[WordNet](https://wordnet.princeton.edu/" \t "blank) is a lexical(relating to the words or vocabulary of a language) database for the English language, which was created by Princeton, and is part of the NLTK corpus.

You can use WordNet alongside the NLTK module to find the meanings of words, synonyms, antonyms, and more. Let's cover some examples.

First, you're going to need to import wordnet:

from nltk.corpus import wordnet

Then, we're going to use the term "program" to find synsets like so:

syns = wordnet.synsets("program")

An example of a synset:

print(syns[0].name())

plan.n.01

Just the word:

print(syns[0].lemmas()[0].name())

plan

Definition of that first synset:

print(syns[0].definition())

a series of steps to be carried out or goals to be accomplished

Examples of the word in use:

print(syns[0].examples())

['they drew up a six-step plan', 'they discussed plans for a new bond issue']

Next, how might we discern synonyms and antonyms to a word? The lemmas will be synonyms, and then you can use .antonyms to find the antonyms to the lemmas. As such, we can populate some lists like:

synonyms = []

antonyms = []

for syn in wordnet.synsets("good"):

for l in syn.lemmas():

synonyms.append(l.name())

if l.antonyms():

antonyms.append(l.antonyms()[0].name())

print(set(synonyms))

print(set(antonyms))

{'beneficial', 'just', 'upright', 'thoroughly', 'in\_force', 'well', 'skilful', 'skillful', 'sound', 'unspoiled', 'expert', 'proficient', 'in\_effect', 'honorable', 'adept', 'secure', 'commodity', 'estimable', 'soundly', 'right', 'respectable', 'good', 'serious', 'ripe', 'salutary', 'dear', 'practiced', 'goodness', 'safe', 'effective', 'unspoilt', 'dependable', 'undecomposed', 'honest', 'full', 'near', 'trade\_good'} {'evil', 'evilness', 'bad', 'badness', 'ill'}

As you can see, we got many more synonyms than antonyms, since we just looked up the antonym for the first lemma, but you could easily balance this buy also doing the exact same process for the term "bad."

***similarity***

Next, we can also easily use WordNet to compare the similarity of two words and their tenses, by incorporating the [Wu and Palmer method](http://search.cpan.org/~tpederse/WordNet-Similarity-1.03/lib/WordNet/Similarity/wup.pm) for semantic related-ness.

Let's compare the noun of "ship" and "boat:"

w1 = wordnet.synset('ship.n.01')

w2 = wordnet.synset('boat.n.01')

print(w1.wup\_similarity(w2))

0.9090909090909091

w1 = wordnet.synset('ship.n.01')

w2 = wordnet.synset('car.n.01')

print(w1.wup\_similarity(w2))

0.6956521739130435

w1 = wordnet.synset('ship.n.01')

w2 = wordnet.synset('cat.n.01')

print(w1.wup\_similarity(w2))

0.38095238095238093