In [1]:
```
import os
import nltk
#nltk.download()
```

In [2]:
```
import nltk.corpus
```

In [3]:
```
print(os.listdir(nltk.data.find('corpora')))
```

```
['abc', 'abc.zip', 'alpino', 'alpino.zip', 'bcp47.zip', 'biocreative_ppi', 'biocr
eative_ppi.zip', 'brown', 'brown.zip', 'brown_tei', 'brown_tei.zip', 'cess_cat',
'cess_cat.zip', 'cess_esp', 'cess_esp.zip', 'chat80', 'chat80.zip', 'city_databas
e', 'city_database.zip', 'cmudict', 'cmudict.zip', 'comparative_sentences', 'comp
arative_sentences.zip', 'comtrans.zip', 'conll2000', 'conll2000.zip', 'conll200
2', 'conll2002.zip', 'conll2007.zip', 'crubadan', 'crubadan.zip', 'dependency_tre
ebank', 'dependency_treebank.zip', 'dolch', 'dolch.zip', 'english_wordnet', 'engl
ish_wordnet.zip', 'europarl_raw', 'europarl_raw.zip', 'extended_omw.zip', 'flores
ta', 'floresta.zip', 'framenet_v15', 'framenet_v15.zip', 'framenet_v17', 'framene
t_v17.zip', 'gazetteers', 'gazetteers.zip', 'genesis', 'genesis.zip', 'gutenber
g', 'gutenberg.zip', 'ieer', 'ieer.zip', 'inaugural', 'inaugural.zip', 'indian',
'indian.zip', 'jeita.zip', 'kimmo', 'kimmo.zip', 'knbc.zip', 'lin_thesaurus', 'li
n_thesaurus.zip', 'machado.zip', 'mac_morpho', 'mac_morpho.zip', 'masc_tagged.zi
p', 'mock_corpus.zip', 'movie_reviews', 'movie_reviews.zip', 'mte_teip5', 'mte_te
ip5.zip', 'names', 'names.zip', 'nombank.1.0.zip', 'nonbreaking_prefixes', 'nonbr
eaking_prefixes.zip', 'nps_chat', 'nps_chat.zip', 'omw-1.4.zip', 'omw.zip', 'opin
ion_lexicon', 'opinion_lexicon.zip', 'panlex_swadesh.zip', 'paradigms', 'paradigm
s.zip', 'pe08', 'pe08.zip', 'pil', 'pil.zip', 'pl196x', 'pl196x.zip', 'ppattach',
'ppattach.zip', 'problem_reports', 'problem_reports.zip', 'product_reviews_1', 'p
roduct_reviews_1.zip', 'product_reviews_2', 'product_reviews_2.zip', 'propbank.zi
p', 'pros_cons', 'pros_cons.zip', 'ptb', 'ptb.zip', 'qc', 'qc.zip', 'reuters.zi
p', 'rte', 'rte.zip', 'semcor.zip', 'senseval', 'senseval.zip', 'sentence_polarit
y', 'sentence_polarity.zip', 'sentiwordnet', 'sentiwordnet.zip', 'shakespeare',
'shakespeare.zip', 'sinica_treebank', 'sinica_treebank.zip', 'smultron', 'smultro
n.zip', 'state_union', 'state_union.zip', 'stopwords', 'stopwords.zip', 'subjecti
vity', 'subjectivity.zip', 'swadesh', 'swadesh.zip', 'switchboard', 'switchboard.
zip', 'timit', 'timit.zip', 'toolbox', 'toolbox.zip', 'treebank', 'treebank.zip',
'twitter_samples', 'twitter_samples.zip', 'udhr', 'udhr.zip', 'udhr2', 'udhr2.zi
p', 'unicode_samples', 'unicode_samples.zip', 'universal_treebanks_v20.zip', 'ver
bnet', 'verbnet.zip', 'verbnet3', 'verbnet3.zip', 'webtext', 'webtext.zip', 'word
net.zip', 'wordnet2021.zip', 'wordnet2022', 'wordnet2022.zip', 'wordnet31.zip',
'wordnet_ic', 'wordnet_ic.zip', 'words', 'words.zip', 'ycoe', 'ycoe.zip']
```

In [4]:
```
AI = '''Artificial Intelligence refers to the intelligence of machines. This is
humans and animals. With Artificial Intelligence, machines perform functions suc
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of h
It is probably the fastest-growing development in the World of technology and in
AI could solve major challenges and crisis situations.'''
```

In [5]:
```
AI
```

Out[5]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of\nhumans and animals. With Artificial Intel
ligence, machines perform functions such as learning, planning, reasoning and\n
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of
human intelligence by machines.\nIt is probably the fastest-growing development
in the World of technology and innovation. Furthermore, many experts believe\nA
I could solve major challenges and crisis situations.'
```

In [6]:
```
print(AI)
```

```
Artificial Intelligence refers to the intelligence of machines. This is in contra
st to the natural intelligence of
humans and animals. With Artificial Intelligence, machines perform functions such
as learning, planning, reasoning and
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of hu
man intelligence by machines.
It is probably the fastest-growing development in the World of technology and inn
ovation. Furthermore, many experts believe
AI could solve major challenges and crisis situations.
```

In [7]:
```
type(AI)
```

Out[7]:
```
str
```

In [8]:
```
from nltk.tokenize import word_tokenize
```

In [9]:
```
AI_tokens = word_tokenize(AI)
AI_tokens
```

Out[9]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines',
 '.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals',
 '.',
 'With',
 'Artificial',
 'Intelligence',
 ',',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning',
 ',',
 'planning',
 ',',
 'reasoning',
 'and',
 'problem-solving',
 '.',
 'Most',
 'noteworthy',
 ',',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines',
 '.',
 'It',
 'is',
 'probably',
 'the',
 'fastest-growing',
 'development',
 'in',
```

```
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation',
 '.',
 'Furthermore',
 ',',
 'many',
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations',
 '.']
```

