# Creating file mapper.py file in gedit and writing the mapper code for it

1. gedit mapper.py

2. Copy the mapper code below:

#!/usr/bin/python

"""mapper.py"""

import sys

for line in sys.stdin:

line = line.strip()

words = line.split()

for word in words:

print '%s\t%s' % (word, 1)

3. # Create a word.txt file and write names of students with different frequencies.

# word.txt should be in same working directory

4. gedit word.txt

ait

dighi

Pune

ait

dighi

Pune

ait

dighi

Pune

5. Execute mapper.py

6. cat word.txt | python mapper.py

prints (word,1) format if works fine

# If mapper works perfectly fine, we move to reducer.py

7. gedit reducer.py

#!/usr/bin/env python

"""reducer.py"""

import sys

current\_word = None

current\_count = 0

word = None

# input comes from STDIN

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# parse the input we got from mapper.py

word, count = line.split('\t', 1)

# convert count (currently a string) to int

try:

count = int(count)

except ValueError:

# count was not a number, so silently

# ignore/discard this line

continue

# this IF-switch only works because Hadoop sorts map output

# by key (here: word) before it is passed to the reducer

if current\_word == word:

current\_count += count

else:

if current\_word:

# write result to STDOUT

print '%s\t%s' % (current\_word, current\_count)

current\_count = count

current\_word = word

# do not forget to output the last word if needed!

if current\_word == word:

print '%s\t%s' % (current\_word, current\_count)

8. # Run reducer with mapper to generate results for the word.txt file

9. cat word.txt | python mapper.py | sort | python reducer.py

It will generate final ouptput