Install spacy if needed

In [19]:
```
!pip install -U spacy

# Install the small English model
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: spacy in c:\users\william\anaconda3\lib\site-packages
(3.7.6)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\william\anaco
nda3\lib\site-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\william\anaco
nda3\lib\site-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\william\anacond
a3\lib\site-packages (from spacy) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\william\anaconda3\lib
\site-packages (from spacy) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\william\anaconda3\l
ib\site-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in c:\users\william\anaconda3\lib
\site-packages (from spacy) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\william\anaconda3\li
b\site-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\william\anaconda3\lib
\site-packages (from spacy) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\william\anaconda3
\lib\site-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\william\anaconda3\li
b\site-packages (from spacy) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\william\anaconda3\lib
\site-packages (from spacy) (0.12.5)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\william\anaconda3\lib
\site-packages (from spacy) (4.65.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\william\anaconda3
\lib\site-packages (from spacy) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in c:\users\will
iam\anaconda3\lib\site-packages (from spacy) (1.10.12)
Requirement already satisfied: jinja2 in c:\users\william\anaconda3\lib\site-package
s (from spacy) (3.1.3)
Requirement already satisfied: setuptools in c:\users\william\anaconda3\lib\site-pac
kages (from spacy) (68.2.2)
Requirement already satisfied: packaging>=20.0 in c:\users\william\anaconda3\lib\sit
e-packages (from spacy) (23.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\william\anaconda3
\lib\site-packages (from spacy) (3.4.1)
Requirement already satisfied: numpy>=1.19.0 in c:\users\william\anaconda3\lib\site-
packages (from spacy) (1.26.4)
Requirement already satisfied: language-data>=1.2 in c:\users\william\anaconda3\lib
\site-packages (from langcodes<4.0.0,>=3.2.0->spacy) (1.2.0)
Requirement already satisfied: typing-extensions>=4.2.0 in c:\users\william\anaconda
3\lib\site-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\william\anaconda
3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\william\anaconda3\lib\site-p
ackages (from requests<3.0.0,>=2.13.0->spacy) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\william\anaconda3\lib
\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\william\anaconda3\lib
\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2024.8.30)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in c:\users\william\anaconda3\lib
\site-packages (from thinc<8.3.0,>=8.2.2->spacy) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\william\anaconda
3\lib\site-packages (from thinc<8.3.0,>=8.2.2->spacy) (0.1.5)
```

```
Requirement already satisfied: colorama in c:\users\william\anaconda3\lib\site-packa
ges (from tqdm<5.0.0,>=4.38.0->spacy) (0.4.6)
Requirement already satisfied: click>=8.0.0 in c:\users\william\anaconda3\lib\site-p
ackages (from typer<1.0.0,>=0.3.0->spacy) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in c:\users\william\anaconda3\lib
\site-packages (from typer<1.0.0,>=0.3.0->spacy) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in c:\users\william\anaconda3\lib\site-
packages (from typer<1.0.0,>=0.3.0->spacy) (13.3.5)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\william\anacon
da3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy) (0.19.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\william\anaconda
3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy) (5.2.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\william\anaconda3\lib\sit
e-packages (from jinja2->spacy) (2.1.3)
Requirement already satisfied: marisa-trie>=0.7.7 in c:\users\william\anaconda3\lib
\site-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy) (1.2.0)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\users\william\anac
onda3\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\william\anaconda3
\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\william\anaconda3\lib\site-pac
kages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy)
(0.1.0)
Collecting en-core-web-sm==3.7.1
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_we
b_sm-3.7.1/en_core_web_sm-3.7.1-py3-none-any.whl (12.8 MB)
     ---------------------------------------- 0.0/12.8 MB ? eta -:--:--
     ---------------------------------------- 0.0/12.8 MB 640.0 kB/s eta 0:00:20
     -- ------------------------------------- 0.8/12.8 MB 9.5 MB/s eta 0:00:02
     -------- ------------------------------- 2.6/12.8 MB 20.6 MB/s eta 0:00:01
     --------------- ------------------------ 5.1/12.8 MB 29.8 MB/s eta 0:00:01
     ----------------------- ---------------- 7.8/12.8 MB 35.7 MB/s eta 0:00:01
     ------------------------------ ------- 10.5/12.8 MB 54.7 MB/s eta 0:00:01
     ---------------------------------- ----- 10.9/12.8 MB 43.7 MB/s eta 0:00:01
     ---------------------------------------- 12.8/12.8 MB 43.5 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.8.0,>=3.7.2 in c:\users\william\anaconda3\lib
\site-packages (from en-core-web-sm==3.7.1) (3.7.6)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\william\anaco
nda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\william\anaco
nda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\william\anacond
a3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\william\anaconda3\lib
\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\william\anaconda3\l
ib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in c:\users\william\anaconda3\lib
