


BASIC TEXT ANALYSIS IN NLP

```
text="the quick brown fox jumps over the lazy dog"  
print(text)
```

 the quick brown fox jumps over the lazy dog

```
#INDEX OF WORD "QUICK"  
def find_word(word,text):  
    return word in text  
find_word("quick",text)
```

 True

```
#INDEX OF WORD "JUMP"  
def find_word(word,text):  
    return word in text  
find_word("jump",text)
```

 True


```
#POSITION OF WORD "QUICK"  
def find_word1(word,text):  
    return text.index(word)  
find_word1("quick",text)
```

 True


```
#RANK OF WORD  
find_word1("quick",text.split())
```

 1


```
#find word "xyz"  
def find_word(word,text):  
    return word in text  
find_word("XYZ",text)
```

 False

```
#FIND THIRD WORD  
def third_word(text):  
    return text.split()[2]  
third_word(text)
```

 'brown'

```
#CONCATENATE FIRST AND LAST WORD  
def conc_word(text):  
    words=text.split()  
    first_word=words[0]  
    last_word=words[-1]  
    return first_word+last_word  
conc_word(text)
```

 'thedog'

```
#CONCATENATE third AND LAST second WORD
```

```
def conc_word(text):
    words=text.split()
    first_word=words[3]
    last_word=words[-2]
    return first_word+last_word
conc_word(text)
```

```
⇒ 'foxlazy'
```

```
# PRINT WORDS AT EVEN POSITION
```

```
def even_word(text):
    words=text.split()
    for i in range(len(words)):
        if i%2==0:
            print(words[i])
even_word(text)
```

```
⇒ the
    brown
    jumps
    the
    dog
```

```
# PRINT WORDS AT ODD POSITION
```

```
def even_word(text):
    words=text.split()
    for i in range(len(words)):
        if i%2!=0:
            print(words[i])
even_word(text)
```

```
⇒ quick
    fox
    over
    lazy
```

```
# PRINT WORDS SKIPPING 2 WORDS
```

```
def even_word(text):
    words=text.split()
    for i in range(0,len(words),2):
        print(words[i])
even_word(text)
```

```
⇒ the
    brown
    jumps
    the
    dog
```

```
#LAST 3 LETTERS OF A SSENTENCE
```

```
print(text[-3:])
```

```
⇒ dog
```

```
print(text[::-1])
```

```
⇒ god yzal eht revo spmuj xof nworb kciuq eht
```

```
def find_rev(text):
    words = text.split()
    return ' '.join([word[::-1] for word in words])
find_rev(text)
```

```
#split,reverse,join with a space
```

```
➦ 'eht kciuq nworb xof spmuj revo eht yzal god'
```

```
#TOKENIZATION OF TEXT ANALYSIS USING NLTK
```

```
!pip install nltk
```

```
➦ Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
```

```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab') # Download the punkt_tab resource
text="Natural Language Processing is amazing.It helps to understand"
#SENTENCE AND TOKENIZATION
sen=nltk.sent_tokenize(text)
word=nltk.word_tokenize(text)
print(sen)
print(word)
```

```
➦ [nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt_tab.zip.
['Natural Language Processing is amazing.It helps to understand']
['Natural', 'Language', 'Processing', 'is', 'amazing.It', 'helps', 'to', 'understand']
```

```
#STOP WORD REMOVAL
```

```
# Remove stop words
```

```
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
word=nltk.word_tokenize(text)
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in text if word.lower() not in stop_words]
print(filtered_words)
```

```
➦ ['N', 'u', 'r', 'l', ' ', 'L', 'n', 'g', 'u', 'g', 'e', ' ', 'P', 'r', 'c', 'e', 'n', 'g', ' ', ' ', ' ', 'z']
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

Start coding or [generate](#) with AI.

