spacy.load('en_core_web_sm')
def show_lemmas(text):
  for token in text:
    print(f'{token.text:{12}} {token.pos_:{6}} {token.lemma:<{22}} {token.lemma_}')
```

In [24]:

```
doc = nlp(u"I am a runner running in a race because I love to run since I ran today.")
show_lemmas (doc)
```

```
I           PRON   4690420944186131903    I
am          AUX    10382539506755952630   be
a           DET    11901859001352538922   a
runner      NOUN   12640964157389618806   runner
running     VERB   12767647472892411841   run
in          ADP    3002984154512732771    in
a           DET    11901859001352538922   a
race        NOUN   8048469955494714898    race
because     SCONJ  16950148841647037698   because
I           PRON   4690420944186131903    I
love        VERB   3702023516439754181    love
to          PART   3791531372978436496    to
run         VERB   12767647472892411841   run
since       SCONJ  10066841407251338481   since
I           PRON   4690420944186131903    I
ran         VERB   12767647472892411841   run
today       NOUN   11042482332948150395   today
.           PUNCT  12646065887601541794   .
```