## Install packages

- python 3.6+
    - Recommend using python3.6+
    - If you've never installed Python before, I would recommend installing "Anaconda". It is a toolkit that equips you with thousands of open-source packages and libraries.
- NLTK 3.2+

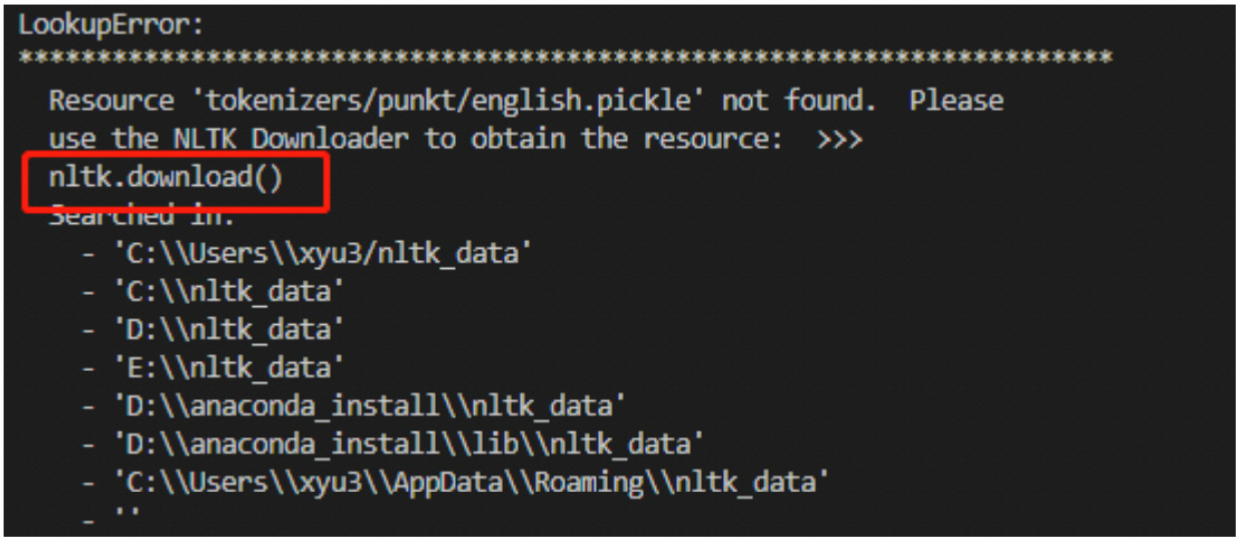
If you install Anaconda, you've already installed the NLTK package. If not , run the command below.

```
pip install nltk
```

Check if you've installed NLTK library.

First, type `python` to go into the py env. Then type the command below, and see if there is any error message.

```
>>> from nltk.corpus import stopwords
```




When you see error messages like above, just follow the instructions, because you need to install some other libraries to make it work.

## NLTK exercise

NLTK is a leading platform for building Python programs to work with human language data. It provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. You will need this library in your next coding homtwork.

- NLTK tokenization
  - link1

```
from nltk.tokenize import word_tokenize
text = "God is Great! I won a lottery."
print(word_tokenize(text))

Output: ['God', 'is', 'Great', '!', 'I', 'won', 'a', 'lottery', '.']
```

- NLTK stop words removal
  - link1

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

example_sent = "This is a sample sentence, showing off the stop words
filtration."

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)

# outputs
['This', 'is', 'a', 'sample', 'sentence', ',', 'showing',
'off', 'the', 'stop', 'words', 'filtration', '.']
['This', 'sample', 'sentence', ',', 'showing', 'stop',
'words', 'filtration', '.']
```

- NLTK stemming
  - link1

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

sentence = "Programers program with programing languages"
```

```
words = word_tokenize(sentence)

for w in words:
    print(w, " : ", ps.stem(w))

# outputs
Programers  :  program
program  :  program
with  :  with
programing  :  program
languages  :  language
```

## Python learning

- English version
  - an interactive learning website Check this
- Chinese version
  - Check this

When you are learning python, pay more attention to the python type "dict" , "list". Cause these datastructs will be used frequently in your next coding homework.

## Some other materials might be helpful

There are some other materials I think you might need after you grasp basic usage of python.

1. python official doc website

   When you want to learn some usage of python built-in function, you can searn it in the official doc website.

2. effective python

   If you want to write some clean python code, or want to learn some tricks about python usage. Try this book

3. design pattern

   If you want to write some industrial level python code, design pattern is one of the most important features you need to learn.