In [10]:
```
len(AI_tokens)
```

Out[10]:
```
81
```

In [11]:
```
AI
```

Out[11]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of\nhumans and animals. With Artificial Intel
ligence, machines perform functions such as learning, planning, reasoning and\n
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of
human intelligence by machines.\nIt is probably the fastest-growing development
in the World of technology and innovation. Furthermore, many experts believe\nA
I could solve major challenges and crisis situations.'
```

In [12]:
```
from nltk.tokenize import sent_tokenize
```

In [13]:
```
AI_sent = sent_tokenize(AI)
AI_sent
```

Out[13]:
```
['Artificial Intelligence refers to the intelligence of machines.',
 'This is in contrast to the natural intelligence of\nhumans and animals.',
 'With Artificial Intelligence, machines perform functions such as learning, pl
anning, reasoning and\nproblem-solving.',
 'Most noteworthy, Artificial Intelligence is the simulation of human intellige
nce by machines.',
 'It is probably the fastest-growing development in the World of technology and
innovation.',
 'Furthermore, many experts believe\nAI could solve major challenges and crisis
situations.']
```

In [14]:
```
len(AI_sent)
```

Out[14]:
```
6
```

In [15]:
```
from nltk.tokenize import blankline_tokenize
AI_blank = blankline_tokenize(AI)
AI_blank
```

Out[15]:
```
['Artificial Intelligence refers to the intelligence of machines. This is in co
ntrast to the natural intelligence of\nhumans and animals. With Artificial Inte
lligence, machines perform functions such as learning, planning, reasoning and
\nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
f human intelligence by machines.\nIt is probably the fastest-growing developme
nt in the World of technology and innovation. Furthermore, many experts believe
\nAI could solve major challenges and crisis situations.']
```

In [16]:
```
len(AI_blank)
```

Out[16]:
```
1
```

In [17]:
```
from nltk.tokenize import WhitespaceTokenizer
wt = WhitespaceTokenizer().tokenize(AI)
wt
```

Out[17]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals.',
 'With',
 'Artificial',
 'Intelligence,',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning,',
 'planning,',
 'reasoning',
 'and',
 'problem-solving.',
 'Most',
 'noteworthy,',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines.',
 'It',
 'is',
 'probably',
 'the',
 'fastest-growing',
 'development',
 'in',
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation.',
 'Furthermore,',
 'many',
```

```
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations.']
```

In [18]:
```
len(wt)
```

Out[18]:
```
70
```

In [19]:
```
s = 'Good apple cost $ 3.56 in hyderabad.'
s
```

Out[19]:
```
'Good apple cost $ 3.56 in hyderabad.'
```

In [20]:
```
from nltk.tokenize import wordpunct_tokenize
wordpunct_tokenize(s)
```

Out[20]:
```
['Good', 'apple', 'cost', '$', '3', '.', '56', 'in', 'hyderabad', '.']
```

In [21]:
```
w_p = wordpunct_tokenize(AI)
w_p
```

Out[21]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines',
 '.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals',
 '.',
 'With',
 'Artificial',
 'Intelligence',
 ',',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning',
 ',',
 'planning',
 ',',
 'reasoning',
 'and',
 'problem',
 '-',
 'solving',
 '.',
 'Most',
 'noteworthy',
 ',',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines',
 '.',
 'It',
 'is',
 'probably',
 'the',
 'fastest',
```

```
 '-',
 'growing',
 'development',
 'in',
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation',
 '.',
 'Furthermore',
 ',',
 'many',
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations',
 '.']
```

In [22]:
```
len(w_p)
```

Out[22]: 85

In [23]:
```
import nltk
```

In [24]:
```
from nltk.util import bigrams , trigrams , ngrams
```

In [25]:
```
string = 'we are learners of FSDS Course at 10:00 am'
quotes_tokens = nltk.word_tokenize(string)
quotes_tokens
```

Out[25]: ['we', 'are', 'learners', 'of', 'FSDS', 'Course', 'at', '10:00', 'am']

In [26]:
```
string
```

Out[26]: 'we are learners of FSDS Course at 10:00 am'

In [27]:
```
quotes_tokens
```

Out[27]: ['we', 'are', 'learners', 'of', 'FSDS', 'Course', 'at', '10:00', 'am']

In [28]:
```
len(quotes_tokens)
```

Out[28]: 9

In [29]:
```
quotes_bigrams = list(nltk.bigrams(quotes_tokens))
quotes_bigrams
```

Out[29]:
```
[('we', 'are'),
 ('are', 'learners'),
 ('learners', 'of'),
 ('of', 'FSDS'),
 ('FSDS', 'Course'),
 ('Course', 'at'),
 ('at', '10:00'),
 ('10:00', 'am')]
```

In [30]:
```
quotes_trigrams = list(nltk.trigrams(quotes_tokens))
quotes_trigrams
```

Out[30]:
```
[('we', 'are', 'learners'),
 ('are', 'learners', 'of'),
 ('learners', 'of', 'FSDS'),
 ('of', 'FSDS', 'Course'),
 ('FSDS', 'Course', 'at'),
 ('Course', 'at', '10:00'),
 ('at', '10:00', 'am')]
```

In [31]:
```
quotes_ngrams = list(nltk.ngrams(quotes_tokens, 7))
quotes_ngrams
```

Out[31]:
```
[('we', 'are', 'learners', 'of', 'FSDS', 'Course', 'at'),
 ('are', 'learners', 'of', 'FSDS', 'Course', 'at', '10:00'),
 ('learners', 'of', 'FSDS', 'Course', 'at', '10:00', 'am')]
```

In [32]:
```
quotes_ngrams_1 = list(nltk.ngrams(quotes_tokens, 11))
quotes_ngrams_1
```

Out[32]:
```
[]
```

In [33]:
```
from nltk import PorterStemmer       # gives the root word
pst = PorterStemmer()
```

In [34]:
```
pst.stem('affection')
```

Out[34]:
```
'affect'
```

In [35]:
```
pst.stem('playing')
```

Out[35]:
```
'play'
```

In [36]:
```
pst.stem('maximum')
```

Out[36]:
```
'maximum'
```

In [37]:
```
words_to_stem = ['give', 'giving', 'given', 'gave']

