

19/01/2024

Name: Roshini B

Reg no.: 21MID0017

```
In [ ]: from nltk.corpus import inaugural
```

```
In [ ]: inaugural.fileids()
```

```
In [ ]: text1='In the quaint town nestled between rolling hills and meandering rivers, life unfolded at a leisurely pace.'
```

```
In [ ]: import nltk
fd=nltk.FreqDist(text1.split())
```

```
In [18]: #number of times the unique words are occurring
fd
```

```
Out[18]: FreqDist({'the': 7, 'with': 5, 'of': 5, 'and': 4, 'a': 3, 'town': 2, 'The': 2, 'to': 2, 'In': 1, 'quaint': 1, ...
})
```

```
In [19]: text2=inaugural.words(fileids='1861-Lincoln.txt')
```

```
In [20]: import nltk
fd1=nltk.FreqDist(text2)
```

```
In [21]: fd1
```

```
Out[21]: FreqDist({'the': 240, ',': 195, 'of': 146, 'to': 132, '.': 110, 'and': 101, 'be': 76, 'in': 72, 'that': 57, 'a':
53, ...})
```

```
In [22]: from nltk.probability import ConditionalFreqDist
cfd=ConditionalFreqDist((len(word),word)for word in text2)
```

```
In [23]: #words with 13 characters in text1
cfd[13]
```

```
Out[23]: FreqDist({'circumstances': 3, 'contemplation': 2, 'revolutionary': 2, 'controversies': 2, 'unsubstantial': 1, 'ju
risprudence': 1, 'hypercritical': 1, 'distinguished': 1, 'Confederation': 1, 'authoritative': 1, ...})
```

```
In [24]: text3=inaugural.words(fileids='2021-Biden.txt')
```

```
In [25]: import nltk
fd2=nltk.FreqDist(text3)
```

```
In [26]: fd2
```

```
Out[26]: FreqDist({'': 230, '.': 155, 'the': 109, 'of': 82, 'and': 80, 'we': 70, 'to': 63, 'a': 45, 'I': 44, '"': 43, ...
})
```

```
In [27]: from nltk.probability import ConditionalFreqDist
cfd1=ConditionalFreqDist((len(word),word)for word in text3)
```

```
In [28]: cfd1[13]
```

```
Out[28]: FreqDist({'possibilities': 2, 'distinguished': 1})
```

```
In [31]: pip install jieba
```

```
Collecting jieba
  Downloading jieba-0.42.1.tar.gz (19.2 MB)
Building wheels for collected packages: jieba
  Building wheel for jieba (setup.py): started
  Building wheel for jieba (setup.py): finished with status 'done'
  Created wheel for jieba: filename=jieba-0.42.1-py3-none-any.whl size=19314477 sha256=fd260ab28caaf1776a9200a33f6e47d14d5bd17ff688f9260f562e13f584e232
  Stored in directory: c:\users\roshini\appdata\local\pip\cache\wheels\ca\38\d8\dfdf73bec1d12026b30cb7ce8da650f3f0ea2cf155ea018ae
Successfully built jieba
Installing collected packages: jieba
Successfully installed jieba-0.42.1
Note: you may need to restart the kernel to use updated packages.
```

```
In [32]: import jieba
```

```
In [33]: seg_list=jieba.cut("似乎在述说着大自然的美",cut_all=True)
```

```
In [34]: print(",".join(seg_list))
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\Roshini\AppData\Local\Temp\jieba.cache
Loading model cost 1.384 seconds.
Prefix dict has been built successfully.
似乎,在,述说,着,大自然,自然,的,美
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

About 27,00,000 results (0.37 seconds)

orthopedics Overview Usage examples Pronunciation Similar and opposite words


Pronunciation

Indian English pronunciation

Sounds like

aw · tuh · pee · duhks

☐ Slow



Feedback