In [2]:

```
# Tokenise into sentences (returns a list):
from nltk.tokenize import sent_tokenize, word_tokenize

EXAMPLE_TEXT = "Hello Mr. Jones, how are you doing today? The weather is grea
t. The sky is pinkish-blue. You shouldn’t need an umbrella."
print(sent_tokenize(EXAMPLE_TEXT))
```

```
['Hello Mr. Jones, how are you doing today?', 'The weather is great.', 'The s
ky is pinkish-blue.', 'You shouldn’t need an umbrella.']
```

In [3]:

```
# Tokenise into words (returns a list):
from nltk.tokenize import sent_tokenize, word_tokenize

EXAMPLE_TEXT = "Hello Mr. Jones, how are you doing today? The weather is grea
t. The sky is pinkish-blue. You shouldn’t need an umbrella."
print(word_tokenize(EXAMPLE_TEXT))
```

```
['Hello', 'Mr.', 'Jones', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'T
he', 'weather', 'is', 'great', '.', 'The', 'sky', 'is', 'pinkish-blue', '.',
'You', 'shouldn', '’', 't', 'need', 'an', 'umbrella', '.']
```

In [4]:

```
# STOP WORDS - very common words that have little or no meaning
# thus should not be stored or processed
# do not filter out if PHRASE SEARCHING is required, though!
from nltk.corpus import stopwords
print(set(stopwords.words('english')))
```

```
{"that'll", 'an', 'mustn', 'about', 'after', 'been', 'on', 'yourself', "shoul
d've", 'doesn', 'that', 'have', 'no', 'my', 'hasn', "haven't", 'had', 'only',
'because', "couldn't", "weren't", 'against', "mightn't", 'can', 'was', "need
n't", 'ain', 'being', 'those', 'from', 'having', 'doing', 'mightn', 'hers',
'haven', "doesn't", 'before', 'while', 'we', 'for', 'whom', 'yourselves', 'he
rself', 'some', 'than', 'to', 'shouldn', 'itself', 'now', 'what', 'which', 'o
ff', 'do', "you're", 'its', 'and', "she's", 'out', 'over', 'when', 'few',
'a', 'don', 'she', 'just', 'couldn', 'him', 't', 'very', 'as', 'll', 'her',
'if', 'me', 'here', 'through', 'under', 'wouldn', 'myself', 'most', "should
n't", 'above', 'isn', "didn't", 'in', 'own', 'not', 'how', 'of', 'you', "woul
dn't", 'both', 'these', "you've", "hasn't", 'other', 'does', 'themselves', 'h
e', 'once', 'each', 'then', 'more', 'but', 'until', "it's", 've', "mustn't",
'yours', 'am', 'did', 'd', 'nor', 'their', 'too', 'has', 's', 'i', 'between',
'your', 'any', 'during', 'ours', "isn't", 'all', 'below', 'down', "you'd", 'h
imself', 'or', 'o', 'at', 'where', 're', 'with', 'should', 'so', 'there', 'wh
o', 'theirs', 'same', "don't", 'it', 'wasn', "won't", 'the', 'weren', 'won',
'are', 'shan', "shan't", 'needn', 'y', "aren't", 'be', 'why', 'didn', 'into',
'up', 'ma', 'aren', 'by', 'his', 'our', "you'll", 'them', 'is', "wasn't", 'we
re', 'will', 'such', 'they', 'this', 'm', 'hadn', 'again', 'ourselves', "had
n't", 'further'}
```

In [5]:

```
from nltk.corpus import stopwords
print(set(stopwords.words('german')))
```

```
{'an', 'damit', 'bin', 'haben', 'solchem', 'anderes', 'deinem', 'welcher', 'w
elche', 'also', 'gewesen', 'jenem', 'alle', 'soll', 'zu', 'einem', 'zum', 'un
d', 'wieder', 'habe', 'dann', 'dieselben', 'warst', 'solchen', 'der', 'alle
m', 'was', 'daß', 'während', 'von', 'dazu', 'das', 'hier', 'dem', 'eures', 'v
iel', 'einen', 'derselben', 'seine', 'sollte', 'aus', 'einiges', 'ihn', 'sei
n', 'derer', 'dir', 'eurer', 'meinen', 'seines', 'ist', 'eurem', 'als', 'ande
rn', 'jenes', 'mancher', 'solcher', 'dort', 'jedem', 'anderer', 'im', 'meine
m', 'derselbe', 'diesem', 'eine', 'andere', 'doch', 'indem', 'zwar', 'einig',
'dies', 'manches', 'kann', 'auf', 'ander', 'ich', 'keiner', 'eines', 'man',
'allen', 'da', 'sondern', 'denn', 'wir', 'unsere', 'ihrer', 'nichts', 'weil',
'euren', 'manchem', 'denselben', 'sehr', 'ihrem', 'einer', 'meines', 'in', 'j
edes', 'musste', 'sonst', 'dieselbe', 'hinter', 'den', 'wo', 'jeder', 'ihne
n', 'jeden', 'euer', 'unter', 'wollen', 'sich', 'einigem', 'welches', 'jede',
'ohne', 'einigen', 'welchem', 'können', 'gegen', 'etwas', 'jenen', 'manche',
'unseren', 'diesen', 'deiner', 'meiner', 'anderen', 'die', 'ins', 'machen',
'werde', 'jener', 'anderr', 'wie', 'bei', 'würde', 'mit', 'würden', 'einmal',
'am', 'mein', 'zur', 'ihm', 'seinen', 'dein', 'desselben', 'muss', 'wollte',
'deinen', 'bist', 'einiger', 'ihres', 'bis', 'deines', 'hatte', 'seiner', 'si
e', 'sind', 'kein', 'werden', 'jetzt', 'könnte', 'dasselbe', 'einige', 'es',
'demselben', 'oder', 'diese', 'hin', 'mich', 'keinen', 'jene', 'vom', 'auch',
'nun', 'dieser', 'um', 'dieses', 'solches', 'keinem', 'so', 'uns', 'seinem',
'hatten', 'zwischen', 'aber', 'anderm', 'mir', 'ihre', 'solche', 'weiter', 'd
ich', 'keines', 'unseres', 'unserem', 'anderem', 'vor', 'nach', 'deine', 'ha
b', 'war', 'unser', 'noch', 'selbst', 'euch', 'ob', 'keine', 'welchen', 'mein
e', 'dessen', 'ihr', 'wenn', 'ihren', 'aller', 'manchen', 'über', 'alles', 'd
u', 'waren', 'er', 'für', 'hat', 'wird', 'eure', 'nur', 'des', 'will', 'weg',
'anders', 'wirst', 'nicht', 'ein', 'durch'}
```

In [2]:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

example_sent = ("This is a sample sentence, showing off the stop words filtrat
ion")

stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence
print(filtered_sentence)
```

```
['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration']
```

In [4]:

```
# Alternative to list comprehension approach

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

example_sent = ("In diesem Satz, so wie bei uns, sind die Beispielwörter")

stop_words = set(stopwords.words('german'))
word_tokens = word_tokenize(example_sent)
filtered_sentence2 = []
for w in word_tokens:
    if w not in stop_words:
        filtered_sentence2.append(w)

filtered_sentence2
print(filtered_sentence2)
```

```
['In', 'Satz', ',', ',', 'Beispielwörter']
```