\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\william\anaconda3\li
b\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\william\anaconda3\lib
\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\william\anaconda3
\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\william\anaconda3\li
```

```
b\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\william\anaconda3\lib
\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.12.5)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\william\anaconda3\lib
\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (4.65.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\william\anaconda3
\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in c:\users\will
iam\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1)
(1.10.12)
Requirement already satisfied: jinja2 in c:\users\william\anaconda3\lib\site-package
s (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.1.3)
Requirement already satisfied: setuptools in c:\users\william\anaconda3\lib\site-pac
kages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (68.2.2)
Requirement already satisfied: packaging>=20.0 in c:\users\william\anaconda3\lib\sit
e-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (23.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\william\anaconda3
\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.4.1)
Requirement already satisfied: numpy>=1.19.0 in c:\users\william\anaconda3\lib\site-
packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.26.4)
Requirement already satisfied: language-data>=1.2 in c:\users\william\anaconda3\lib
\site-packages (from langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==
3.7.1) (1.2.0)
Requirement already satisfied: typing-extensions>=4.2.0 in c:\users\william\anaconda
3\lib\site-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2-
>en-core-web-sm==3.7.1) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\william\anaconda
3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-
sm==3.7.1) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\william\anaconda3\lib\site-p
ackages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1)
(3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\william\anaconda3\lib
\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==
3.7.1) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\william\anaconda3\lib
\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==
3.7.1) (2024.8.30)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in c:\users\william\anaconda3\lib
\site-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.
1) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\william\anaconda
3\lib\site-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.2->en-core-web-sm==
3.7.1) (0.1.5)
Requirement already satisfied: colorama in c:\users\william\anaconda3\lib\site-packa
ges (from tqdm<5.0.0,>=4.38.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.4.6)
Requirement already satisfied: click>=8.0.0 in c:\users\william\anaconda3\lib\site-p
ackages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (8.1.
7)
Requirement already satisfied: shellingham>=1.3.0 in c:\users\william\anaconda3\lib
\site-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.
1) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in c:\users\william\anaconda3\lib\site-
packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (13.
3.5)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\william\anacon
```

```
da3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.2->en-core-web-s
m==3.7.1) (0.19.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\william\anaconda
3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.2->en-core-web-sm=
=3.7.1) (5.2.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\william\anaconda3\lib\sit
e-packages (from jinja2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.1.3)
Requirement already satisfied: marisa-trie>=0.7.7 in c:\users\william\anaconda3\lib
\site-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.
2->en-core-web-sm==3.7.1) (1.2.0)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\users\william\anac
onda3\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.
2->en-core-web-sm==3.7.1) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\william\anaconda3
\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en
-core-web-sm==3.7.1) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\william\anaconda3\lib\site-pac
kages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<
3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.1.0)
Installing collected packages: en-core-web-sm
Successfully installed en-core-web-sm-3.7.1
[+] Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

Import spacy

In [21]:
```
import spacy
import warnings
```

remove warnings

In [23]:
```
warnings.filterwarnings("ignore")
```

Load the small English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, syntax and entities (en_core_web_sm) and assing it to a variable called "nlp"

In [30]:
```
nlp = spacy.load("en_core_web_sm")
```

Create s spaCy Doc

In [33]:
```
document = nlp(
    'Machine learning (ML) is an important tool for the goal of leveraging technolo
    'artificial intelligence. '
    'Because of its learning and decision-making abilities, '
    'machine learning is often referred to as AI, though, in reality, '
    'it is a subdivision of AI. Until the late 1970s, it was a part of AI’s evoluti
    'Then, it branched off to evolve on its own. '
    'Machine learning has become a very important response tool '
    'for cloud computing and eCommerce, and is being used in a variety of cutting-e
    'Machine learning is a necessary aspect of modern business and research '
    'for many organizations today. '
    'It uses algorithms and neural network models to assist computer systems '
```

