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

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

In [8]:
```
nlp.pipe_names
```

Out[8]:
```
['tok2vec', 'tagger', 'parser', 'attribute_ruler', 'lemmatizer', 'ner']
```

In [3]:
```
doc = nlp("Elon flew to mars yesterday. He carried biryani masala with him")
```

In [6]:
```
for token in doc:
    print(token,"|", token.pos_, "|", spacy.explain(token.pos_))
```

```
Elon | PROPN | proper noun
flew | VERB | verb
to | ADP | adposition
mars | NOUN | noun
yesterday | NOUN | noun
. | PUNCT | punctuation
He | PRON | pronoun
carried | VERB | verb
biryani | ADJ | adjective
masala | NOUN | noun
with | ADP | adposition
him | PRON | pronoun
```

In [18]:
```
doc = nlp("Wow! Dr. Strange made 265 million $ on the very first day")
for token in doc:
    print(token, "|", token.pos_, "|", spacy.explain(token.pos_), " | ", token.tag

# tags are used for understand form of verb
```

```
Wow | INTJ | interjection  |  UH interjection
! | PUNCT | punctuation  |  . punctuation mark, sentence closer
Dr. | PROPN | proper noun  |  NNP noun, proper singular
Strange | PROPN | proper noun  |  NNP noun, proper singular
made | VERB | verb  |  VBD verb, past tense
265 | NUM | numeral  |  CD cardinal number
million | NUM | numeral  |  CD cardinal number
$ | NUM | numeral  |  CD cardinal number
on | ADP | adposition  |  IN conjunction, subordinating or preposition
the | DET | determiner  |  DT determiner
very | ADV | adverb  |  RB adverb
first | ADJ | adjective  |  JJ adjective (English), other noun-modifier (Chinese)
day | NOUN | noun  |  NN noun, singular or mass
```

In below sentences Spacy figures out the past vs present tense for quit

In [19]:
```
doc = nlp("He quits the job")
doc[1]
```

Out[19]:
```
quits
```

In [22]:
```
doc = nlp("He quits the job")
print(doc[1].text, " | ",doc[1].tag_, " | ",spacy.explain(doc[1].tag_))
```

```
quits  |  VBZ  |  verb, 3rd person singular present
```

In [23]:
```
doc = nlp("he quit the job")

print(doc[1].text, "|", doc[1].tag_, "|", spacy.explain(doc[1].tag_))
```

```
quit | VBD | verb, past tense
```

Removing all SPACE, PUNCT and X token from text

In [25]:
```
earnings_text="""Microsoft Corp. today announced the following results for the quar

·         Revenue was $51.7 billion and increased 20%
·         Operating income was $22.2 billion and increased 24%
·         Net income was $18.8 billion and increased 21%
·         Diluted earnings per share was $2.48 and increased 22%
“Digital technology is the most malleable resource at the world’s disposal to overc
“Solid commercial execution, represented by strong bookings growth driven by long-t
```

In [30]:
```
doc = nlp(earnings_text)

filtered_tokens= []
for token in doc:
    if token.pos not in ["SPACE", "X", "PUNCT"]:
        filtered_tokens.append(token)
```

In [31]:
```
filtered_tokens[:20]
```

Out[31]:
```
[Microsoft,
 Corp.,
 today,
 announced,
 the,
 following,
 results,
 for,
 the,
 quarter,
 ended,
 December,
 31,
 ,,
 2021,
 ,,
 as,
 compared,
 to,
 the]
```

To understand how many nouns are present in tags

In [32]:
```
count = doc.count_by(spacy.attrs.POS)
count
```

Out[32]:
```
{96: 13,
 92: 46,
 100: 24,
 90: 9,
 85: 16,
 93: 16,
 97: 27,
 98: 1,
 84: 20,
 103: 10,
 87: 6,
 99: 5,
 89: 12,
 86: 3,
 94: 3,
 95: 2}
```

In [34]:
```
doc.vocab[96].text
```

Out[34]:
```
'PROPN'
```

Putting into dictionary

In [35]:
```
for k,v in count.items():
    print(doc.vocab[k].text, "|",v)
```

```
PROPN | 13
NOUN | 46
VERB | 24
DET | 9
ADP | 16
NUM | 16
PUNCT | 27
SCONJ | 1
ADJ | 20
SPACE | 10
AUX | 6
SYM | 5
CCONJ | 12
ADV | 3
PART | 3
PRON | 2
```