for words in words_to_stem:
    print(words+ ':'  +pst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
```

In [38]:
```
words_to_stem = ['give', 'giving', 'given', 'gave', 'thinking', 'prachi' , 'prac
```

```
for words in words_to_stem:
    print(words+ ':'  +pst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
thinking:think
prachi:prachi
prachi singare:prachi singar
singare:singar
```

In [39]:
```
from nltk import LancasterStemmer
lst = LancasterStemmer()

for words in words_to_stem:
    print(words+ ':'  +lst.stem(words))
```

```
give:giv
giving:giv
given:giv
gave:gav
thinking:think
prachi:prach
prachi singare:prachi singare
singare:sing
```

In [40]:
```
from nltk import SnowballStemmer
sbst = SnowballStemmer('english')

for words in words_to_stem:
    print(words+ ':'  +sbst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
thinking:think
prachi:prachi
prachi singare:prachi singar
singare:singar
```

In [41]:
```
stemmer = SnowballStemmer("german")
stemmer.stem("Autobahnen")
```

Out[41]: 'autobahn'

In [42]:
```
from nltk.stem import wordnet                # give the complete word
from nltk.stem import WordNetLemmatizer
word_lem = WordNetLemmatizer()
```

In [43]:
```
words_to_stem
```

Out[43]:
```
['give',
 'giving',
 'given',
 'gave',
 'thinking',
 'prachi',
 'prachi singare',
 'singare']
```

In [44]:
```
for words in words_to_stem:
    print(words+ ':' +word_lem.lemmatize(words))
```

```
give:give
giving:giving
given:given
gave:gave
thinking:thinking
prachi:prachi
prachi singare:prachi singare
singare:singare
```

In [45]:
```
from nltk.corpus import  stopwords       # corpus - folder
```

In [46]:
```
a = stopwords.words('english')
```

In [47]:
```
len(a)
```

Out[47]:
```
198
```

In [48]:
```
stopwords.words('french')
```

Out[48]:
```
['au',
 'aux',
 'avec',
 'ce',
 'ces',
 'dans',
 'de',
 'des',
 'du',
 'elle',
 'en',
 'et',
 'eux',
 'il',
 'ils',
 'je',
 'la',
 'le',
 'les',
 'leur',
 'lui',
 'ma',
 'mais',
 'me',
 'même',
 'mes',
 'moi',
 'mon',
 'ne',
 'nos',
 'notre',
 'nous',
 'on',
 'ou',
 'par',
 'pas',
 'pour',
 'qu',
 'que',
 'qui',
 'sa',
 'se',
 'ses',
 'son',
 'sur',
 'ta',
 'te',
 'tes',
 'toi',
 'ton',
 'tu',
 'un',
 'une',
 'vos',
 'votre',
 'vous',
 'c',
 'd',
 'j',
 'l',
```

```
'à',
'm',
'n',
's',
't',
'y',
'été',
'étée',
'étées',
'étés',
'étant',
'étante',
'étants',
'étantes',
'suis',
'es',
'est',
'sommes',
'êtes',
'sont',
'serai',
'seras',
'sera',
'serons',
'serez',
'seront',
'serais',
'serait',
'serions',
'seriez',
'seraient',
'étais',
'était',
'étions',
'étiez',
'étaient',
'fus',
'fut',
'fûmes',
'fûtes',
'furent',
'sois',
'soit',
'soyons',
'soyez',
'soient',
'fusse',
'fusses',
'fût',
'fussions',
'fussiez',
'fussent',
'ayant',
'ayante',
'ayantes',
'ayants',
'eu',
'eue',
'eues',
'eus',
```

```
 'ai',
 'as',
 'avons',
 'avez',
 'ont',
 'aurai',
 'auras',
 'aura',
 'aurons',
 'aurez',
 'auront',
 'aurais',
 'aurait',
 'aurions',
 'auriez',
 'auraient',
 'avais',
 'avait',
 'avions',
 'aviez',
 'avaient',
 'eut',
 'eûmes',
 'eûtes',
 'eurent',
 'aie',
 'aies',
 'ait',
 'ayons',
 'ayez',
 'aient',
 'eusse',
 'eusses',
 'eût',
 'eussions',
 'eussiez',
 'eussent']
```

In [49]:
```
len(stopwords.words('french'))
```

Out[49]: 157

In [50]:
```
stopwords.words('german')
```

Out[50]:
```
['aber',
 'alle',
 'allem',
 'allen',
 'aller',
 'alles',
 'als',
 'also',
 'am',
 'an',
 'ander',
 'andere',
 'anderem',
 'anderen',
 'anderer',
 'anderes',
 'anderm',
 'andern',
 'anderr',
 'anders',
 'auch',
 'auf',
 'aus',
 'bei',
 'bin',
 'bis',
 'bist',
 'da',
 'damit',
 'dann',
 'der',
 'den',
 'des',
 'dem',
 'die',
 'das',
 'dass',
 'daß',
 'derselbe',
 'derselben',
 'denselben',
 'desselben',
 'demselben',
 'dieselbe',
 'dieselben',
 'dasselbe',
 'dazu',
 'dein',
 'deine',
 'deinem',
 'deinen',
 'deiner',
 'deines',
 'denn',
 'derer',
 'dessen',
 'dich',
 'dir',
 'du',
 'dies',
```