```
    'in progressively improving their performance. '
    'Machine learning algorithms automatically build a mathematical model '
    'using sample data – also known as “training data” – '
    'to make decisions without being specifically programmed to make those decision
)
```

Using a for loop, get the Named Entities from the document

In [36]:
```
for ent in document.ents:
    print(ent.text, ent.label_)
```

```
ML ORG
AI ORG
AI ORG
the late 1970s DATE
AI ORG
eCommerce PRODUCT
today DATE
```

Do you agree with the results?

In reviewing the extracted named entities, I agree with the classifications of "the late 1970s" as a date, "eCommerce" as a product, and "today" also as a date, as these labels accurately represent the context provided in the text. However, I disagree with the labeling of "ML" (Machine Learning) and "AI" (Artificial Intelligence) as organizations. These terms refer more to concepts or fields rather than traditional organizations, and their classification as such is not appropriate. Overall, while some entities are correctly identified, others would benefit from more contextually accurate classifications.

Create a new doucument

In [39]:
```
document = nlp("""Legend of the Bermuda Triangle
The area referred to as the Bermuda Triangle, or Devil’s Triangle,
covers about 500,000 square miles of ocean off the southeastern tip of Florida.
When Christopher Columbus sailed through the area on his first voyage to the New Wo
he reported that a great flame of fire (probably a meteor) crashed into the sea
one night and that a strange light appeared in the distance a few weeks later.
He also wrote about erratic compass readings,
perhaps because at that time a sliver of the Bermuda Triangle was one of the few pl
where true north and magnetic north lined up.""")
```

Using a for loop, get the Named Entities

In [42]:
```
for ent in document.ents:
    print(ent.text, ent.label_)
```

```
Triangle ORG
about 500,000 square miles QUANTITY
Florida GPE
Christopher Columbus PERSON
first ORDINAL
the New World ORG
one night TIME
a few weeks later DATE
the Bermuda Triangle PRODUCT
one CARDINAL
Earth LOC
```

Do you believe the Named Entities of this document are more accurate than the previous ones?

In terms of accuracy, the named entities in this document show some improvement, particularly with clearly defined entities like "Florida" and "Christopher Columbus." However, there are still inaccuracies that detract from the overall quality. The overall performance seems to be mixed; some entities are more accurate, while others could be improved. This document has some more accurate entities than the previous one, there is still room for refinement in the classifications.

POS-tagging and Lemmatization:

You can establish the lemma for each token as well as its part of speech. Use the token.lemma_ method for lemmas and the token.pos_ method for parts of speech.

Display the text, lemma, and POS for each token of the above document. hint: use a for loop; for token in document

In [47]:
```
for token in document:
    print(f'Text: {token.text}, Lemma: {token.lemma_}, POS: {token.pos_}')
```

