```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Vedantika\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Vedantika\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
from nltk.chunk import RegexpParser
from nltk.tokenize import word_tokenize


sentence = "Educative Answers is a free web encyclopedia written by devs for devs."
```

Tokenization

```
tokens = word_tokenize(sentence)


tokens
```

```
['Educative',
 'Answers',
 'is',
 'a',
 'free',
 'web',
 'encyclopedia',
 'written',
 'by',
 'devs',
 'for',
 'devs',
 '.']
```

POS tagging

```
pos_tags = nltk.pos_tag(tokens)


pos_tags
```

```
[('Educative', 'JJ'),
 ('Answers', 'NNPS'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('free', 'JJ'),
 ('web', 'NN'),
 ('encyclopedia', 'NN'),
 ('written', 'VBN'),
 ('by', 'IN'),
 ('devs', 'NN'),
 ('for', 'IN'),
 ('devs', 'NN'),
 ('.', '.')]
```

Chunking patterns

```
chunk_patterns = r"""
    NP: {<DT>?<JJ>*<NN>}  # Chunk noun phrases
    VP: {<VB.*><NP|PP>}  # Chunk verb phrases
"""


chunk_patterns
```

```
'\n    NP: {<DT>?<JJ>*<NN>}  # Chunk noun phrases\n    VP: {<VB.*><NP|PP>}  # Chunk verb phrases\n'
```

Create a chunk parser

```
chunk_parser = RegexpParser(chunk_patterns)
```

```
chunk_parser
```

```
     <chunk.RegexpParser with 2 stages>
```

Perform chunking

```
result = chunk_parser.parse(pos_tags)
```

```
print(result)
```

```
     (S
       Educative/JJ
       Answers/NNPS
       (VP is/VBZ (NP a/DT free/JJ web/NN))
       (NP encyclopedia/NN)
       written/VBN
       by/IN
       (NP devs/NN)
       for/IN
       (NP devs/NN)
       ./.)
```