```
'diese',
'diesem',
'diesen',
'dieser',
'dieses',
'doch',
'dort',
'durch',
'ein',
'eine',
'einem',
'einen',
'einer',
'eines',
'einig',
'einige',
'einigem',
'einigen',
'einiger',
'einiges',
'einmal',
'er',
'ihn',
'ihm',
'es',
'etwas',
'euer',
'eure',
'eurem',
'euren',
'eurer',
'eures',
'für',
'gegen',
'gewesen',
'hab',
'habe',
'haben',
'hat',
'hatte',
'hatten',
'hier',
'hin',
'hinter',
'ich',
'mich',
'mir',
'ihr',
'ihre',
'ihrem',
'ihren',
'ihrer',
'ihres',
'euch',
'im',
'in',
'indem',
'ins',
'ist',
'jede',
```

```
'jedem',
'jeden',
'jeder',
'jedes',
'jene',
'jenem',
'jenen',
'jener',
'jenes',
'jetzt',
'kann',
'kein',
'keine',
'keinem',
'keinen',
'keiner',
'keines',
'können',
'könnte',
'machen',
'man',
'manche',
'manchem',
'manchen',
'mancher',
'manches',
'mein',
'meine',
'meinem',
'meinen',
'meiner',
'meines',
'mit',
'muss',
'musste',
'nach',
'nicht',
'nichts',
'noch',
'nun',
'nur',
'ob',
'oder',
'ohne',
'sehr',
'sein',
'seine',
'seinem',
'seinen',
'seiner',
'seines',
'selbst',
'sich',
'sie',
'ihnen',
'sind',
'so',
'solche',
'solchem',
'solchen',
```