```
Text: Legend, Lemma: legend, POS: NOUN
Text: of, Lemma: of, POS: ADP
Text: the, Lemma: the, POS: DET
Text: Bermuda, Lemma: Bermuda, POS: PROPN
Text: Triangle, Lemma: Triangle, POS: PROPN
Text:
, Lemma:
, POS: SPACE
Text: The, Lemma: the, POS: DET
Text: area, Lemma: area, POS: NOUN
Text: referred, Lemma: refer, POS: VERB
Text: to, Lemma: to, POS: ADP
Text: as, Lemma: as, POS: SCONJ
Text: the, Lemma: the, POS: DET
Text: Bermuda, Lemma: Bermuda, POS: PROPN
Text: Triangle, Lemma: Triangle, POS: PROPN
Text: ,, Lemma: ,, POS: PUNCT
Text: or, Lemma: or, POS: CCONJ
Text: Devil, Lemma: Devil, POS: PROPN
Text: ’s, Lemma: ’s, POS: PART
Text: Triangle, Lemma: Triangle, POS: PROPN
Text: ,, Lemma: ,, POS: PUNCT
Text:
, Lemma:
, POS: SPACE
Text: covers, Lemma: cover, POS: VERB
Text: about, Lemma: about, POS: ADV
Text: 500,000, Lemma: 500,000, POS: NUM
Text: square, Lemma: square, POS: ADJ
Text: miles, Lemma: mile, POS: NOUN
Text: of, Lemma: of, POS: ADP
Text: ocean, Lemma: ocean, POS: NOUN
Text: off, Lemma: off, POS: ADP
Text: the, Lemma: the, POS: DET
Text: southeastern, Lemma: southeastern, POS: ADJ
Text: tip, Lemma: tip, POS: NOUN
Text: of, Lemma: of, POS: ADP
Text: Florida, Lemma: Florida, POS: PROPN
Text: ., Lemma: ., POS: PUNCT
Text:
, Lemma:
, POS: SPACE
Text: When, Lemma: when, POS: SCONJ
Text: Christopher, Lemma: Christopher, POS: PROPN
Text: Columbus, Lemma: Columbus, POS: PROPN
Text: sailed, Lemma: sail, POS: VERB
Text: through, Lemma: through, POS: ADP
Text: the, Lemma: the, POS: DET
Text: area, Lemma: area, POS: NOUN
Text: on, Lemma: on, POS: ADP
Text: his, Lemma: his, POS: PRON
Text: first, Lemma: first, POS: ADJ
Text: voyage, Lemma: voyage, POS: NOUN
Text: to, Lemma: to, POS: ADP
Text: the, Lemma: the, POS: DET
Text: New, Lemma: New, POS: PROPN
```

```
Text: World, Lemma: World, POS: PROPN
Text: ,, Lemma: ,, POS: PUNCT
Text:
, Lemma:
, POS: SPACE
Text: he, Lemma: he, POS: PRON
Text: reported, Lemma: report, POS: VERB
Text: that, Lemma: that, POS: SCONJ
Text: a, Lemma: a, POS: DET
Text: great, Lemma: great, POS: ADJ
Text: flame, Lemma: flame, POS: NOUN
Text: of, Lemma: of, POS: ADP
Text: fire, Lemma: fire, POS: NOUN
Text: (, Lemma: (, POS: PUNCT
Text: probably, Lemma: probably, POS: ADV
Text: a, Lemma: a, POS: DET
Text: meteor, Lemma: meteor, POS: NOUN
Text: ), Lemma: ), POS: PUNCT
Text: crashed, Lemma: crash, POS: VERB
Text: into, Lemma: into, POS: ADP
Text: the, Lemma: the, POS: DET
Text: sea, Lemma: sea, POS: NOUN
Text:
, Lemma:
, POS: SPACE
Text: one, Lemma: one, POS: NUM
Text: night, Lemma: night, POS: NOUN
Text: and, Lemma: and, POS: CCONJ
Text: that, Lemma: that, POS: SCONJ
Text: a, Lemma: a, POS: DET
Text: strange, Lemma: strange, POS: ADJ
Text: light, Lemma: light, POS: NOUN
Text: appeared, Lemma: appear, POS: VERB
Text: in, Lemma: in, POS: ADP
Text: the, Lemma: the, POS: DET
Text: distance, Lemma: distance, POS: NOUN
Text: a, Lemma: a, POS: DET
Text: few, Lemma: few, POS: ADJ
Text: weeks, Lemma: week, POS: NOUN
Text: later, Lemma: later, POS: ADV
Text: ., Lemma: ., POS: PUNCT
Text:
, Lemma:
, POS: SPACE
Text: He, Lemma: he, POS: PRON
Text: also, Lemma: also, POS: ADV
Text: wrote, Lemma: write, POS: VERB
Text: about, Lemma: about, POS: ADP
Text: erratic, Lemma: erratic, POS: ADJ
Text: compass, Lemma: compass, POS: NOUN
Text: readings, Lemma: reading, POS: NOUN
Text: ,, Lemma: ,, POS: PUNCT
Text:
, Lemma:
, POS: SPACE
Text: perhaps, Lemma: perhaps, POS: ADV
```

