Wordnet is a project started in mid 1980s at Princeton. It is a hierarchical organization of nouns, verbs, adjectives and adverbs.

```
import nltk
nltk.download('omw-1.4')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
from nltk.corpus import wordnet as wn
```

```
wn.synsets('cry') #Get sysnets of 'cry'
```

```
[Synset('cry.n.01'),
 Synset('cry.n.02'),
 Synset('war_cry.n.01'),
 Synset('cry.n.04'),
 Synset('cry.n.05'),
 Synset('shout.v.02'),
 Synset('cry.v.02'),
 Synset('exclaim.v.01'),
 Synset('cry.v.04'),
 Synset('cry.v.05'),
 Synset('cry.v.06'),
 Synset('cry.v.07')]
```

```
wn.synset('cry.n.02').definition() #get definition of the second noun synset
```

```
'a loud utterance of emotion (especially when inarticulate)'
```

```
wn.synset('cry.n.02').examples() #get its examples
```

```
['a cry of rage', 'a yell of pain']
```

```
wn.synset('cry.n.02').lemmas() #get its lemmas
```

```
[Lemma('cry.n.02.cry'), Lemma('cry.n.02.yell')]
```

```
cry = wn.synset('cry.n.02')
```

```
#Go up the hierarchy
hyp = cry.hypernyms()[0]
```

```
top = wn.synset('entity.n.01')
while hyp:
    print(hyp)
    if hyp == top:
        break
    if hyp.hypernyms():
        hyp = hyp.hypernyms()[0]
```

```
 Synset('utterance.n.01')
 Synset('auditory_communication.n.01')
 Synset('communication.n.02')
 Synset('abstraction.n.06')
 Synset('entity.n.01')
```

The way nouns get organized makes a tree where the higher the level the less specific the word. The words can then fit a lot more actual specific words.

```
hyper = lambda s: s.hypernyms()
list(cry.closure(hyper))
```

```
[Synset('utterance.n.01'),
 Synset('auditory_communication.n.01'),
 Synset('communication.n.02'),
 Synset('abstraction.n.06'),
 Synset('entity.n.01')]
```

```
hypo = lambda s: s.hyponyms()
list(cry.closure(hypo))
```

```
[Synset('complaint.n.02'), Synset('exclamation.n.02'), Synset('lament.n.01')]
```

```
mero = lambda s: s.part_meronyms()
list(cry.closure(mero))
```

```
[]
```

```
hyper = lambda s: s.part_holonyms()
list(cry.closure(hyper))
```

```
[]
```

```
anto = cry.lemmas()[0].antonyms()
anto
```

```
[]
```

5. Selecting verb "jump"

```
wn.synsets('jump')
```

```
[Synset('jump.n.01'),
 Synset('leap.n.02'),
 Synset('jump.n.03'),
 Synset('startle.n.01'),
 Synset('jump.n.05'),
 Synset('jump.n.06'),
 Synset('jump.v.01'),
 Synset('startle.v.02'),
 Synset('jump.v.03'),
 Synset('jump.v.04'),
 Synset('leap_out.v.01'),
 Synset('jump.v.06'),
 Synset('rise.v.11'),
 Synset('jump.v.08'),
 Synset('derail.v.02'),
 Synset('chute.v.01'),
 Synset('jump.v.11'),
 Synset('jumpstart.v.01'),
 Synset('jump.v.13'),
 Synset('leap.v.02'),
 Synset('alternate.v.01')]
```

```
wn.synset('jump.v.01').definition()  #get definition of the second noun synset
```

```
'move forward by leaps and bounds'
```

```
wn.synset('jump.v.01').examples() #get its examples
```

```
['The horse bounded across the meadow',
 'The child leapt across the puddle',
 'Can you jump over the fence?']
```

```
wn.synset('jump.v.01').lemmas() #get its lemmas
```

```
[Lemma('jump.v.01.jump'),
 Lemma('jump.v.01.leap'),
 Lemma('jump.v.01.bound'),
 Lemma('jump.v.01.spring')]
```

```
#Go up the hierarchy
jump = wn.synset('jump.v.01')
hyp = jump.hypernyms()[0]
top = wn.synset('entity.n.01')
while hyp:
    print(hyp)
    if hyp == top:
        break
```

```
        if hyp.hypernyms():
            hyp = hyp.hypernyms()[0]
```

```
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
Synset('move.v.03')
```

```
Synset('move.v.03')
```