**Code Documentation**

**Packages/Libraries Imported**

**os:** The OS module in python provides functions for interacting with the operating system

**csv :** To read and write csv files (data sets)

**math :** To incorporate math modules and functions

**nltk.stem :** For stemming (producing morphological variants of a root/base word) to shorten the lookup, and normalize sentences

**nltk.tokenize :** Incorporating libraries for tokenization and removal of stopwords

**Functions Used**

**Sort\_Tuple(tp):**

We define this function to sort tuples (key-value pair) based on their key values

**Sort\_Tuple2(tp):**

We define this function to sort tuples (key-value pair) based on their values

**Convert(tp, di):**

We define this function to merge term incidences documents to create the posting list for each term

**removeDuplicates(lst):**

We define this function to remove duplicates of the term

## **CORPUS USED:**

<https://www.kaggle.com/shankarpandala/mann-ki-baat-speech-corpus>

Corpus is in a collection of text files of the Prime Minister’s speeches.

## **DATA STRUCTURES USED:**

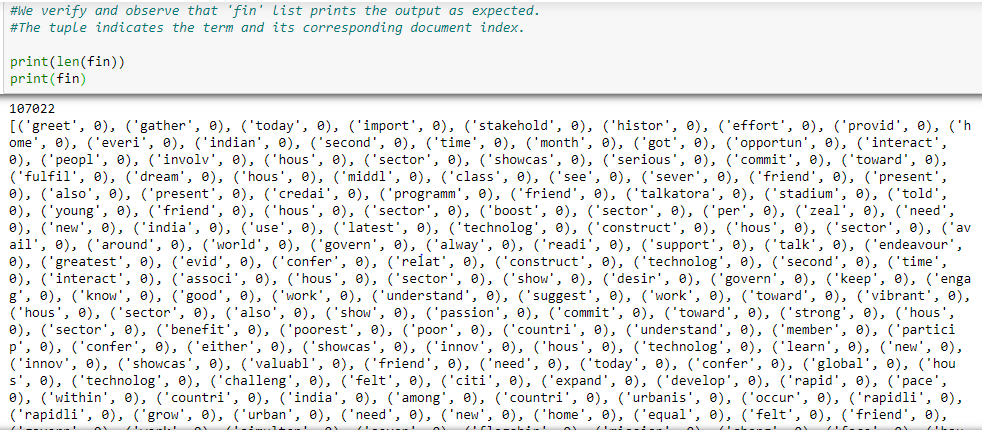
### **Tuple**

**Tuple** is a collection of Python objects much like a list. The sequence of values stored in a tuple can be of any type, and they are indexed by integers. The important difference between a list and a tuple is that tuples are immutable.

Example: ('aadhar', 64)

### **List (list)**

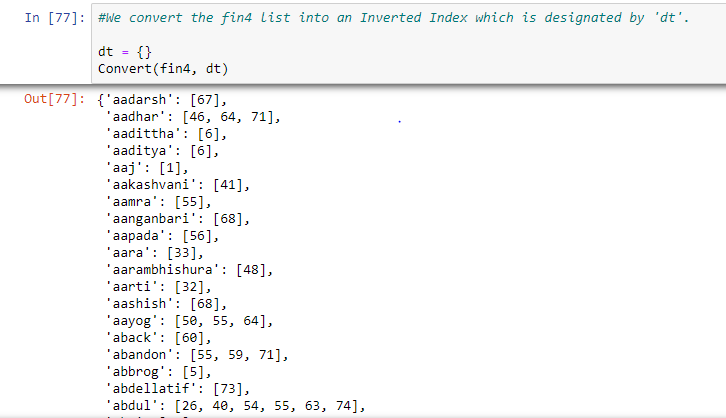
Contains lists enclosed within a list It will contain the stemmed tokens from each file in the corpus as individual lists. All are appended to make a list. Example:



*Snapshot of a list of tuples used in the project.*

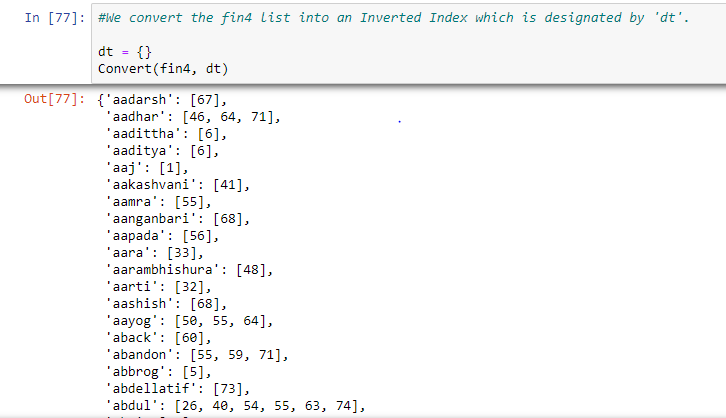
### **Dictionary (dict)**

Contains key-value pair. Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.   
Example:[1: 'I', 2: 'am', 3: 'fine']  
  
In our project, Inverted Index has been stored using the dictionary data structure where  
< key: distinct-normalized-terms-in-corpus, value: posting-list >



### **Vocabulary**

Will contain a dictionary of all the unique words in the corpus. Example:



*Snapshot of the Inverted Index*

A nested dictionary containing the following structure explained through the following example:(Numbers are just representational )

## **Preprocessed Data:**

The data has been stored in the .csv files:

* dict.csv - contains the Inverted Index along with the document frequency and the IDF scores
* vct.csv - contains the TF scores in various documents