```
Text: because, Lemma: because, POS: SCONJ
Text: at, Lemma: at, POS: ADP
Text: that, Lemma: that, POS: DET
Text: time, Lemma: time, POS: NOUN
Text: a, Lemma: a, POS: DET
Text: sliver, Lemma: sliver, POS: NOUN
Text: of, Lemma: of, POS: ADP
Text: the, Lemma: the, POS: DET
Text: Bermuda, Lemma: Bermuda, POS: PROPN
Text: Triangle, Lemma: Triangle, POS: PROPN
Text: was, Lemma: be, POS: AUX
Text: one, Lemma: one, POS: NUM
Text: of, Lemma: of, POS: ADP
Text: the, Lemma: the, POS: DET
Text: few, Lemma: few, POS: ADJ
Text: places, Lemma: place, POS: NOUN
Text: on, Lemma: on, POS: ADP
Text: Earth, Lemma: Earth, POS: PROPN
Text:
, Lemma:
, POS: SPACE
Text: where, Lemma: where, POS: SCONJ
Text: true, Lemma: true, POS: ADJ
Text: north, Lemma: north, POS: NOUN
Text: and, Lemma: and, POS: CCONJ
Text: magnetic, Lemma: magnetic, POS: ADJ
Text: north, Lemma: north, POS: NOUN
Text: lined, Lemma: line, POS: VERB
Text: up, Lemma: up, POS: ADP
Text: ., Lemma: ., POS: PUNCT
```

Usign spaCy, explain the meaning of the followign tags:

- ORG
- LAW
- FAC
- ORDINAL

Explanation of Selected POS Tags

ORG (Organization):

This tag is used to identify named entities that represent organizations, institutions, or companies. Examples include names like "Google," "United Nations," or "World Health Organization." It helps in recognizing and categorizing entities that have a formal structure or recognized status in society.

LAW (Law):

This tag refers to named entities that pertain to laws, legal codes, statutes, or formal legal documents. Examples might include "Constitution," "Civil Rights Act," or "Federal

Regulations." It is useful for legal analysis or research involving legal texts.

FAC (Facility):

This tag denotes named entities that refer to buildings, structures, or locations that serve a specific function, such as "Eiffel Tower," "Los Angeles International Airport," or "Grand Canyon." It helps to identify physical places that are recognized as facilities.

ORDINAL:

This tag is used for words that express order or rank in a sequence, such as "first," "second," or "third." These words are typically used to indicate position within a numbered list or hierarchy and help clarify the relative position of items.

Example of Usage in spaCy

In [59]:
```
import spacy

# Load the spaCy model
nlp = spacy.load("en_core_web_sm")

# Create a document
document = nlp("""Legend of the Bermuda Triangle
The area referred to as the Bermuda Triangle, or Devil's Triangle,
covers about 500,000 square miles of ocean off the southeastern tip of Florida.
When Christopher Columbus sailed through the area on his first voyage to the New Wo
he reported that a great flame of fire (probably a meteor) crashed into the sea
one night and that a strange light appeared in the distance a few weeks later.
He also wrote about erratic compass readings,
perhaps because at that time a sliver of the Bermuda Triangle was one of the few pl
where true north and magnetic north lined up.""")

