Map Reduce steps  
===============  
To check in windows machine:  
#run command :   type .\animal.txt| python map.py | sort | python reduce.py

Now let’s check our reducer code reducer.py with mapper.py is it working properly or not with the help of the below command.

cat word\_count\_data.txt | python mapper.py | sort -k1,1 | python reducer.py

Now make a directory word\_count\_in\_python in our HDFS in the root directory that will store our word\_count\_data.txt file with the below command.

hdfs dfs -mkdir /word\_count\_in\_python

Copy word\_count\_data.txt to this folder in our HDFS with help of copyFromLocal command.

Syntax to copy a file from your local file system to the HDFS is given below:

hdfs dfs -copyFromLocal /path 1 /path 2 .... /path n /destination  
Actual command(in my case)

hdfs dfs -copyFromLocal /home/prashant/Documents/word\_count\_data.txt    /word\_count\_in\_python

===check the file with folder  
hdfs dfs -ls /       # list down content of the root directory  
hdfs dfs -ls /word\_count\_in\_python    # list down content of /word\_count\_in\_python directory

==change mode of python programs for execution in linux/unix env  
chmod 777 mapper.py reducer.py     # changing the permission to read, write, execute for user, group and others

Now download the latest hadoop-streaming jar file from this Link.   
<https://jar-download.com/artifacts/org.apache.hadoop/hadoop-streaming/2.7.3/source-code>

Then place, this Hadoop,-streaming jar file to a place from you can easily access it.   
here placing it to /Documents folder where mapper.py and reducer.py file is present.

Now let’s run our python files with the help of the Hadoop streaming utility as shown below.

hadoop jar /home/prashant/Documents/hadoop-streaming-2.7.3.jar \  
> -input /word\_count\_in\_python/word\_count\_data.txt \  
> -output /word\_count\_in\_python/output \  
> -mapper /home/prashant/Documents/mapper.py \  
> -reducer /home/prashant/Documents/reducer.py

In the above command in -output, specify the location in HDFS where we want our output to be stored.   
So let’s check our output in output file at location /word\_count\_in\_python/output/part-00000 in this case.   
Check results by manually visiting the location in HDFS or with the help of cat command as shown below.

hdfs dfs -cat /word\_count\_in\_python/output/part-00000

 mapper code

import sys

#Word Count Example

# input comes from standard input STDIN

for line in sys.stdin:

    line = line.strip() #remove leading and trailing whitespaces

    words = line.split() #split the line into words and returns as a list

for word in words:

#write the results to standard output STDOUT

    print('%s    %s' % (word,1)) #Emit the word

reducer code

#animal.txt :  lion cat dog lion tiger lion cat dog

#run command :   type .\animal.txt| python map.py | sort | python reduce.py

import sys

from operator import itemgetter

# using a dictionary to map words to their counts

current\_word = None

current\_count = 0

word = None

# input comes from STDIN

for line in sys.stdin:

    line = line.strip()

    word,count = line.split('   ',1)

    try:

        count = int(count)

    except ValueError:

        continue

    if current\_word == word:

        current\_count += count

    else:

        if current\_word:

            print ('%s   %s' % (current\_word, current\_count))

        current\_count = count

        current\_word = word

if current\_word == word:

    print ('%s   %s' % (current\_word,current\_count))