Task:

Write a program for finding all words in a text collection that meet the following criteria

1. ending in "ize"
2. containing letter "z"
3. containing the sequence of letter "pt"
4. all lowercases except for an initial capital
5. all words larger than 4 characters

In [1]:

```
import nltk
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/i
ndex.xml (https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/inde
x.xml)
```

Out[1]:

```
True
```

In [2]:

```
from nltk.book import *
```

```
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

In [3]:

```
text1
```

Out[3]:

```
<Text: Moby Dick by Herman Melville 1851>
```

In [4]:

```
from nltk.corpus import gutenberg
from nltk.text import Text
```

In [5]:

```
corpus = gutenberg.words('melville-moby_dick.txt')
```

In [6]:

```
text = Text(corpus)
```

TASK 1:

In [10]:

```
ize_words = [word for word in text if word.endswith('ize')]
```

In [17]:

```
print(ize_words)
```

```
['size', 'symbolize', 'size', 'tranquillize', 'seize', 'size', 'populari
ze', 'seize', 'symbolize', 'tranquillize', 'jeopardize', 'seize', 'pulve
rize', 'seize', 'pulverize', 'size', 'realize', 'seize', 'size', 'priz
e', 'parenthesize', 'capsize', 'Subtilize', 'size', 'size', 'hypothesiz
e', 'size', 'seize', 'size', 'seize', 'size', 'size', 'size', 'prize',
'seize', 'prize']
```

TASK 2:

In [18]:

```
z_words = [word for word in text if 'z' in word]
z_words
```

```
 'Lazarus',
 'Lazarus',
 'Czar',
 'frozen',
 'puzzled',
 'froze',
 'gazed',
 'frozen',
 'dazzling',
 'bamboozingly',
 'crazy',
 'doze',
 'blaze',
 'doze',
 'frozen',
 'puzzle',
 'civilized',
 'civilized',
 'civilized',
 'amazement',
```

TASK 3:

In [19]:

```
pt_words = [word for word in text if 'pt' in word]
```

In [20]:

```
pt_words
```

Out[20]:

```
['Consumptive',
 'empty',
 'Hampton',
 'contemptible',
 'capture',
 'captain',
 'Captain',
 'Egyptians',
 'captain',
 'promptly',
 'captains',
 'kept',
 'deceptive',
 'attempt',
 'eruption',
 'description',
 'kept',
 'slept',
```

TASK 4:

In [23]:

```
lc_words = [word for word in text if word.islower() == False and word[1:].islower()]
```

In [24]:

```
lc_words
```

Out[24]:

```
['Moby',
 'Dick',
 'Herman',
 'Melville',
 'Supplied',
 'Late',
 'Consumptive',
 'Usher',
 'Grammar',
 'School',
 'The',
 'Usher',
 'He',
 'He',
 'While',
 'Sw',
 'Dan',
 'This',
```

TASK 5:

In [26]:

```
long_words = [word for word in text if len(word) > 4]
```

In [27]:

```
long_words
```

```
 'roundness',
 'rolling',
 'HVALT',
 'arched',
 'vaulted',
 'WEBSTER',
 'DICTIONARY',
 'WHALE',
 'immediately',
 'WALLEN',
 'wallow',
 'RICHARDSON',
 'DICTIONARY',
 'KETOS',
 'GREEK',
 'CETUS',
 'LATIN',
 'WHOEL',
 'ANGLO',
 'SAXON',
```