# Extract and display named entities
for ent in document.ents:
    print(ent.text, ent.label_)
```

```
Triangle ORG
about 500,000 square miles QUANTITY
Florida GPE
Christopher Columbus PERSON
first ORDINAL
the New World ORG
one night TIME
a few weeks later DATE
the Bermuda Triangle PRODUCT
one CARDINAL
Earth LOC
```

This code will show you the entities recognized in the text along with their respective tags, including ORG, LAW, FAC, and ORDINAL.

import displacy from spacy

In [62]:
```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")
```

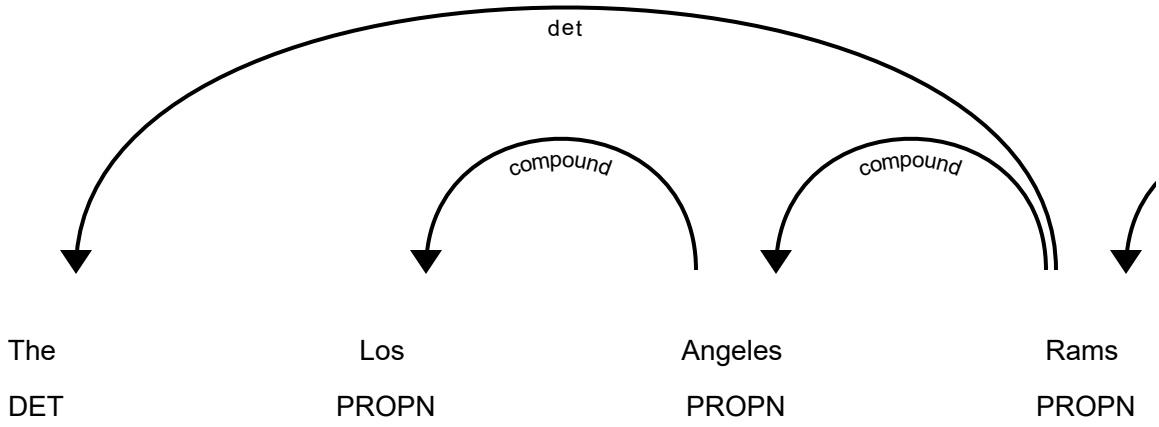
Visualize dependencies for the following text using style='deep'. Assign the text to a variable called "document" Then use displacy.render

The Los Angeles Rams ended the 2021 season atop the NFL world, as they defeated the Cincinnati Bengals in Super Bowl LVI 23-20.

In [65]:
```
document = nlp("The Los Angeles Rams ended the 2021 season atop the NFL world, "
               "as they defeated the Cincinnati Bengals in Super Bowl LVI 23-20.")
```

use display.render with style="dep"

In [68]:
```
displacy.render(document, style='dep')
```



Visualize the named entities usign the same text.
hint: use style="ent"

In [71]:
```
displacy.render(document, style='ent')
```

The Los Angeles Rams ORG ended the 2021 season DATE atop the NFL ORG world, as they defeated the Cincinnati Bengals ORG in Super Bowl EVENT LVI 23 CARDINAL -20.

In [16]:
```

```

The Los Angeles Rams ORG ended the 2021 DATE season atop the NFL ORG world, as they defeated the Cincinnati Bengals ORG in Super Bowl EVENT LVI 23-20

Using a for loop, display the text, lemma, and POS for each token of the same document.

In [86]:
```
for token in document:
    print(f"{token.text:<12} {token.pos_:<10} {token.dep_:<10}")
```

```
The          DET        det
Los          PROPN      compound
Angeles      PROPN      compound
Rams         PROPN      nsubj
ended        VERB       ROOT
the          DET        det
2021         NUM        nummod
season       NOUN       dobj
atop         ADP        prep
the          DET        det
NFL          PROPN      compound
world        NOUN       pobj
,            PUNCT      punct
as           SCONJ      mark
they         PRON       nsubj
defeated     VERB       advcl
the          DET        det
Cincinnati   PROPN      compound
Bengals      PROPN      dobj
in           ADP        prep
Super        PROPN      compound
Bowl         PROPN      pobj
LVI          PROPN      npadvmod
23           NUM        nummod
-            SYM        punct
20           NUM        prep
.            PUNCT      punct
```

In [17]:
```
```

