

```
!pip install -U spacy
!python -m spacy download en_core_web_md
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.20.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy<=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.1)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (1)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->sp)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy) (8)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->s)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0)
Collecting en-core-web-md==3.8.0
```

Using cached [https://github.com/explosion/spacy-models/releases/download/en\\_core\\_web\\_md-3.8.0/en\\_core\\_web\\_md-3.8.0-py3-none-any.whl](https://github.com/explosion/spacy-models/releases/download/en_core_web_md-3.8.0/en_core_web_md-3.8.0-py3-none-any.whl)  
 ✓ Download and installation successful  
 You can now load the package via `spacy.load('en_core_web_md')`  
 ⚠ Restart to reload dependencies  
 If you are in a Jupyter or Colab notebook, you may need to restart Python in order to load all the package's dependencies. You can do this by selecting the 'Restart kernel' or 'Restart runtime' option.

```
import spacy
from spacy import displacy

# Tải mô hình
nlp = spacy.load("en_core_web_md")

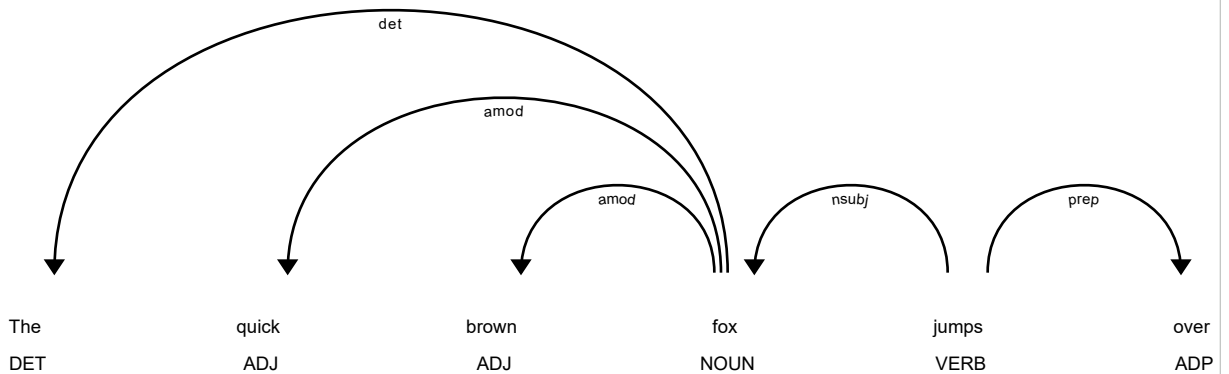
# Ví dụ câu
text = "The quick brown fox jumps over the lazy dog."
doc = nlp(text)

# In từng token
for token in doc:
    print(token.text, "|", token.dep_, "|", token.head.text)
```

```
The | det | fox
quick | amod | fox
brown | amod | fox
fox | nsubj | jumps
jumps | ROOT | jumps
over | prep | jumps
the | det | dog
lazy | amod | dog
dog | pobj | over
. | punct | jumps
```

```
displacy.serve(doc, style="dep")
```

```
/usr/local/lib/python3.12/dist-packages/spacy/displacy/__init__.py:108: UserWarning: [W011] It looks like you're calling displac
warnings.warn(Warnings.W011)
```



Using the 'dep' visualizer  
 Serving on <http://0.0.0.0:5000> ...  
 Shutting down server on port 5000.

Câu 1: Từ nào là ROOT của câu?

ROOT là: jumps

Trong câu này, jumps là động từ chính, đứng làm trung tâm của cả câu.

Câu 2: jumps có những dependent nào? Và quan hệ là gì?

Các token phụ thuộc trực tiếp vào jumps:

Dependent Dependency label Giải thích fox nsubj chủ ngữ của hành động "jumps"

jumps có 2 dependent:

Dependent	Dependency label	Giải thích
fox	nsubj	chủ ngữ của hành động "jumps"
over	prep	giới từ nối vào động từ

Câu 3: fox là head của những từ nào?

