```
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 internet gives everyone a voice"
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)
```

```
(S
  (NP The/DT internet/NN)
  (VP gives/VBZ (NP everyone/NN a/DT voice/NN)))
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```