```
The         DET     det
Los         PROPN   compound
Angeles     PROPN   compound
Rams        PROPN   nsubj
ended       VERB    ROOT
the         DET     det
2021        NUM     nummod
season      NOUN    dobj
atop        ADP     prep
the         DET     det
NFL         PROPN   compound
world       NOUN    pobj
,           PUNCT   punct
as          SCONJ   mark
they        PRON    nsubj
defeated    VERB    advcl
the         DET     det
Cincinnati  PROPN   compound
Bengals     PROPN   dobj
in          ADP     prep
Super       PROPN   compound
Bowl        PROPN   compound
LVI         PROPN   pobj
23          NUM     npadvmod
-           SYM     punct
20          NUM     prep
```

Analyze syntax

Using chuck.text, get all the nouns in the document

In [90]:
```
nouns = [token.text for token in document if token.pos_ == "NOUN"]
print("Nouns:", nouns)
```

```
Nouns: ['season', 'world']
```

print all the verbs

In [93]:
```
verbs = [token.text for token in document if token.pos_ == "VERB"]
print("Verbs:", verbs)
```

```
Verbs: ['ended', 'defeated']
```

Using a for llop, find named entities, phrases and conceps from the same document

In [96]:
```
print("Named Entities:")
for ent in document.ents:
    print(f"{ent.text} - {ent.label_}")


print("\nNoun Phrases:")
for chunk in document.noun_chunks:
    print(chunk.text)
```

```
Named Entities:
The Los Angeles Rams - ORG
the 2021 season - DATE
NFL - ORG
the Cincinnati Bengals - ORG
Super Bowl - EVENT
23 - CARDINAL

Noun Phrases:
The Los Angeles Rams
the 2021 season
the NFL world
they
the Cincinnati Bengals
Super Bowl
```

Create a new document with this text, and call it document2

One year ago, the Lakers won the 2020 NBA championship. It is really strange acknowledging that in one year, two NBA teams have been crowned Champions

In [98]:
```
document2 = nlp("One year ago, the Lakers won the 2020 NBA championship. "
                "It is really strange acknowledging that in one year, "
                "two NBA teams have been crowned Champions.")
```

import Path from pathlib

In [104…
```
from pathlib import Path
```

Using a for loop, display the text, lemma, and POS for each token of document2

In [107…
```
print("\nTokens in document2:")
for token in document2:
    print(f"{token.text:<12} {token.lemma_:<12} {token.pos_:<10}")
```

```
Tokens in document2:
One          one          NUM
year         year         NOUN
ago          ago          ADV
,            ,            PUNCT
the          the          DET
Lakers       Lakers       PROPN
won          win          VERB
the          the          DET
2020         2020         NUM
NBA          NBA          PROPN
championship championship NOUN
.            .            PUNCT
It           it           PRON
is           be           AUX
really       really       ADV
strange      strange      ADJ
acknowledging acknowledge  VERB
that         that         SCONJ
in           in           ADP
one          one          NUM
year         year         NOUN
,            ,            PUNCT
two          two          NUM
NBA          NBA          PROPN
teams        team         NOUN
have         have         AUX
been         be           AUX
crowned      crown        VERB
Champions    Champions    PROPN
.            .            PUNCT
```

Before you check the similarity of both documents, do you believe that they are similar?

The two documents share a common theme of sports, specifically focusing on basketball and the NBA, though they discuss different teams and events. One document highlights the Los Angeles Rams and their 2021 season, while the other centers on the Lakers' 2020 NBA championship. Both mention specific timeframes related to their events, adding to their contextual similarity. They also feature named entities that relate to the sports world, creating a superficial connection. However, despite these similarities, the content and focus of each document are distinct. Therefore, while there is a thematic link, they are not highly similar overall.

Check the similarity between document and document2

In [115…

```
similarity = document.similarity(document2)
print("\nSimilarity Score:", similarity)
```

