:[1] In

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

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

Out[1]:

```
True
```

:[2] In

```
text="Welcome readers. I hope you find it interesting. Please do reply."
from nltk.tokenize import sent_tokenize
print(sent_tokenize(text))
```

```
['.Welcome readers.', 'I hope you find it interesting.', 'Please do reply']
```

:[3] In

```
#How many sentence you had? 3
#How many sentence will we have if we replace full stop “.”With “,” in text? 1
```

:[4] In

```
import nltk
tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
text="Hello everyone. Hope all are fine and doing well. Hope you find the book interesting.
tokenizer.tokenize(text)
```

Out[4]:

```
,'.Hello everyone']
,'.Hope all are fine and doing well'
['.Hope you find the book interesting'
```

:[5] In

```
#Exercise 2: Try to tokenize this sentence:
text=".مرحبا بكم. نحن نتعلم اساسيات مبادئ استرجاع المعلومات"

tokenizer.tokenize(text)
```

Out[5]:

```
['مرحبا بكم.', 'نحن نتعلم اساسيات مبادئ استرجاع المعلومات']
```

:[6] In

```
text=nltk.word_tokenize("Welcome readers. I hope you find it interesting. Please do reply..
print(text)
```

```
 Welcome', 'readers', '.', 'I', 'hope', 'you', 'find', 'it', 'interesting',']
['«' ,'..'.', 'Please', 'do', 'reply
```

:[7] In

```
print(nltk.word_tokenize(input()))
```

```
shoaib rajeh alharbi
['shoaib', 'rajeh', 'alharbi']
```

:[8] In

```
import nltk
from nltk.tokenize import TreebankWordTokenizer
tokenizer = TreebankWordTokenizer()
tokenizer.tokenize("Have a nice day. You do great!")
```

Out[8]:

```
['!' ,'Have', 'a', 'nice', 'day.', 'You', 'do', 'great']
```

:[9] In

```
from nltk.tokenize import RegexpTokenizer
tokenizer=RegexpTokenizer("[\w]+")
tokenizer.tokenize('''Don't hesitate to ask
questions or send to me your question to
mohsarem@gmail.com''')
```

Out[9]:

```
,'Don']
,'t'
,'hesitate'
,'to'
,'ask'
,'questions'
,'or'
,'send'
,'to'
,'me'
,'your'
,'question'
,'to'
,'mohsarem'
,'gmail'
['com'
```

:[10] In

```
#Exercise 4: Modify the regular expression at step 3 above to find email address.
tokenizer=RegexpTokenizer("\S+@\S+")
tokenizer.tokenize('''Don't hesitate to ask
questions or send to me your question to
mohsarem@gmail.com''')
```

Out[10]:

```
['mohsarem@gmail.com']
```

:[12] In

```
text=[" It is a pleasant evening.","Guests, who came from US arrived at the venue","Food wa
from nltk.tokenize import word_tokenize
tokenized_docs=[word_tokenize(doc) for doc in text]
print(tokenized_docs)
```

```
It', 'is', 'a', 'pleasant', 'evening', '.'], ['Guests', ',', 'who', 'cam']]
 ,'e', 'from', 'US', 'arrived', 'at', 'the', 'venue'], ['Food', 'was', 'tasty
[['.'
```

:[13] In

```
import re
import string
x=re.compile('[%s]' % re.escape(string.punctuation))
tokenized_docs_no_punctuation = []
for review in tokenized_docs:
    new_review = []
    for token in review:
        new_token = x.sub(u'', token)
        if not new_token == u'':
            new_review.append(new_token)
    tokenized_docs_no_punctuation.append(new_review)
print(tokenized_docs_no_punctuation)
```

```
 It', 'is', 'a', 'pleasant', 'evening'], ['Guests', 'who', 'came', 'from',']]
[[''US', 'arrived', 'at', 'the', 'venue'], ['Food', 'was', 'tasty
```

:[14] In

```
#Exercise 6. Apply lower () function and upper() function on the sentence below: Text= ‘NLT

text= "NLTK allows you to convert Text into Lowercase and uppercase"
print(text.upper())
print(text.lower())
```

```
NLTK ALLOWS YOU TO CONVERT TEXT INTO LOWERCASE AND UPPERCASE
nltk allows you to convert text into lowercase and uppercase
```

:[15] In

```
nltk.download('stopwords')
```

```
nltk_data] Downloading package stopwords to]
...nltk_data]     C:\Users\USER\AppData\Roaming\nltk_data]
!nltk_data]   Package stopwords is already up-to-date]
```

Out[15]:

```
True
```

:[16] In

```
import nltk
from nltk.corpus import stopwords
stops=set(stopwords.words('english'))
words=["Don't",'hesitate','to','ask','questions']
[word for word in words if word not in stops]
```

Out[16]:

```
['Don't", 'hesitate', 'ask', 'questions"]
```

:[ ] In

:[ ] In