In [1]:
```
"Hi Siri Good Morning"
```

Out[1]:
```
'Hi Siri Good Morning'
```

```
Hi
Siri
Good
Morning
```

In [15]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /home/punit/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[15]:
```
True
```

Tokenization

In [12]:
```
text = "Mumbai or Bombay is the capital city of the Indian state of Maharashtra. According to the United
```

In [13]:
```
text
```

Out[13]:
```
'Mumbai or Bombay is the capital city of the Indian state of Maharashtra. According to the United Nations,
as of 2018, Mumbai was the second most populated city in India after Delhi. In the world with a population
of roughly 20 million'
```

using sent_tokenize()

In [16]:
```
sentence = nltk.sent_tokenize(text)
```

In [17]:
```
1 sentence
```

Out[17]:
```
['Mumbai or Bombay is the capital city of the Indian state of Maharashtra.',
 'According to the United Nations, as of 2018, Mumbai was the second most populated city in India after Del
hi.',
 'In the world with a population of roughly 20 million']
```

In [21]:
```
1 for items in sentence:
2     print(items)
```

```
Mumbai or Bombay is the capital city of the Indian state of Maharashtra.
According to the United Nations, as of 2018, Mumbai was the second most populated city in India after Delh
i.
In the world with a population of roughly 20 million
```

using split function

In [22]:
```
1 val = "This is a cat"
```

In [23]:
```
1 val.split(' ')
```

Out[23]:
```
['This', 'is', 'a', 'cat']
```

In [24]:
```
1 val = "This is.a ,cat"
```

In [25]:
```
1 val
```

Out[25]:
```
'This is.a ,cat'
```

In [26]:
```
1 val.split(' .,')
```

Out[26]:
```
['This is.a ,cat']
```

In [ ]:
```
1
```

using regex

In [32]:
```
text = """Founded in 1991 Python Programming was built to give users a easier way of writing code.
In the Intial day's Python was considered a slow programming language.
Later Many Improvements  were made by updating the language.
And today it has more than 3 lakh librarys available.
Currently used in Data Related Field Heavily."""
```

In [33]:
```
text
```

Out[33]:
```
"Founded in 1991 Python Programming was built to give users a easier way of writing code.\nIn the Intial da
y's Python was considered a slow programming language.\nLater Many Improvements  were made by updating the
language.\nAnd today it has more than 3 lakh librarys available.\nCurrently used in Data Related Field Heav
ily."
```

In [29]:
```
import re
```

In [30]:
```
res = re.findall("[\w']+",text)
```

In [31]:
```
1 res
```

Out[31]:
```
['Founded',
 'in',
 '1991',
 'Python',
 'Programming',
 'was',
 'built',
 'to',
 'give',
 'users',
 'a',
 'easier',
 'way',
 'of',
 'writing',
 'code',
 'In',
 'the',
 'Intial',
 "day's",
 'Python',
 'was',
 'considered',
 'a',
 'slow',
 'programming',
 'language',
 'Later',
 'Many',
 'Improvements',
 'were',
 'made',
 'by',
 'updating',
 'the',
 'language',
 'And',
 'today',
 'it',
```

```
 'has',
 'more',
 'than',
 '3',
 'lakh',
 'librarys',
 'available',
 'Currently',
 'used',
 'in',
 'Data',
 'Related',
 'Field',
 'Heavily']
```

In [39]:
```
text

```

Out[39]:
```
"Founded in 1991 Python Programming was built to give users a easier way of writing code.\nIn the Intial da
y's Python was considered a slow programming language.\nLater Many Improvements  were made by updating the
language.\nAnd today it has more than 3 lakh librarys available.\nCurrently used in Data Related Field Heav
ily."
```

In [37]:
```
res1 = re.compile('[.]').split(text)
```

In [38]:
```
res1
```

Out[38]:
```
['Founded in 1991 Python Programming was built to give users a easier way of writing code',
 "\nIn the Intial day's Python was considered a slow programming language",
 '\nLater Many Improvements  were made by updating the language',
 '\nAnd today it has more than 3 lakh librarys available',
 '\nCurrently used in Data Related Field Heavily',
 '']
```

using word_tokenize (nltk)

