

a. Study of wordnet dictionary with methods such as synsets, definitions, examples and lemmas.

Code and Output:

```
import nltk

from nltk.corpus import wordnet as wn

wn.synsets('motorcar')
| [Synset('car.n.01')]

wn.synset('car.n.01').lemma_names()
| ['car', 'auto', 'automobile', 'machine', 'motorcar']

wn.synset('car.n.01').examples()
| ['he needs a car to get to work']

wn.synset('car.n.01').definition()
| 'a motor vehicle with four wheels; usually propelled by an internal combustion engine'

wn.synset('fruit.n.01').definition()
| 'the ripened reproductive body of a seed plant'

wn.synset('cartoon.n.01').definition()
| 'a humorous or satirical drawing published in a newspaper or magazine'

wn.synset('bike.n.01').definition()
| 'a motor vehicle with two wheels and a strong frame'

wn.synset('animal.n.01').definition()
| 'a living organism characterized by voluntary movement'

wn.synset('fruit.n.01').lemmas
| <bound method Synset.lemmas of Synset('fruit.n.01')>

wn.lemma('car.n.01.automobile')
| Lemma('car.n.01.automobile')

wn.lemma('car.n.01.automobile').synset()
```

```
Synset('car.n.01')

wn.lemma('car.n.01.automobile').name()

'automobile'

wn.lemmas('car')

[Lemma('car.n.01.car'), Lemma('car.n.02.car'), Lemma('car.n.03.car'), Lemma('car.n.04.car'), Lemma('cable_car.n.01.car')]

for synset in wn.synsets('car'):

    print(synset.lemma_names())

['car', 'auto', 'automobile', 'machine', 'motorcar']
['car', 'railcar', 'railway_car', 'railroad_car']
['car', 'gondola']
['car', 'elevator_car']
['cable_car', 'car']
```

b. Study of hyponyms, hypernyms, meronyms and entailments.

Code and Output:

```
motorcar = wn.synset('car.n.01')
```

```
motorcar
```

```
Synset('car.n.01')
```

```
types_of_motorcar = motorcar.hyponyms()
```

```
types_of_motorcar
```

```
[Synset('sport_utility.n.01'), Synset('horseless_carriage.n.01'), Synset('ambulance.n.01'), Synset('roadster.n.01'), Synset('convertible.n.01'), Synset('gas_guzzler.n.01'), Synset('subcompact.n.01'), Synset('touring_car.n.01'), Synset('beach_wagon.n.01'), Synset('coupe.n.01'), Synset('pace_car.n.01'), Synset('stanley_steamer.n.01'), Synset('electric.n.01'), Synset('jeep.n.01'), Synset('loaner.n.02'), Synset('minicar.n.01'), Synset('compact.n.03'), Synset('hot_rod.n.01'), Synset('cruiser.n.01'), Synset('hatchback.n.01'), Synset('sedan.n.01'), Synset('stock_car.n.01'), Synset('sports_car.n.01'), Synset('cab.n.03'), Synset('racer.n.02'), Synset('hardtop.n.01'), Synset('model_t.n.01'), Synset('minivan.n.01'), Synset('limousine.n.01'), Synset('used-car.n.01'), Synset('bus.n.04')]
```

```
types_of_motorcar[8]
```

```
Synset('beach_wagon.n.01')
```

```
motorcar.hypernyms()
```

```
[Synset('motor_vehicle.n.01')]
```

```
paths = motorcar.hypernym_paths()
```

```
len(paths)
```

```
2
```

```
[synset.name for synset in paths[0]]
```

```
[<bound method Synset.name of Synset('entity.n.01')>, <bound method Synset.name of Synset('physical_entity.n.01')>, <bound method Synset.name of Synset('object.n.01')>, <bound method Synset.name of Synset('whole.n.02')>, <bound method Synset.name of Synset('artifact.n.01')>, <bound method Synset.name of Synset('instrumentality.n.03')>, <bound method Synset.name of Synset('container.n.01')>, <bound method Synset.name of Synset('wheeled_vehicle.n.01')>, <bound method Synset.name of Synset('self-propelled_vehicle.n.01')>, <bound method Synset.name of Synset('motor_vehicle.n.01')>, <bound method Synset.name of Synset('car.n.01')>]
```

```
[synset.name for synset in paths[1]]
```

```
[<bound method Synset.name of Synset('entity.n.01')>, <bound method Synset.name of Synset('physical_entity.n.01')>, <bound method Synset.name of Synset('object.n.01')>, <bound method Synset.name of Synset('whole.n.02')>, <bound method Synset.name of Synset('artifact.n.01')>, <bound method Synset.name of Synset('instrumentality.n.03')>, <bound method Synset.name of Synset('conveyance.n.03')>, <bound method Synset.name of Synset('vehicle.n.01')>, <bound method Synset.name of Synset('wheeled_vehicle.n.01')>, <bound method Synset.name of Synset('self-propelled_vehicle.n.01')>, <bound method Synset.name of Synset('motor_vehicle.n.01')>, <bound method Synset.name of Synset('car.n.01')>]
```

```
motorcar.root_hyponyms()
```

```
[Synset('entity.n.01')]
```

```
wn.synset('tree.n.01').part_meronyms()
```

```
[Synset('trunk.n.01'), Synset('limb.n.02'), Synset('stump.n.01'), Synset('crown.n.07'), Synset('burl.n.02')]
```

```
wn.synset('tree.n.01').substance_meronyms()
```

```
[Synset('sapwood.n.01'), Synset('heartwood.n.01')]
```

```
wn.synset('walk.v.01').entailments()
```

```
[Synset('step.v.01')]
```

```
wn.synset('eat.v.01').entailments()
```

```
[Synset('chew.v.01'), Synset('swallow.v.01')]
```

```
wn.lemma('supply.n.02.supply').antonyms()
```

```
[Lemma('demand.n.02.demand')]
```

```
wn.lemma('rush.v.01.rush').antonyms()
```

```
[Lemma('linger.v.04.linger')]
```

```
wn.lemma('horizontal.a.01.horizontal').antonyms()
```

```
[Lemma('vertical.a.01.vertical'), Lemma('inclined.a.02.inclined')]
```

c. Write a program using python to find synonyms and antonyms.

Code and Output:

```
from nltk.corpus import wordnet
```

```
print(wordnet.synsets("active"))
```

```
[Synset('active_agent.n.01'), Synset('active_voice.n.01'), Synset('active.n.03'), Synset('active.a.01'), Synset('active.s.02'), Synset('active.a.03'), Synset('active.s.04'), Synset('active.a.05'), Synset('active.a.06'), Synset('active.a.07'), Synset('active.s.08'), Synset('active.a.09'), Synset('active.a.10'), Synset('active.a.11'), Synset('active.a.12'), Synset('active.a.13'), Synset('active.a.14')]
```

```
print(wn.lemma('active.a.01.active').antonyms())
```

```
[Lemma('inactive.a.02.inactive')]
```