Các token có head là fox:

Token Dependency Giải thích The det mạo từ bổ nghĩa cho fox quick amod tính từ bổ nghĩa brown amod tính từ bổ nghĩa

fox là head của: The, quick, brown

```
text = "Apple is looking at buying U.K. startup for $1 billion"
doc = nlp(text)

print(f"{'TEXT':<12} | {'DEP':<10} | {'HEAD TEXT':<12} | {'HEAD POS':<8} | CHILDREN")
print("-" * 80)

for token in doc:
    children = [child.text for child in token.children]
    print(f"{token.text:<12} | {token.dep_:<10} | {token.head.text:<12} | {token.head.pos_:<8} | {children}")
```

TEXT	DEP	HEAD TEXT	HEAD POS	CHILDREN
Apple	nsubj	looking	VERB	[]
is	aux	looking	VERB	[]
looking	ROOT	looking	VERB	['Apple', 'is', 'at']

at	prep	looking	VERB	['buying']
buying	pcomp	at	ADP	['startup']
U.K.	compound	startup	NOUN	[]
startup	dobj	buying	VERB	['U.K.', 'for']
for	prep	startup	NOUN	['billion']
\$	quantmod	billion	NUM	[]
1	compound	billion	NUM	[]
billion	pobj	for	ADP	['\$', '1']

```
text = "The cat chased the mouse and the dog watched them."
doc = nlp(text)

for token in doc:
    if token.pos_ == "VERB":
        verb = token.text
        subject = None
        obj = None

        # duyệt con của động từ
        for child in token.children:
            if child.dep_ == "nsubj":
                subject = child.text
            if child.dep_ == "dobj":
                obj = child.text

        if subject and obj:
            print(f"Found Triplet: ({subject}, {verb}, {obj})")
```

```
Found Triplet: (cat, chased, mouse)
Found Triplet: (dog, watched, them)
```

```
text = "The big, fluffy white cat is sleeping on the warm mat."
doc = nlp(text)

for token in doc:
    if token.pos_ == "NOUN":
        adjectives = [child.text for child in token.children if child.dep_ == "amod"]
        if adjectives:
            print(f"Danh từ '{token.text}' được bổ nghĩa bởi các tính từ: {adjectives}")
```

```
Danh từ 'cat' được bổ nghĩa bởi các tính từ: ['big', 'fluffy', 'white']
Danh từ 'mat' được bổ nghĩa bởi các tính từ: ['warm']
```

```
def find_main_verb(doc):
    for token in doc:
        if token.dep_ == "ROOT":
            return token
    return None

text = "The boy is running in the park."
doc = nlp(text)

print("Main verb (ROOT):", find_main_verb(doc))
```

```
Main verb (ROOT): running
```

```
def extract_noun_chunk(token):
    modifiers = [
        child.text for child in token.children
        if child.dep_ in ("det", "amod", "compound")
    ]
    return " ".join(modifiers + [token.text])

def custom_noun_chunks(doc):
    chunks = []
    for token in doc:
        if token.pos_ == "NOUN":
            chunks.append(extract_noun_chunk(token))
    return chunks

text = "The big fluffy cat chased a small mouse."
doc = nlp(text)
```

```
print("Custom noun chunks:", custom_noun_chunks(doc))
```

```
print(custom_noun_chunks, custom_noun_chunks.doc)
```

Custom noun chunks: ['The big fluffy cat', 'a small mouse']

```
def get_path_to_root(token):
    path = [token]
    while token.dep_ != "ROOT":
        token = token.head
        path.append(token)
    return path

text = "Apple is looking at buying a startup."
doc = nlp(text)

token = doc[3]
path = get_path_to_root(token)

print("Path to ROOT:", " -> ".join([t.text for t in path]))
```

Path to ROOT: at -> looking