```
 'solcher',
 'solches',
 'soll',
 'sollte',
 'sondern',
 'sonst',
 'über',
 'um',
 'und',
 'uns',
 'unsere',
 'unserem',
 'unseren',
 'unser',
 'unseres',
 'unter',
 'viel',
 'vom',
 'von',
 'vor',
 'während',
 'war',
 'waren',
 'warst',
 'was',
 'weg',
 'weil',
 'weiter',
 'welche',
 'welchem',
 'welchen',
 'welcher',
 'welches',
 'wenn',
 'werde',
 'werden',
 'wie',
 'wieder',
 'will',
 'wir',
 'wird',
 'wirst',
 'wo',
 'wollen',
 'wollte',
 'würde',
 'würden',
 'zu',
 'zum',
 'zur',
 'zwar',
 'zwischen']
```

In [51]:
```
len(stopwords.words('german'))
```

Out[51]:
```
232
```

In [52]:
```
stopwords.words('chinese')
```

Out[52]: ['一',
'一下',
'一些',
'一切',
'一则',
'一天',
'一定',
'一方面',
'一旦',
'一时',
'一来',
'一样',
'一次',
'一片',
'一直',
'一致',
'一般',
'一起',
'一边',
'一面',
'万一',
'上下',
'上升',
'上去',
'上来',
'上述',
'上面',
'下列',
'下去',
'下来',
'下面',
'不一',
'不久',
'不仅',
'不会',
'不但',
'不光',
'不单',
'不变',
'不只',
'不可',
'不同',
'不够',
'不如',
'不得',
'不怕',
'不惟',
'不成',
'不拘',
'不敢',
'不断',
'不是',
'不比',
'不然',
'不特',
'不独',
'不管',
'不能',
'不要',
'不论',

'不足',
'不过',
'不问',
'与',
'与其',
'与否',
'与此同时',
'专门',
'且',
'两者',
'严格',
'严重',
'个',
'个人',
'个别',
'中小',
'中间',
'丰富',
'临',
'为',
'为主',
'为了',
'为什么',
'为什麽',
'为何',
'为着',
'主张',
'主要',
'举行',
'乃',
'乃至',
'么',
'之',
'之一',
'之前',
'之后',
'之後',
'之所以',
'之类',
'乌乎',
'乎',
'乘',
'也',
'也好',
'也是',
'也罢',
'了',
'了解',
'争取',
'于',
'于是',
'于是乎',
'云云',
'互相',
'产生',
'人们',
'人家',
'什么',
'什么样',
'什麽',

'今后',
'今天',
'今年',
'今後',
'仍然',
'从',
'从事',
'从而',
'他',
'他人',
'他们',
'他的',
'代替',
'以',
'以上',
'以下',
'以为',
'以便',
'以免',
'以前',
'以及',
'以后',
'以外',
'以後',
'以来',
'以至',
'以至于',
'以致',
'们',
'任',
'任何',
'任凭',
'任务',
'企图',
'伟大',
'似乎',
'似的',
'但',
'但是',
'何',
'何况',
'何处',
'何时',
'作为',
'你',
'你们',
'你的',
'使得',
'使用',
'例如',
'依',
'依照',
'依靠',
'促进',
'保持',
'俺',
'俺们',
'倘',
'倘使',
'倘或',

'倘然',
'倘若',
'假使',
'假如',
'假若',
'做到',
'像',
'允许',
'充分',
'先后',
'先後',
'先生',
'全部',
'全面',
'兮',
'共同',
'关于',
'其',
'其一',
'其中',
'其二',
'其他',
'其余',
'其它',
'其实',
'其次',
'具体',
'具体地说',
'具体说来',
'具有',
'再者',
'再说',
'冒',
'冲',
'决定',
'况且',
'准备',
'几',
'几乎',
'几时',
'凭',
'凭借',
'出去',
'出来',
'出现',
'分别',
'则',
'别',
'别的',
'别说',
'到',
'前后',
'前者',
'前进',
'前面',
'加之',
'加以',
'加入',
'加强',
'十分',

'即',
'即令',
'即使',
'即便',
'即或',
'即若',
'却不',
'原来',
'又',
'及',
'及其',
'及时',
'及至',
'双方',
'反之',
'反应',
'反映',
'反过来',
'反过来说',
'取得',
'受到',
'变成',
'另',
'另一方面',
'另外',
'只是',
'只有',
'只要',
'只限',
'叫',
'叫做',
'召开',
'叮咚',
'可',
'可以',
'可是',
'可能',
'可见',
'各',
'各个',
'各人',
'各位',
'各地',
'各种',
'各级',
'各自',
'合理',
'同',
'同一',
'同时',
'同样',
'后来',
'后面',
'向',
'向着',
'吓',
'吗',
'否则',
'吧',
'吧哒',

'吱',
'呀',
'呃',
'呕',
'呗',
'呜',
'呜呼',
'呢',
'周围',
'呵',
'呸',
'呼哧',
'咋',
'和',
'咚',
'咦',
'咱',
'咱们',
'咳',
'哇',
'哈',
'哈哈',
'哉',
'哎',
'哎呀',
'哎哟',
'哗',
'哟',
'哦',
'哩',
'哪',
'哪个',
'哪些',
'哪儿',
'哪天',
'哪年',
'哪怕',
'哪样',
'哪边',
'哪里',
'哼',
'哼唷',
'唉',
'啊',
'啐',
'啥',
'啦',
'啪达',
'喂',
'喏',
'喔唷',
'嗡嗡',
'嗬',
'嗯',
'嗳',
'嘎',
'嘎登',
'嘘',
'嘛',
'嘻',

'嘿',
'因',
'因为',
'因此',
'因而',
'固然',
'在',
'在下',
'地',
'坚决',
'坚持',
'基本',
'处理',
'复杂',
'多',
'多少',
'多数',
'多次',
'大力',
'大多数',
'大大',
'大家',
'大批',
'大约',
'大量',
'失去',
'她',
'她们',
'她的',
'好的',
'好象',
'如',
'如上所述',
'如下',
'如何',
'如其',
'如果',
'如此',
'如若',
'存在',
'宁',
'宁可',
'宁愿',
'宁肯',
'它',
'它们',
'它们的',
'它的',
'安全',
'完全',
'完成',
'实现',
'实际',
'宣布',
'容易',
'密切',
'对',
'对于',
'对应',
'将',

'少数',
'尔后',
'尚且',
'尤其',
'就',
'就是',
'就是说',
'尽',
'尽管',
'属于',
'岂但',
'左右',
'巨大',
'巩固',
'己',
'已经',
'帮助',
'常常',
'并',
'并不',
'并不是',
'并且',
'并没有',
'广大',
'广泛',
'应当',
'应用',
'应该',
'开外',
'开始',
'开展',
'引起',
'强烈',
'强调',
'归',
'当',
'当前',
'当时',
'当然',
'当着',
'形成',
'彻底',
'彼',
'彼此',
'往',
'往往',
'待',
'後来',
'後面',
'得',
'得出',
'得到',
'心里',
'必然',
'必要',
'必须',
'怎',
'怎么',
'怎么办',
'怎么样',

'怎样',
'怎麽',
'总之',
'总是',
'总的来看',
'总的来说',
'总的说来',
'总结',
'总而言之',
'恰恰相反',
'您',
'意思',
'愿意',
'慢说',
'成为',
'我',
'我们',
'我的',
'或',
'或是',
'或者',
'战斗',
'所',
'所以',
'所有',
'所谓',
'打',
'扩大',
'把',
'抑或',
'拿',
'按',
'按照',
'换句话说',
'换言之',
'据',
'掌握',
'接着',
'接著',
'故',
'故此',
'整个',
'方便',
'方面',
'旁人',
'无宁',
'无法',
'无论',
'既',
'既是',
'既然',
'时候',
'明显',
'明确',
'是',
'是否',
'是的',
'显然',
'显著',
'普通',

'普遍',
'更加',
'曾经',
'替',
'最后',
'最大',
'最好',
'最後',
'最近',
'最高',
'有',
'有些',
'有关',
'有利',
'有力',
'有所',
'有效',
'有时',
'有点',
'有的',
'有着',
'有著',
'望',
'朝',
'朝着',
'本',
'本着',
'来',
'来着',
'极了',
'构成',
'果然',
'果真',
'某',
'某个',
'某些',
'根据',
'根本',
'欢迎',
'正在',
'正如',
'正常',
'此',
'此外',
'此时',
'此间',
'毋宁',
'每',
'每个',
'每天',
'每年',
'每当',
'比',
'比如',
'比方',
'比较',
'毫不',
'没有',
'沿',
'沿着',

'注意',
'深入',
'清楚',
'满足',
'漫说',
'焉',
'然则',
'然后',
'然後',
'然而',
'照',
'照着',
'特别是',
'特殊',
'特点',
'现代',
'现在',
'甚么',
'甚而',
'甚至',
'用',
'由',
'由于',
'由此可见',
'的',
'的话',
'目前',
'直到',
'直接',
'相似',
'相信',
'相反',
'相同',
'相对',
'相对而言',
'相应',
'相当',
'相等',
'省得',
'看出',
'看到',
'看来',
'看看',
'看见',
'真是',
'真正',
'着',
'着呢',
'矣',
'知道',
'确定',
'离',
'积极',
'移动',
'突出',
'突然',
'立即',
'第',
'等',
'等等',

'管',
'紧接着',
'纵',
'纵令',
'纵使',
'纵然',
'练习',
'组成',
'经',
'经常',
'经过',
'结合',
'结果',
'给',
'绝对',
'继续',
'继而',
'维持',
'综上所述',
'罢了',
'考虑',
'者',
'而',
'而且',
'而况',
'而外',
'而已',
'而是',
'而言',
'联系',
'能',
'能否',
'能够',
'腾',
'自',
'自个儿',
'自从',
'自各儿',
'自家',
'自己',
'自身',
'至',
'至于',
'良好',
'若',
'若是',
'若非',
'范围',
'莫若',
'获得',
'虽',
'虽则',
'虽然',
'虽说',
'行为',
'行动',
'表明',
'表示',
'被',
'要',

'要不',
'要不是',
'要不然',
'要么',
'要是',
'要求',
'规定',
'觉得',
'认为',
'认真',
'认识',
'让',
'许多',
'论',
'设使',
'设若',
'该',
'说明',
'诸位',
'谁',
'谁知',
'赶',
'起',
'起来',
'起见',
'趁',
'趁着',
'越是',
'跟',
'转动',
'转变',
'转贴',
'较',
'较之',
'边',
'达到',
'迅速',
'过',
'过去',
'过来',
'运用',
'还是',
'还有',
'这',
'这个',
'这么',
'这么些',
'这么样',
'这么点儿',
'这些',
'这会儿',
'这儿',
'这就是说',
'这时',
'这样',
'这点',
'这种',
'这边',
'这里',
'这麽',

'进入',
'进步',
'进而',
'进行',
'连',
'连同',
'适应',
'适当',
'适用',
'逐步',
'逐渐',
'通常',
'通过',
'造成',
'遇到',
'遭到',
'避免',
'那',
'那个',
'那么',
'那么些',
'那么样',
'那些',
'那会儿',
'那儿',
'那时',
'那样',
'那边',
'那里',
'那麽',
'部分',
'鄙人',
'采取',
'里面',
'重大',
'重新',
'重要',
'鉴于',
'问题',
'防止',
'阿',
'附近',
'限制',
'除',
'除了',
'除此之外',
'除非',
'随',
'随着',
'随著',
'集中',
'需要',
'非但',
'非常',
'非徒',
'靠',
'顺',
'顺着',
'首先',