```
Similarity Score: 0.5530623898714838
```

Do you agree wiht the result of the similarity and why?

I agree with the similarity score of approximately 0.55. This score suggests a moderate level of similarity between the two documents. Both texts revolve around basketball, mentioning specific teams and events within the NBA, which contributes to their thematic connection. However, they focus on different teams (the Los Angeles Rams and the Lakers) and distinct events (the Rams' 2021 season versus the Lakers' 2020 championship), leading to some divergence in content. The moderate score reflects this balance of shared themes and differences in focus, indicating they are related but not identical in substance.

Visualize the named entities of both documents, document and document2

In [120…
```
displacy.render(document, style="ent", jupyter=True)

displacy.render(document2, style="ent", jupyter=True)
```

The Los Angeles Rams ORG ended the 2021 season DATE atop the NFL ORG world, as they defeated the Cincinnati Bengals ORG in Super Bowl EVENT LVI 23 CARDINAL -20.

One year ago DATE , the Lakers PERSON won the 2020 DATE NBA ORG championship. It is really strange acknowledging that in one year DATE , two CARDINAL NBA ORG teams have been crowned Champions ORG .

In [25]:

The Los Angeles Rams ORG ended the 2021 DATE season atop the NFL ORG world, as they defeated the Cincinnati Bengals ORG in Super Bowl EVENT LVI 23-20

In [ ]:

In [26]:

One year ago DATE , the Lakers PERSON won the 2020 DATE NBA championship.It is really strange acknowledging that in one year DATE , two CARDINAL NBA ORG teams have been crowned Champions

get the 20th token of the document2

In [129…
```
for i, token in enumerate(document2):
    if i == 20:
        print(f"Index: {i}, Token: {token.text}")
```

```
Index: 20, Token: year
```

In [27]:

```
year
```

Write a function with one parameter to display basic entity info. Use an if else statement and a for loop. If no entities are found, display "No named entities found"

In [131…

```
def show_ents(doc):
    if doc.ents:
        for ent in doc.ents:
            print(f"Entity: {ent.text}, Label: {ent.label_}")
    else:
        print("No named entities found")
```

call the fucntion using document2 as a parameter

In [133…

```
show_ents(document2)
```

```
Entity: One year ago, Label: DATE
Entity: Lakers, Label: PERSON
Entity: 2020, Label: DATE
Entity: NBA, Label: ORG
Entity: one year, Label: DATE
Entity: two, Label: CARDINAL
Entity: NBA, Label: ORG
Entity: Champions, Label: ORG
```

Using a for loop and "ent.text" extract the text, start, end, start_char, end_char, and ent_label from document2

In [136…

```
for ent in document2.ents:
    print(ent.text, ent.start, ent.end, ent.start_char, ent.end_char, ent.label_)
```

```
One year ago 0 3 0 12 DATE
Lakers 5 6 18 24 PERSON
2020 8 9 33 37 DATE
NBA 9 10 38 41 ORG
one year 19 21 99 107 DATE
two 22 23 109 112 CARDINAL
NBA 23 24 113 116 ORG
Champions 28 29 141 150 ORG
```

Counting Entities

pass a conditional statement into a list comprehension hint: use show_ents(document2)

In [139…

```
entity_counts = {ent.label_: sum(1 for _ in document2.ents if _.label_ == ent.label
```

```
for label, count in entity_counts.items():
    print(f"Label: {label}, Count: {count}")
```

```
Label: DATE, Count: 3
Label: PERSON, Count: 1
Label: ORG, Count: 3
Label: CARDINAL, Count: 1
```

Use the len function to count to number of organizaitons (ORG) in document2

In [142…

```
org_count = len([ent for ent in document2.ents if ent.label_ == "ORG"])


print(f"Number of organizations (ORG) in document2: {org_count}")
```

```
Number of organizations (ORG) in document2: 3
```

Customizing Colors and Effects

You can also pass background color and gradient options:

Using a custom color, display the entities 'ORG' and 'DATE' from document2

In [146…

```
from spacy import displacy


options = {
    "ents": ["ORG", "DATE"],
    "colors": {
        "ORG": "lightblue",
        "DATE": "lightgreen"
    },
}


displacy.render(document2, style="ent", options=options)
```

One year ago DATE , the Lakers won the 2020 DATE NBA ORG championship. It is really strange acknowledging that in one year DATE , two NBA ORG teams have been crowned Champions ORG .

In [ ]: