Exercises:

Question: 1

```
Import spacy
nlp = spacy.load("en-core-web-sm")
doc = nlp("Welcome all of you for this NLP with spacy
                    course").
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

Ques: 3

```
my_text = ('Rajkumar kannan is a ML developer currently'
        'working for a London-based Edtech'
        'Company. He is interested in learning'
        'Natural language processing'.
        'He keeps organizing local Python meetups'
        'and several internal talks at his workplace.')
```

Question: 4

```
doc = nlp(my-text)
for token in doc:
    print(token.text, token.lemma_, token.pos_; token.tag_)
          token.dep_, token.shape_, token.is_alpha, token
          is_stop)
```

Question: 5

```
Import re
Import spacy
from spacy.tokenizer import Tokenizer.
from spacy.util import compile_prefix_regex, compile_infix_regex.
```

Natural Language Processing Lab
Lab15. Text Processing using SpaCy

In this lab session, you will install spacy, displacy and textacy and perform text processing. After completing this lab, you will perform the following NLP tasks.

- Sentence Detection
- Tokenization in spaCy
- Stop Words Removal
- Lemmatization
- Part of Speech Tagging
- Visualization using displaCy
- Rule-Based Matching Using spaCy
- Dependency Parsing Using spaCy
- Navigating the Tree and Subtree
- Shallow Parsing
- Named Entity Recognition

EXERCISES

Question 1. Print the tokens of the string, "welcome all of you for this NLP with spacy course"

Question 2. Create a text file that contains the above string, open that file and print the tokens

Question 3. Consider the following sentences and print each sentence in one line

```
my_text = ('Rajkumar Kannan is a ML developer currently'
...               ' working for a London-based Edtech'
...               ' company. He is interested in learning'
...               ' Natural Language Processing.'
...               ' He keeps organizing local Python meetups'
...               ' and several internal talks at his workplace.')
```

Question 4. For the list of strings from **my_text**, print the following for each token:

```
token, token.idx, token.text_with_ws,
token.is_alpha, token.is_punct, token.is_space,
token.shape_, token.is_stop
```

Question 5. Detect and print hyphenated words from **my_text**. For example, `London-based`.

Question 6. Print all stop words defined in SpaCy

Question 7. Remove all stop words and print the rest of tokens from, **my_text**

Question 8. Print all lemma from **my_text**

Question 9. Perform Part of Speech Tagging on my_text and print the following tag informations

```
token, token.tag_, token.pos_, spacy.explain(token.tag_)
```

Question 10. How many NOUN and ADJ are there in **my_text**?. Print them and its count.

Question 11. Visualize POS tags of a sentence, **my_text**, using displaCy

Question 12. Extract and print First Name and Last Name from **my_text** using Matcher.

Question 13. Print the dependency parse tag values for the text, "`Rajkumar is learning piano`". Also, display dependency parse tree using displaCy.

```
Compile - suffix - regex.

def custom_tokenizer(nlp):
    infix_re = re.compile(r'''[.\,\?\:\;\...\'\'\'\"\"\"\"\-]''')
    prefix_re = compile_prefix_regex(nlp.Defaults.prefixes)
    suffix_re = compile_suffix_regex(nlp.Defaults.suffixes)

    return tokenizer(nlp.vocab, prefix_search=prefix_re.search,
                                suffix_search=suffix_re.search,
                                infix_finditer=infix_re.finditer,
                                token_match=None)

nlp=spacy.load('en')
nlp.tokenizer=custom_tokenizer(nlp)
doc=nlp(my_text)
[token.text for token in doc]
```

Question:6

```
print(nlp.Defaults.stop_words)
```

Question:7

```
all_stopwords = nlp.Defaults.stop_words
[token.text for token in doc if not token.text in all_stopwords]
```

Question:8:

```
for token in doc:
    print(token, token.lemma_)
```

Question:9

```
doc = nlp(my_text)
for token in doc:
    print(token.text, token.pos_, token.tag_, spacy.explain(token.tag_))
```

Scanned with CamScanner

Question 14. Consider the following string.

```
d_text = ('Sam Peter is a Python developer currently working for a London-based
Fintech company')
```

a. Print the children of `developer`
b. Print the previous neighboring node of `developer`
c. Print the next neighboring node of `developer`
d. Print the all tokens on the left of `developer`
e. Print the tokens on the right of `developer`
f. Print the Print subtree of `developer`

Question 15. Print all Noun Phrases in the text

```
conference_text = ('There is a developer conference happening on 21 July 2020 in
New Delhi.')
```

Question 16. Print all Verb Phrases in the text (you need to install textacy)

```
about_talk_text = ('The talk will introduce reader about Use'
...                ' cases of Natural Language Processing in'
...                ' Fintech')
```

Question 17. Print all Named Entities in the text

```
piano_class_text = ('Great Piano Academy is situated'
...     ' in Mayfair or the City of London and has'
...     ' world-class piano instructors.')
```

You will have to print the values such as

```
ent.text, ent.start_char, ent.end_char, ent.label_, spacy.explain(ent.label_)
```

Question:10

```
nouns = []
for tokens in doc:
    if token.pos_ ==' NOUN':
        nouns.append(token)
print(len(nouns), nouns)

adjectives = []
for tokens in doc:
    if token.pos_ == "ADJ':
        adjectives.append(token)
print(len(adjectives), adjectives)
```

Question:- 11

```
from spacy import disblay.
display. render (doc, style='dep', jupyter=True)
```

Question:- 12

```
from spacy. matcher. import Matcher.
from spacy.tokens import span.
matcher= Matcher (nlp.vocab).
matcher. add ("PERSON", [[{"lower": "rajkumar"},
                         {"lower": "kannan"}]])
matches = matcher(doc)
for match_id, start, end in matches:
    span= span(doc, start, end, label=match_id)
    print(span. text, span. label_)
```

Question:- 13

```
doc = nlp (u'Rajkumar is learning piano').
for token in doc:
    print (token. text, token. dep_)
display. render(doc, style='dep', jupyter=True)
```

NOTES

Question:14

(a) d_text = 'Sam peter is a Python developer currently working for a London-based. Fintech Company'

doc = nlp(d_text)

[t.text for t in doc[5].children].

(b) print(doc[5].nbor(-1))

(c) print(doc[5].nbor(1))

(d) [t.text for t in doc[5].lefts].

(e) for the tokens on the right developer.

[t.text for t in doc[5].rights].

(f) [t.text for t in doc[5].subtree]

Question:15

conference_text = ('There is a developer conference happening on 21 july 2020 in New Delhi')

conference_doc = nlp(conference_text)

for chunk in conference_doc.noun_chunks:

print(chunk)

Question: 16

```
import spacy, en-core-web-sm.

import textacy.

# about-talk-text = ('The talk will introduce reader about
                                                          use'
                  # 'Cases of Natural Language processing in'
                  # 'Fintech').

# pattern = r'((<VERB>)? <ADV> * <VERB> +)'.

# about-talk-doc = textacy.make-spacy-doc(about-talk-text,
                                   lang='en-core-web-sm').

# verb-phrases = textacy.extract.pos-regex-matches(about-talk-doc,
                                                   pattern).

      for chunk in verb-phrases:
         # print(chunk.text)
   for chunk in about-talk-doc.noun-chunks:.
         print(chunk)
```

Question: 17

```
piano-class-text = ('Great piano Academy is situated'
               'in Mayfair or the City of London and has'
               'World-Class piano instructors.")

piano-class-doc = nlp(piano-class-text)
   for ent in piano-class-doc.ents:
print(ent.text, ent.start-char, ent.end-char, ent.label-,
         spacy.explain(ent.label-))
```

Scanned with CamScanner