```
 '高兴',
 '是不是']
```

In [53]:
```
len(stopwords.words('chinese'))
```

Out[53]:
```
841
```

In [54]:
```
stopwords.words('tamil')
```

Out[54]: ['அங்கு',
'அங்கே',
'அடுத்த',
'அதனால்',
'அதன்',
'அதற்கு',
'அதிக',
'அதில்',
'அது',
'அதே',
'அதை',
'அந்த',
'அந்தக்',
'அந்தப்',
'அன்று',
'அல்லது',
'அவன்',
'அவரது',
'அவர்',
'அவர்கள்',
'அவள்',
'அவை',
'ஆகிய',
'ஆகியோர்',
'ஆகும்',
'இங்கு',
'இங்கே',
'இடத்தில்',
'இடம்',
'இதனால்',
'இதனை',
'இதன்',
'இதற்கு',
'இதில்',
'இது',
'இதை',
'இந்த',
'இந்தக்',
'இந்தத்',
'இந்தப்',
'இன்னும்',
'இப்போது',
'இரு',
'இருக்கும்',
'இருந்த',
'இருந்தது',
'இருந்து',
'இவர்',
'இவை',
'உன்',
'உள்ள',
'உள்ளது',
'உள்ளன',
'எந்த',
'என',
'எனக்',
'எனக்கு',
'எனப்படும்',
'எனவும்',
'எனவே',

'எனினும்',
'எனும்',
'என்',
'என்ன',
'என்னும்',
'என்பது',
'என்பதை',
'என்ற',
'என்று',
'என்றும்',
'எல்லாம்',
'ஏன்',
'ஒரு',
'ஒரே',
'ஓர்',
'கொண்ட',
'கொண்டு',
'கொள்ள',
'சற்று',
'சிறு',
'சில',
'சேர்ந்த',
'தனது',
'தன்',
'தவிர',
'தான்',
'நான்',
'நாம்',
'நீ',
'பற்றி',
'பற்றிய',
'பல',
'பலரும்',
'பல்வேறு',
'பின்',
'பின்னர்',
'பிற',
'பிறகு',
'பெரும்',
'பேர்',
'போது',
'போன்ற',
'போல',
'போல்',
'மட்டுமே',
'மட்டும்',
'மற்ற',
'மற்றும்',
'மிக',
'மிகவும்',
'மீது',
'முதல்',
'முறை',
'மேலும்',
'மேல்',
'யார்',
'வந்த',
'வந்து',
'வரும்',
'வரை',

