In [1]:
```
import spacy
```

In [2]:
```
nlp = spacy.load('en_core_web_sm')
```

In [3]:
```
# create a simple doc object
doc = nlp(u"The quick brown fox jumped over the lazy dog's back.")
```

In [4]:
```
# Print the document text
print(doc.text)
```

```
The quick brown fox jumped over the lazy dog's back.
```

In [5]:
```
# To get a specific word or token
print(doc[4])
```

```
jumped
```

In [6]:
```
print(doc[4].text)
```

```
jumped
```

In [7]:
```
# Coarse POS tags
print(doc[4].pos_)
```

```
VERB
```

In [8]:
```
# Fine grained tag
print(doc[4].tag_)
```

```
VBD
```

In [9]:
```
# To get numerical ID of the coarse POS -- note no underscore
print(doc[4].pos)
```

```
99
```

In [10]:
```
# To get numberical ID for the fine grained POS -- note no underscore
print(doc[4].tag)
```

```
17109001835818727656
```

In [12]:
```
# To get info on the token such as text, coarse POS,
# fine grained POS (tag) and Spacy's explanation for POS

for token in doc:
    print(f'{token.text:{10}} {token.pos_:{10}} {token.tag_:{10}} \
    {spacy.explain(token.tag_)}')
```

```
The        DET        DT              determiner
quick      ADJ        JJ              adjective
brown      ADJ        JJ              adjective
fox        NOUN       NN              noun, singular or mass
jumped     VERB       VBD             verb, past tense
over       ADP        IN              conjunction, subordinating or p
reposition
the        DET        DT              determiner
lazy       ADJ        JJ              adjective
dog        NOUN       NN              noun, singular or mass
's         PART       POS             possessive ending
back       NOUN       NN              noun, singular or mass
.          PUNCT      .               punctuation mark, sentence clos
er
```

In [13]:
```
doc1 = nlp(u'I read books on NLP.')
```

In [14]:
```
word = doc1[1]
```

In [15]:
```
word.text
```

Out[15]:
```
'read'
```

In [16]:
```
token = word
print(f'{token.text:{10}} {token.pos_:{10}} {token.tag_:{10}} \
    {spacy.explain(token.tag_)}')
```

```
read       VERB       VBP             verb, non-3rd person singular p
resent
```

In [17]:
```
doc2 = nlp(u'I read a book on NLP.')
word = doc2[1]
```

In [18]:
```
word.text
```

Out[18]:
```
'read'
```

In [19]:
```
token = word
print(f'{token.text:{10}} {token.pos_:{10}} {token.tag_:{10}} {spacy.explain(token.tag_)}')
```

```
read       VERB       VBD        verb, past tense
```

Counting POS tags

In [20]:
```
doc = nlp(u"The quick brown fox jumped over the lazy dog's back.")
```

In [21]:
```
# Count the frequencies
POS_count = doc.count_by(spacy.attrs.POS)
```

In [22]:
```
print(POS_count)
```

```
{96: 1, 83: 3, 99: 1, 84: 1, 89: 2, 91: 3, 93: 1}
```

In [27]:
```
TAG_counts = doc.count_by(spacy.attrs.TAG)
```

In [28]:
```
print(TAG_counts)
```

```
{15308085513773655218: 3, 12646065887601541794: 1, 17109001835818727
656: 1, 15267657372422890137: 2, 10554686591937588953: 3, 74: 1, 129
2078113972184607: 1}
```

In [26]:
```
POS_count
```

Out[26]:
```
{96: 1, 83: 3, 99: 1, 84: 1, 89: 2, 91: 3, 93: 1}
```

In [29]:
```
doc.vocab[83].text
```

Out[29]:
```
'ADJ'
```

In [30]:
```
doc.vocab[99].text
```

Out[30]:
```
'VERB'
```

In [34]:
```
for k,v in sorted(POS_count.items()):
    print(f'{k}.  {doc.vocab[k].text:{10}}:{v} ')
```

```
83.  ADJ       :3
84.  ADP       :1
89.  DET       :2
91.  NOUN      :3
93.  PART      :1
96.  PUNCT     :1
99.  VERB      :1
```

In [35]:
```
TAG_counts = doc.count_by(spacy.attrs.TAG)

for k, v in sorted(TAG_counts.items()):
    print(f'{k}    . {doc.vocab[k].text}  : {v}')
```

```
74    . POS  : 1
1292078113972184607    . IN  : 1
10554686591937588953    . JJ  : 3
12646065887601541794    . .  : 1
15267657372422890137    . DT  : 2
15308085513773655218    . NN  : 3
17109001835818727656    . VBD  : 1
```

In [36]:
```
# DEP_COUNTS
DEP_counts = doc.count_by(spacy.attrs.DEP)

for k, v in sorted(DEP_counts.items()):
    print(f'{k}    . {doc.vocab[k].text}  : {v}')
```

```
399    . amod  : 3
412    . det  : 2
426    . nsubj  : 1
436    . pobj  : 1
437    . poss  : 1
440    . prep  : 1
442    . punct  : 1
8110129090154140942    . case  : 1
8206900633647566924    . ROOT  : 1
```

In [ ]:
```
```