In [40]:
```
from nltk.tokenize import word_tokenize
```

In [41]:
```
1 word_tokenize(text)
```

Out[41]:
```
['Founded',
 'in',
 '1991',
 'Python',
 'Programming',
 'was',
 'built',
 'to',
 'give',
 'users',
 'a',
 'easier',
 'way',
 'of',
 'writing',
 'code',
 '.',
 'In',
 'the',
 'Intial',
 'day',
 "'s",
 'Python',
 'was',
 'considered',
 'a',
 'slow',
 'programming',
 'language',
 '.',
 'Later',
 'Many',
 'Improvements',
 'were',
 'made',
 'by',
 'updating',
 'the',
```

```
 'language',
 '.',
 'And',
 'today',
 'it',
 'has',
 'more',
 'than',
 '3',
 'lakh',
 'librarys',
 'available',
 '.',
 'Currently',
 'used',
 'in',
 'Data',
 'Related',
 'Field',
 'Heavily',
 '.']
```

using sent_tokenize (nltk)

In [42]:
```
1 from nltk.tokenize import sent_tokenize
```

In [43]:
```
1 sent_tokenize(text)
```

Out[43]:
```
['Founded in 1991 Python Programming was built to give users a easier way of writing code.',
 "In the Intial day's Python was considered a slow programming language.",
 'Later Many Improvements  were made by updating the language.',
 'And today it has more than 3 lakh librarys available.',
 'Currently used in Data Related Field Heavily.']
```

In [ ]:
```
1
```

punctuation removal

In [44]:
```
1 from nltk.tokenize import RegexpTokenizer
```

In [45]:
```
1 tokenizer = RegexpTokenizer(r'\w+')
```

In [48]:
```
1 result = tokenizer.tokenize("Wow! I am excited to learn NLP!!")
```

In [49]:
```
1 print(result)
```

```
['Wow', 'I', 'am', 'excited', 'to', 'learn', 'NLP']
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [50]:
```
1 txt = "Wow! I am excited to learn NLP!!"
```

In [51]:
```
1 import re
```

In [52]:
```
1 re.findall('\w+',txt)
```

Out[52]:
```
['Wow', 'I', 'am', 'excited', 'to', 'learn', 'NLP']
```

In [ ]:
```
1
```

white space tokenizer

White space tokenizer module of NLTK tokenizes a string on white space (space, tab, newline) It is an alternate to split()

In [53]:
```
1 from nltk.tokenize import WhitespaceTokenizer
```

In [54]:
```
1 txt = "Good Breads cost Rs 40\nIn Mumbai. Can you buy me\ntwo of them\n\nThanks."
```

In [55]:
```
1 txt
```

Out[55]:
```
'Good Breads cost Rs 40\nIn Mumbai. Can you buy me\ntwo of them\n\nThanks.'
```

In [58]:
```
1 tokenizer = WhitespaceTokenizer() #making an object
```

In [59]:
```
1 tokenizer.tokenize(txt)
```

Out[59]:
```
['Good',
 'Breads',
 'cost',
 'Rs',
 '40',
 'In',
 'Mumbai.',
 'Can',
 'you',
 'buy',
 'me',
 'two',
 'of',
 'them',
 'Thanks.']
```

In [ ]:
```
1
```

In [60]:
```
1 import pandas as pd
```

In [62]:
```
1 df = pd.DataFrame({'Phrases':['Stay Hungry Stay Foolish',
2                               'Faith Can Move Mountains',
3                                'The way to get started is to quit talking and begin doing',
4                                "If you set your goals ridiculously high and it's a failur, you will fail"
5                                "They say dreams do come true but nightmare are also dreams"
6                               ]})
```

In [63]:
```
1 df
```

Out[63]:

	Phrases
0	Stay Hungry Stay Foolish
1	Faith Can Move Mountains
2	The way to get started is to quit talking and ...
3	If you set your goals ridiculously high and it...
4	They say dreams do come true but nightmare are...

In [64]:
```
1 df['tokenize_col'] = df.apply(lambda row: nltk.word_tokenize(row['Phrases']),axis=1)
```

In [65]:
```
1 df
```

Out[65]:

	Phrases	tokenize_col
0	Stay Hungry Stay Foolish	[Stay, Hungry, Stay, Foolish]
1	Faith Can Move Mountains	[Faith, Can, Move, Mountains]
2	The way to get started is to quit talking and ...	[The, way, to, get, started, is, to, quit, tal...
3	If you set your goals ridiculously high and it...	[If, you, set, your, goals, ridiculously, high...
4	They say dreams do come true but nightmare are...	[They, say, dreams, do, come, true, but, night...

In [ ]:
```
1
```