In [1]:
```
#Converting words to their base forms using stemming
from nltk.stem.porter import PorterStemmer
from nltk.stem.lancaster import LancasterStemmer
from nltk.stem.snowball import SnowballStemmer
```

In [2]:
```
#Define some input words:
input_words = ['writing', 'calves', 'be', 'branded', 'horse', 'randomize',
        'possibly', 'provision', 'hospital', 'kept', 'scratchy', 'code']
```

In [3]:
```
# Create various stemmer objects
porter = PorterStemmer()
lancaster = LancasterStemmer()
snowball = SnowballStemmer('english')
```

In [4]:
```
# Create a list of stemmer names for display
stemmer_names = ['PORTER', 'LANCASTER', 'SNOWBALL']
formatted_text = '{:>16}' * (len(stemmer_names) + 1)
print('\n', formatted_text.format('INPUT WORD', *stemmer_names),
        '\n', '='*68)
```

```
      INPUT WORD          PORTER       LANCASTER        SNOWBALL
====================================================================
```

In [5]:
```
#Iterate through the words and stem them using the three stemmers:
# Stem each word and display the output
for word in input_words:
    output = [word, porter.stem(word),
            lancaster.stem(word), snowball.stem(word)]
    print(formatted_text.format(*output))
```

```
         writing           write            writ           write
          calves            calv            calv            calv
              be              be              be              be
         branded           brand           brand           brand
           horse            hors            hors            hors
       randomize          random          random          random
        possibly         possibl            poss         possibl
       provision          provis          provid          provis
        hospital          hospit          hospit          hospit
            kept            kept            kept            kept
        scratchy        scratchi        scratchy        scratchi
            code            code             cod            code
```

In [6]:
```
#Converting words to their base forms using lemmatization
#Create a new Python file and import the following packages:
from nltk.stem import WordNetLemmatizer
```

In [9]:
```
#Define some input words. We will be using the same set of words that we used in the previous section so that we can compare the outputs:
input_words = ['writing', 'calves', 'be', 'branded', 'horse', 'randomize',
        'possibly', 'provision', 'hospital', 'kept', 'scratchy', 'code']
```

In [10]:
```
# Create lemmatizer object
lemmatizer = WordNetLemmatizer()
```

In [11]:
```
#Create a list of lemmatizer names for the table display and format the text accordingly:

lemmatizer_names = ['NOUN LEMMATIZER', 'VERB LEMMATIZER']
formatted_text = '{:>24}' * (len(lemmatizer_names) + 1)
print('\n', formatted_text.format('INPUT WORD', *lemmatizer_names),
        '\n', '='*75)
```

```
                INPUT WORD         NOUN LEMMATIZER         VERB LEMMATIZER
===========================================================================
```

In [12]:
```
# Lemmatize each word and display the output
for word in input_words:
    output = [word, lemmatizer.lemmatize(word, pos='n'),
           lemmatizer.lemmatize(word, pos='v')]
    print(formatted_text.format(*output))
```

```
                 writing                 writing                   write
                  calves                    calf                   calve
                      be                      be                      be
                 branded                 branded                   brand
                   horse                   horse                   horse
               randomize               randomize               randomize
                possibly                possibly                possibly
               provision               provision               provision
                hospital                hospital                hospital
                    kept                    kept                    keep
                scratchy                scratchy                scratchy
                    code                    code                    code
```

In [ ]:
```
```