

```

import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger_eng')
nltk.download("punkt_tab")

# Import the necessary libraries for parsing.
from nltk import pos_tag, word_tokenize
from nltk.tree import Tree

# Define the grammar for parsing.
grammar = r"""
NP: {<DT|JJ|NN.*>+}      # Noun phrase
VP: {<VB.*><NP|PP>+$}     # Verb phrase
PP: {<IN><NP>}            # Prepositional phrase
"""

# Create a parser based on the grammar.
cp = nltk.RegexpParser(grammar)

# Tokenize and tag the sentence.
sentence = "The man saw the dog with the telescope"
tokens = word_tokenize(sentence)
tagged_tokens = pos_tag(tokens)

# Parse the sentence and create the tree.
tree = cp.parse(tagged_tokens)

# Print the tree.
print(tree)

```

```

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]   /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
(S
  (NP The/DT man/NN)
  saw/VBD
  (NP the/DT dog/NN)
  (PP with/IN (NP the/DT telescope/NN)))

```