```
 'வரையில்',
 'விட',
 'விட்டு',
 'வேண்டும்',
 'வேறு']
```

In [55]:
```
stopwords.words('bengali')
```

Out[55]: ['অতএব',
'অথচ',
'অথবা',
'অনুযায়ী',
'অনেক',
'অনেকে',
'অনেকেই',
'অন্তত',
'অন্য',
'অবধি',
'অবশ্য',
'অর্থাত',
'আই',
'আগামী',
'আগে',
'আগেই',
'আছে',
'আজ',
'আদ্যভাগে',
'আপনার',
'আপনি',
'আবার',
'আমরা',
'আমাকে',
'আমাদের',
'আমার',
'আমি',
'আর',
'আরও',
'ই',
'ইত্যাদি',
'ইহা',
'উচিত',
'উত্তর',
'উনি',
'উপর',
'উপরে',
'এ',
'এঁদের',
'এঁরা',
'এই',
'একই',
'একটি',
'একবার',
'একে',
'এক্',
'এখন',
'এখনও',
'এখানে',
'এখানেই',
'এটা',
'এটাই',
'এটি',
'এত',
'এতটাই',
'এতে',
'এদের',
'এব',
'এবং',
'এবার',

'এমন',
'এমনকী',
'এমনি',
'এর',
'এরা',
'এল',
'এস',
'এসে',
'ঐ',
'ও',
'ওঁদের',
'ওঁর',
'ওঁরা',
'ওই',
'ওকে',
'ওখানে',
'ওদের',
'ওর',
'ওরা',
'কখনও',
'কত',
'কবে',
'কমনে',
'কয়েক',
'কয়েকটি',
'করছে',
'করছেন',
'করতে',
'করবে',
'করবেন',
'করলে',
'করলেন',
'করা',
'করাই',
'করায়',
'করার',
'করি',
'করিতে',
'করিয়া',
'করিয়ে',
'করে',
'করেই',
'করেছিলেন',
'করেছে',
'করেছেন',
'করেন',
'কাউকে',
'কাছ',
'কাছে',
'কাজ',
'কাজে',
'কারও',
'কারণ',
'কি',
'কিংবা',
'কিছু',
'কিছুই',
'কিন্তু',
'কী',
'কে',

'কেউ',
'কেউই',
'কেখা',
'কেন',
'কোটি',
'কোন',
'কোনও',
'কোনো',
'ক্ষেত্রে',
'কয়েক',
'খুব',
'গিয়ে',
'গিয়েছে',
'গিয়ে',
'গুলি',
'গেছে',
'গেল',
'গেলে',
'গোটা',
'চলে',
'চান',
'চায়',
'চার',
'চালু',
'চেয়ে',
'চেষ্টা',
'ছাড়া',
'ছাড়াও',
'ছিল',
'ছিলেন',
'জন',
'জনকে',
'জনের',
'জন্য',
'জন্যওজে',
'জানতে',
'জানা',
'জানানো',
'জানায়',
'জানিয়ে',
'জানিয়েছে',
'জে',
'জনজন',
'টি',
'ঠিক',
'তখন',
'তত',
'তথা',
'তবু',
'তবে',
'তা',
'তাঁকে',
'তাঁদের',
'তাঁর',
'তাঁরা',
'তাঁাহারা',
'তাই',
'তাও',
'তাকে',
'তাতে',

'তাদের',
'তার',
'তারপর',
'তারা',
'তারৈ',
'তাহলে',
'তাহা',
'তাহাতে',
'তাহার',
'তিনঐ',
'তিনি',
'তিনিও',
'তুমি',
'তুলে',
'তেমন',
'তো',
'তোমার',
'থাকবে',
'থাকবেন',
'থাকা',
'থাকায়',
'থাকে',
'থাকেন',
'থেকে',
'থেকেই',
'থেকেও',
'দিকে',
'দিতে',
'দিন',
'দিয়ে',
'দিয়েছে',
'দিয়েছেন',
'দিলেন',
'দু',
'দুই',
'দুটি',
'দুটো',
'দেওয়া',
'দেওয়ার',
'দেওয়া',
'দেখতে',
'দেখা',
'দেখে',
'দেন',
'দেয়',
'দ্বারা',
'ধরা',
'ধরে',
'ধামার',
'নতুন',
'নয়',
'না',
'নাই',
'নাকি',
'নাগাদ',
'নানা',
'নিজে',
'নিজেই',
'নিজেদের',
'নিজের',

'নিতে',
'নিয়ে',
'নিয়ে',
'নেই',
'নেওয়া',
'নেওয়ার',
'নেওয়া',
'নয়',
'পক্ষে',
'পর',
'পরে',
'পরেই',
'পরেও',
'পর্যন্ত',
'পাওয়া',
'পাচ',
'পারি',
'পারে',
'পারেন',
'পি',
'পেয়ে',
'পেয়্র',
'প্রতি',
'প্রথম',
'প্রভৃতি',
'প্রযন্ত',
'প্রাথমিক',
'প্রায়',
'প্রায়',
'ফলে',
'ফিরে',
'ফের',
'বক্তব্য',
'বদলে',
'বন',
'বরং',
'বলতে',
'বলল',
'বললেন',
'বলা',
'বলে',
'বলেছেন',
'বলেন',
'বসে',
'বহু',
'বা',
'বাদে',
'বার',
'বি',
'বিনা',
'বিভিন্ন',
'বিশেষ',
'বিষয়টি',
'বেশ',
'বেশি',
'ব্যবহার',
'ব্যাপারে',
'ভাবে',
'ভাবেই',
'মতো',

'মতোই',
'মধ্যভাগে',
'মধ্যে',
'মধ্যেই',
'মধ্যেও',
'মনে',
'মাত্র',
'মাধ্যমে',
'মোট',
'মোটেই',
'যখন',
'যত',
'যতটা',
'যথেষ্ট',
'যদি',
'যদিও',
'যা',
'যাঁর',
'যাঁরা',
'যাওয়া',
'যাওয়ার',
'যাওয়া',
'যাকে',
'যাচ্ছে',
'যাতে',
'যাদের',
'যান',
'যাবে',
'যায়',
'যার',
'যারা',
'যিনি',
'যে',
'যেখানে',
'যেতে',
'যেন',
'যেমন',
'র',
'রকম',
'রয়েছে',
'রাখা',
'রেখে',
'লক্ষ',
'শুধু',
'শুরু',
'সঙ্গে',
'সঙ্গেও',
'সব',
'সবার',
'সমস্ত',
'সম্প্রতি',
'সহ',
'সহিত',
'সাধারণ',
'সামনে',
'সি',
'সুতরাং',
'সে',
'সেই',
'সেখান',

