Synonyms.py  
- Adds all hypernyms, based on the previous project.  
- For each file and for each object:  
 > Generates a list of synonyms ( ss.lema\_names() ) in lowercase   
 > Checks if the synonym consists of multiple words (splitted by underscore)   
 > Generates 2 synonym lists: depending if the first letter is a vowel(to distinct ‘a’ and ‘an’)  
 > Generates strings of nouns and adjectives to put in the lexicon   
 > Put all object strings in a dictionary   
 > Value can consist of 1 or 2 lists (‘a’ lists and ‘an’ lists)

> Pluralizes all noun synonyms (if multiple word 🡪 pluralizes only the last word)  
 >> Based on a rule-based script found on the internet (rules and exceptions)   
 >> Puts in the noun dict

> Writes all prolog strings to a lexicon file

grimSearch.py  
- Generates a list of all nouns in the lexicon (nouns.pl)  
 > line.partition 🡺 gets part of string between to characters  
 > First creates a dictionary with sinsets as key and synonym list as value  
 > Makes grouped list (deletes doubles making a set)

- Takes a query phrase as input:  
 > makes unigram, bigrams and trigrams of the query  
 > Generates list of ngrams that are in the lexicon   
 > POS tags the query phrase  
 > Makes ngrams of (consecutive) nouns

> Checks if the ngram lists are consistent  
 > If the query ngrams have a nn that is not found in the POS ngrams 🡪 disambiguate  
 >> Asks user what he ment:  
 ex. a big cat drinks  
 - query bigram (big, cat) because big\_cat\_n1 is in lexicon   
 - not in POS ngrams because tagged as: big(‘JJ’) cat(‘NN’) drinks(‘NN’)  
 System asks: Did you mean  
 1) big cat = NN  
 2) big = JJ cat = NN

> Automatically discards other suggestions resulting from the noun choice. For example the POS bigram: ‘cat drinks’ is discarded when the ‘big cat’ is chosen chosen

> Checks if resulting nouns are in the lexicon (list)  
 > If not: it searches for a hypernym, until it finds one that is in the lexicon  
 > Replaces old noun by the hypernym in the query phrase

> Parses the query with the nouns that are in the lexicon