In [2]:

```
# TextBlob is an object-oriented NLP text-processing library that is built on the NLTK a
# pattern NLP libraries and simplifies many of their capabilities.
# To install textblob, we will execute following command:

pip install textblob
```

```
Requirement already satisfied: textblob in c:\python\lib\site-packages (0.
17.1)
Requirement already satisfied: nltk>=3.1 in c:\python\lib\site-packages (f
rom textblob) (3.6.3)
Requirement already satisfied: tqdm in c:\python\lib\site-packages (from n
ltk>=3.1->textblob) (4.51.0)
Requirement already satisfied: click in c:\python\lib\site-packages (from
nltk>=3.1->textblob) (7.1.2)
Requirement already satisfied: regex in c:\python\lib\site-packages (from
nltk>=3.1->textblob) (2020.11.13)
Requirement already satisfied: joblib in c:\python\lib\site-packages (from
nltk>=3.1->textblob) (0.17.0)
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 21.2.4; however, version 22.0.3 is avai
lable.
You should consider upgrading via the 'c:\python\python.exe -m pip install
--upgrade pip' command.
```

In [1]:

```
# Once installation completes, execute the following command to download the NLTK
# corpora used by TextBlob:

import textblob
import subprocess
cmd = ['python','-m','textblob.download_corpora']
subprocess.run(cmd)
```

Out[1]:

```
CompletedProcess(args=['python', '-m', 'textblob.download_corpora'], retur
ncode=0)
```

In [3]:

```
# TextBlob is the fundamental class for NLP with the textblob module. Let's create a
# TextBlob containing two sentences:

from textblob import TextBlob
```

In [4]:

```
text = 'Today is a beautiful day. Tomorrow looks like bad weather.'
```

In [5]:

```
blob = TextBlob(text)
```

In [6]:

```
blob
```

Out[6]:

```
TextBlob("Today is a beautiful day. Tomorrow looks like bad weather.")
```

In [7]:

```
# Natural language processing often requires tokenizing text before performing other NLP
# tasks. TextBlob provides convenient properties for accessing the sentences and words f
# TextBlobs. Let's use the sentence property to get a list of Sentence objects:

blob.sentences
```

Out[7]:

```
[Sentence("Today is a beautiful day."),
 Sentence("Tomorrow looks like bad weather.")]
```

In [8]:

```
# The words property returns a WordList object containing a list of Word objects, repres
# each word in the TextBlob with the punctuation removed:

blob.words
```

Out[8]:

```
WordList(['Today', 'is', 'a', 'beautiful', 'day', 'Tomorrow', 'looks', 'li
ke', 'bad', 'weather'])
```

In [9]:

```
# Parts-of-speech (POS) tagging is the process of evaluating words based on their contex
# to determine each word's part of speech. There are eight primary English parts of spee
# nouns, pronouns, verbs, adjectives, adverbs, prepositions, conjunctions and interject
# (words that express emotion and that are typically followed by punctuation, like "Yes!
# "Ha!"). Within each category there are many subcategories.

# Some words have multiple meanings. For example, the words "set" and "run" have
# hundreds of meanings each! If you look at the dictionary.com definitions of the word
# "run," you'll see that it can be a verb, a noun, an adjective or a part of a verb phra
# important use of POS tagging is determining a word's meaning among its possibly many
# meanings. This is important for helping computers "understand" natural language.

# The tags property returns a list of tuples, each containing a word and a string repres
# its part-of-speech tag:

blob
```

Out[9]:

```
TextBlob("Today is a beautiful day. Tomorrow looks like bad weather.")
```

In [10]:

```
blob.tags
```

Out[10]:

```
[('Today', 'NN'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('beautiful', 'JJ'),
 ('day', 'NN'),
 ('Tomorrow', 'NNP'),
 ('looks', 'VBZ'),
 ('like', 'IN'),
 ('bad', 'JJ'),
 ('weather', 'NN')]
```

In [11]:

```
# A TextBlob's noun_phrases property returns a WordList object containing a list of
# Word objects—one for each noun phrase in the text:

blob.noun_phrases

# Note that a Word representing a noun phrase can contain multiple words. A WordList is
# an extension of Python's built-in list type. WordLists provide additional methods for
# stemming, lemmatizing, singularizing and pluralizing.
```

Out[11]:

```
WordList(['beautiful day', 'tomorrow', 'bad weather'])
```