```
 'সেখানে',
 'সেটা',
 'সেটাই',
 'সেটাও',
 'সেটি',
 'স্পষ্ট',
 'স্বয়ং',
 'হইতে',
 'হইবে',
 'হইয়া',
 'হওয়া',
 'হওয়ায়',
 'হওয়ার',
 'হচ্ছে',
 'হত',
 'হতে',
 'হতেই',
 'হন',
 'হবে',
 'হবেন',
 'হয়',
 'হয়তো',
 'হয়নি',
 'হয়ে',
 'হয়েই',
 'হয়েছিল',
 'হয়েছে',
 'হয়েছেন',
 'হল',
 'হলে',
 'হলেই',
 'হলেও',
 'হলো',
 'হাজার',
 'হিসাবে',
 'হৈলে',
 'হোক',
 'হয়']
```

In [56]:
```
sent = " sam is a natural when it is comes to drawing"
sent_tokens = word_tokenize(sent)
sent_tokens
```

Out[56]:
```
['sam', 'is', 'a', 'natural', 'when', 'it', 'is', 'comes', 'to', 'drawing']
```

In [57]:
```
for token in sent_tokens:
    print(nltk.pos_tag([token]))
```

```
[('sam', 'NN')]
[('is', 'VBZ')]
[('a', 'DT')]
[('natural', 'JJ')]
[('when', 'WRB')]
[('it', 'PRP')]
[('is', 'VBZ')]
[('comes', 'VBZ')]
[('to', 'TO')]
[('drawing', 'VBG')]
```

In [58]:
```
# NER = NAMED ENTITY RECOGNITION
from nltk import ne_chunk
```

In [59]:
```
NE_sent = 'the US president stays in the WHITEHOUSE'
```

In [60]:
```
NE_tokens = word_tokenize(NE_sent)
NE_tokens
```

Out[60]:
```
['the', 'US', 'president', 'stays', 'in', 'the', 'WHITEHOUSE']
```

In [61]:
```
NE_tags = nltk.pos_tag(NE_tokens)
NE_tags
```

Out[61]:
```
[('the', 'DT'),
 ('US', 'NNP'),
 ('president', 'NN'),
 ('stays', 'NNS'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('WHITEHOUSE', 'NNP')]
```

In [62]:
```
NE_NER = ne_chunk(NE_tags)
print(NE_NER)
```

```
(S
  the/DT
  (GSP US/NNP)
  president/NN
  stays/NNS
  in/IN
  the/DT
  (ORGANIZATION WHITEHOUSE/NNP))
```

In [63]:
```
# pip install WordCloud
```

In [64]:
```
# NLG :- natural language generation
```

In [65]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

In [66]:
```
text=("Python Python Python Matplotlib Matplotlib Seaborn Network Plot Violin Ch
```

In [67]:
```
# craete the word cloud object
wordcloud = WordCloud(width=420, height=200, margin = 2, background_color = 'bla
```

In [68]:
```
# display the generated image:
plt.imshow(wordcloud, interpolation='gaussian')
plt.axis("off")
plt.margins(x=0, y=0)
plt.show()
```

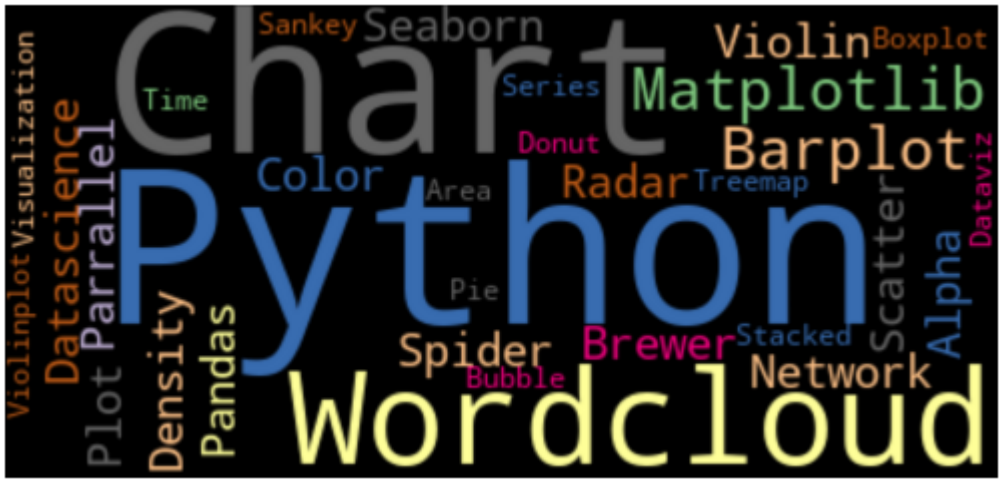
Chart
Python
Wordcloud
Sankey
Seaborn
Violin
Boxplot
Time
Series
Matplotlib
Donut
Barplot
Color
Area
Radar
Treemap
Dataviz
Violinplot
Visualization
Datascience
Parrallel
Scatter
Alpha
Pie
Density
Pandas
Spider
Brewer
Stacked
Network
Plot
Bubble

In [ ]:

In [ ]: