

# Kick Start Hadoop

This Blog is intended to give budding MapReduce developers a start off in developing hadoop based applications. It involves some development tips and tricks on hadoop MapReduce programming, tools that use map reduce under the hood and some practical applications of hadoop using these tools. Most of the code samples provided here is tested on hadoop environment but still do post me if you find any not working.

Friday, April 29, 2011

## Word Count - Hadoop Map Reduce Example

Word count is a typical example where Hadoop map reduce developers start their hands on with. This sample map reduce is intended to count the no of occurrences of each word in the provided input files.

### What are the minimum requirements?

1. Input text files – any text file
2. Cloudera test VM
3. The mapper, reducer and driver classes to process the input files

### How it works

The word count operation takes place in two stages a mapper phase and a reducer phase. In mapper phase first the text is tokenized into words then we form a key value pair with these words where the key being the word itself and value '1'. For example consider the sentence

"tring tring the phone rings"

In map phase the sentence would be split as words and form the initial key value pair as

```
<tring,1>
<tring,1>
<the,1>
<phone,1>
<rings,1>
```

In the reduce phase the keys are grouped together and the values for similar keys are added. So here there are only one pair of similar keys 'tring' the values for these keys would be added so the out put key value pairs would be

```
<tring,2>
<the,1>
<phone,1>
<rings,1>
```

This would give the number of occurrence of each word in the input. Thus reduce forms an aggregation phase for keys.

The point to be noted here is that first the mapper class executes completely on the entire data set splitting the words and forming the initial key value pairs. Only after this entire process is completed the reducer starts. Say if we have a total of 10 lines in our input files combined together, first the 10 lines are tokenized and key value pairs are formed in parallel, only after this the aggregation/ reducer would start its operation.

The figure below would throw more light to your understanding

Search This Blog

Search

### Blog Archive

- 2015 (1)
- 2012 (9)
- ▼ 2011 (23)
  - October (3)
  - September (2)
  - June (5)
  - May (11)
  - ▼ April (2)

Word Count - Hadoop Map Reduce Example

[How to run hadoop - map reduce jobs without a clu...](#)

### Something About Me



**Bejoy KS**

India

Hadoop enthusiast and an active

consumer of Hadoop technology stack like HDFS, MapReduce, Mahout, Hive, Sqoop, Pig, HBase, Oozie etc. More details <http://in.linkedin.com/pub/bejoy-ks/42/a52/aa0>

[View my complete profile](#)

### Total Pageviews



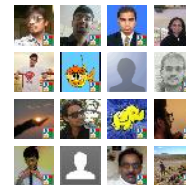
**285,226**

### Followers

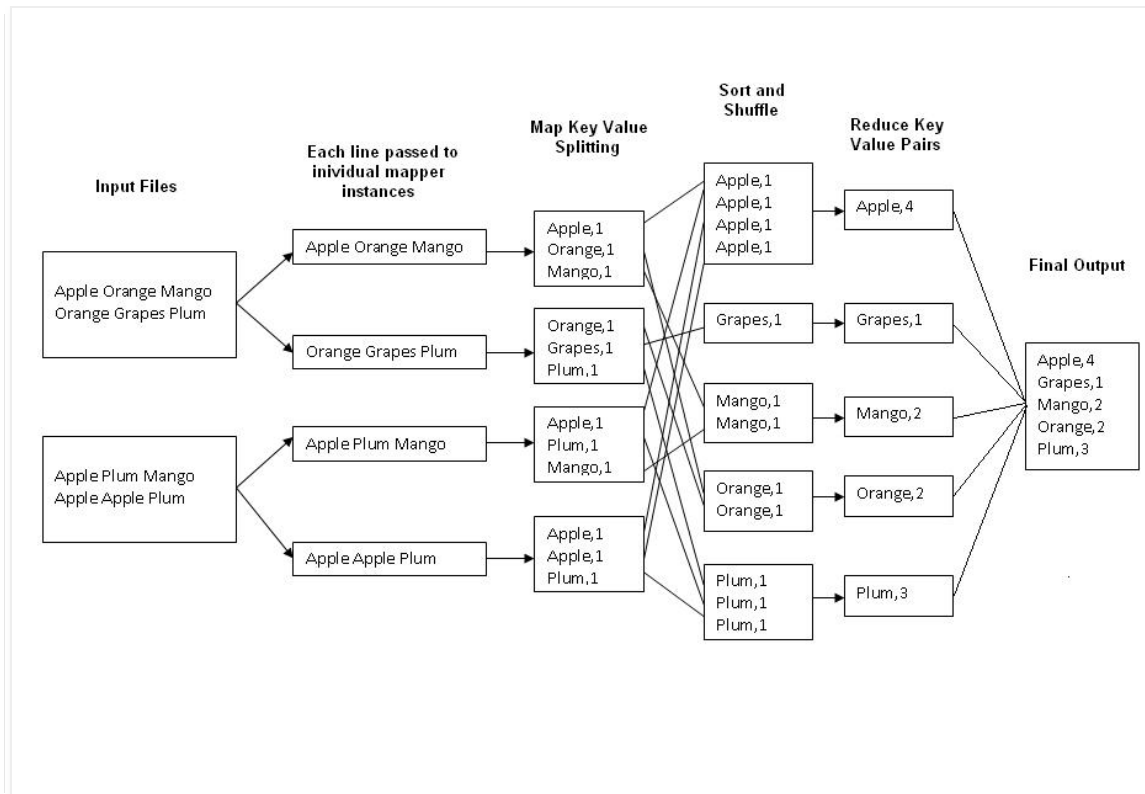
#### Join this site

with Google Friend Connect

**Members (80)** [More »](#)



Already a member? [Sign in](#)



Now coming to the practical side of implementation we need our input file and map reduce program jar to do the process job. In a common map reduce process two methods do the key job namely the map and reduce, the main method would trigger the map and reduce methods. For convenience and readability it is better to include the map, reduce and main methods in 3 different class files. We'd look at the 3 files we require to perform the word count job

#### Word Count Mapper

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class WordCountMapper extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable>
{
    //hadoop supported data types
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    //map method that performs the tokenizer job and framing
    the initial key value pairs
    public void map(LongWritable key, Text value,
    OutputCollector<Text, IntWritable> output, Reporter reporter)
    throws IOException
    {
        //taking one line at a time and tokenizing the same
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);

        //iterating through all the words available in that
        line and forming the key value pair
        while (tokenizer.hasMoreTokens())
        {
            word.set(tokenizer.nextToken());
            //sending to output collector which return passes
            the same to reducer
            output.collect(word, one);
        }
    }
}
```

Let us dive in details of this source code we can see the usage of a few deprecated classes and interfaces; this is because the code has been written to be compliant with Hadoop versions 0.18 and later. From Hadoop version 0.20 some of the methods are deprecated by still supported.

Lets now focus on the class definition part

```
implements Mapper<LongWritable, Text, Text, IntWritable>
```

What does this ~~Mapper~~<LongWritable, Text, Text, IntWritable> stand for?

The data types provided here are Hadoop specific data types designed for operational efficiency suited for massive parallel and lightning fast read write operations. All these data types are based out of java data types itself, for example LongWritable is the equivalent for long in java, IntWritable for int and Text for String.

When we use it as ~~Mapper~~<LongWritable, Text, Text, IntWritable>, it refers to the data type of input and output key value pairs specific to the mapper or rather the map method, ie Mapper<Input Key Type, Input Value Type, Output Key Type, Output Value Type>. In our example the input to a mapper is a single line, so this Text (one input line) forms the input value. The input key would a long value assigned in default based on the position of Text in input file. Our output from the mapper is of the format "Word, 1" hence the data type of our output key value pair is <Text(String), IntWritable(int)>

The next key component out here is the **map method**

```
map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter reporter)
```

We'd now look into each of the input parameters in detail. The first and second parameter refers to the Data type of the input Key and Value to the mapper. The third parameter is the output collector which does the job of taking the output data either from the mapper or reducer, with the output collector we need to specify the Data Types of the output Key and Value from the mapper. The fourth parameter, the reporter is used to report the task status internally in Hadoop environment to avoid time outs.

The functionality of the map method is as follows

1. Create a IntWritable variable 'one' with value as 1
2. Convert the input line in Text type to a String
3. Use a tokenizer to split the line into words
4. Iterate through each word and a form key value pairs as
  - a. Assign each work from the tokenizer(of String type) to a Text 'word'
  - b. Form key value pairs for each word as <word,one> and push it to the output collector

#### Word Count Reducer

```
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class WordCountReducer extends MapReduceBase implements
Reducer<Text, IntWritable, Text, IntWritable>
{
    //reduce method accepts the Key Value pairs from mappers,
    do the aggregation based on keys and produce the final out put
    public void reduce(Text key, Iterator<IntWritable> values,
    OutputCollector<Text, IntWritable> output, Reporter reporter)
    throws IOException
    {
        int sum = 0;
        /*iterates through all the values available with a
        key and add them together and give the
        final result as the key and sum of its values*/
        while (values.hasNext())
        {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

Here like for the mapper the reducer implements

```
Reducer<Text, IntWritable, Text, IntWritable>
```

The first two refers to data type of Input Key and Value to the reducer and the last two refers to data type of output key and value. Our mapper emits output as <apple,1>, <grapes,1>, <apple,1> etc. This is the input for reducer so here the data types of key and value in java would be String and int, the equivalent in Hadoop would be Text and IntWritable. Also we get the output as <word, no of occurrences> so the data type of output Key Value would be <Text, IntWritable>

Now the key component here, the **reduce method**.

The input to reduce method from the mapper after the sort and shuffle phase

would be the key with the list of associated values with it. For example here we have multiple values for a single key from our mapper like <apple,1> , <apple,1> , <apple,1> , <apple,1> . This key values would be fed into the reducer as < apple, {1,1,1,1} > .

Now let us evaluate our reduce method  
`reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter)`  
 Here all the input parameters are hold the same functionality as that of a mapper, the only difference is with the input Key Value. As mentioned earlier the input to a reducer instance is a key and list of values hence 'Text key, Iterator<IntWritable> values' . The next parameter denotes the output collector of the reducer with the data type of output Key and Value.

The functionality of the reduce method is as follows

1. Initiaze a variable 'sum' as 0
2. Iterate through all the values with respect to a key and sum up all of them
3. Push to the output collector the Key and the obtained sum as value

#### Driver Class

The last class file is the driver class. This driver class is responsible for triggering the map reduce job in Hadoop, it is in this driver class we provide the name of our job, output key value data types and the mapper and reducer classes. The source code for the same is as follows

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WordCount extends Configured implements Tool{
    public int run(String[] args) throws Exception
    {
        //creating a JobConf object and assigning a job name
        //for identification purposes
        JobConf conf = new JobConf(getConf(),
        WordCount.class);
        conf.setJobName("WordCount");

        //Setting configuration object with the Data Type of
        //output Key and Value
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        //Providing the mapper and reducer class names
        conf.setMapperClass(WordCountMapper.class);
        conf.setReducerClass(WordCountReducer.class);

        //the hdfs input and output directory to be fetched
        //from the command line
        FileInputFormat.addInputPath(conf, new
        Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new
        Path(args[1]));

        JobClient.runJob(conf);
        return 0;
    }

    public static void main(String[] args) throws Exception
    {
        int res = ToolRunner.run(new Configuration(), new
        WordCount(),args);
        System.exit(res);
    }
}
```

Create all the three java files in your project. Now you'd be having compilation errors just get the latest release of Hadoop and add the jars on to your class path. Once free from compilation errors we have to package them to a jar. If you are using eclipse then right click on the project and use the export utility. While packing the jar it is better not to give the main class, because in future when you have multiple map reduce and multiple drivers for the same project we should leave an option to choose the main class file during run time through the command line.

Follow the steps to execute the job

1. Copy the jar to a location in LFS (/home/training/usecase/wordcount/wordcount.jar)
2. Copy the input files from windows to LFS(/home/training/usecase/wordcount/input/)
3. Create an input directory in HDFS  
`hadoop fs -mkdir /projects/wordcount/input/`

4. Copy the input files from LFS to HDFS  
Hadoop fs -copyFromLocal /home/training/usecase/wordcount/input/\* /projects/wordcount/input/
5. Execute the jar  
hadoop jar /home/training/usecase/wordcount/wordcount.jar  
com.bejoy.samples.wordcount.WordCount /projects/wordcount/input/ /projects/wordcount/output/

We'd just look at the command in detail with each parameter  
/home/training/usecase/wordcount/wordcount.jar -> full path of the jar file in LFS  
com.bejoy.samples.wordcount.WordCount -> full package name of the Driver Class  
/projects/wordcount/input/ -> input files location in HDFS  
/projects/wordcount/output/ -> a directory in HDFS where we need the output files

**NOTE:** In Hadoop the map reduce process creates the output directory in hdfs and store the output files on to the same. If the output directory already exists in Hadoop then the m/r job wont execute, in that case either you need to change the output directory or delete the provided output directory in HDFS before running the jar again

6. Once the job shows a success status we can see the output file in the output directory(part-00000)  
Hadoop fs -ls /projects/wordcount/output/
7. For any further investigation of output file we can retrieve the data from hdfs to LFS and from there to the desired location  
hadoop fs -copyToLocal /projects/wordcount/output/ /home/training/usecase/wordcount/output/

#### Some better practices

In our current example with the configuration parameters or during runtime we are not specifying the number of reducers. In default Hadoop map reduce jobs have the default no of reducers as one, hence one only one reducer instance is used to process the result set from all the mappers and therefore greater the load a single reducer instance and slower the whole process. We are not exploiting parallelism here, to exploit the same we have to assign the no of reducers explicitly. In runtime we can specify the no of reducers as

```
hadoop jar /home/training/usecase/wordcount/wordcount.jar  
com.bejoy.samples.wordcount.WordCount -D mapred.reduce.tasks=15 /projects/wordcount/input/ /projects/wordcount/output/
```

The key point to be noted here is that the no of output files is same as the no of reducers used as every reducer would produce its own output file. All these output files would be available in the hdfs output directory we assigned in the run command. It would be a cumbersome job to combine all these files manually to obtain the result set. For that Hadoop has provided a get merge command

```
hadoop fs -getmerge /projects/wordcount/output/ /home/training/usecase/wordcount/output/WordCount.txt
```

This command would combine the contents of all the files available directly within the /projects/wordcount/output/ hdfs directory and write the same to /home/training/usecase/wordcount/output/WordCount.txt file in LFS

You can find the working copy of the word count implementation with hadoop 0.20 API at the following location [word count example with hadoop 0.20](#)

Posted by Bejoy KS at 5:39 AM



+6 Recommend this on Google

#### 66 comments:



**Ratan Kumar Nath** August 31, 2012 at 3:23 AM

Nice example with details, Please add the new api example if possible.

[Reply](#)

[Replies](#)



**Sandy** July 17, 2014 at 6:45 AM

For the latest api, a working example with complete source code and explanation can be found at <http://hadooptuts.com>

[Reply](#)



**Bejoy KS** [September 10, 2012 at 10:06 AM](#)

Hi Ratan

You can find the sample code for mapreduce API @ <http://kickstarthadoop.blogspot.in/2011/05/word-count-example-with-hadoop-020.html>

[Reply](#)



**Arockiaraj Durairaj** [September 11, 2012 at 5:08 PM](#)

Thanks a lot for this article. It is really a kick starter.

[Reply](#)



**Arockiaraj Durairaj** [September 11, 2012 at 5:40 PM](#)

Can u please explain about how the input file is specified for the mapper and who sends line by line to mapper function?

[Reply](#)



**Bejoy KS** [September 12, 2012 at 1:51 PM](#)

Hi Arockiaraj

In a mapreduce program, the JobTracker assigns input splits to each map task based on factors like data locality, slot availability etc. A map task actually processes a certain hdfs block. If you have a large file that comprises of 10 blocks and if your mapped split properties complement with the hdfs block size then you'll have 10 map tasks processing 1 block each.

Now once the mapper has its own share of input based on the input format and certain other properties it is the RecordReader that reads record by record and given them as input to each execution of the map() method. In default TextInputFormat the record reader reads till a new line character for a record.

[Reply](#)



**As Sun Shines** [December 5, 2012 at 6:13 PM](#)

How is default number of reducers chosen by mapreduce framework? Is it according to data load or any configured property?

[Reply](#)



**hanu** [January 12, 2013 at 12:40 AM](#)

Thank you very much... :)

[Reply](#)



**hemanth vijay musunuru** [February 4, 2013 at 3:55 AM](#)

hi,  
what is the type of KEYIN ??? what do we call it?? datatype, class, interface etc???

[Reply](#)



**hemanth vijay musunuru** [February 4, 2013 at 3:57 AM](#)

public class Mapper what does KEYIN mean? I have searched in source code but unable to find declaration of KEYIN

[Reply](#)



**Bejoy KS** [March 25, 2013 at 6:02 AM](#)

Hi Hemanth

By KEYIN, I'm assuming you are referring to input key in mapper.  
Here I'm using the default TextInputFormat and for that the default Key is LongWritable, which is an offset value from beginning of the file.  
KEYIN is a subclass of Writable.

[Reply](#)



**Jey** [April 4, 2013 at 3:07 AM](#)

Hello,

Please help me to understand what is BigData and purpose with example.

Thanks

[Reply](#)



**saikumar allaka** April 20, 2013 at 6:39 AM

Default reducers will be 1, but you can still change it based on your requirement.

[Reply](#)



**David Ben Shimon** April 27, 2013 at 5:12 AM

Hello,

Thanks a lot for a clear overview. I have a question - What happened if I wish to output the result from a reducer to lets say two different files with some logic related to that. Something like - mapper is reading, reducer accepts those reads, generate two different lists and write those lists into two different outputs/files - one for each list.

Thanks a lot

David

[Reply](#)



**Muthu Krishnan** May 12, 2013 at 11:09 PM

Check out the visual explanation I made  
<http://bit.ly/13s2Tf0>

[Reply](#)



**Diva Dollar** May 23, 2013 at 6:48 AM

Nice article. I need to find out how one can extend this example to doing Word Count on an xml file.

[Reply](#)



**Chintan Desai** June 23, 2013 at 12:24 PM

this is awesome; thanks for helping the community.

[Reply](#)



**Ahmed Abo Bakr** July 3, 2013 at 8:07 AM

very nice tutorial,, i found it very useful thank you,

[Reply](#)



**Mohamed Azmy** August 8, 2013 at 12:14 PM

nice

[Reply](#)



**me** September 4, 2013 at 12:56 AM

I tried the code , it works for text file for both inside and outside and inside HDFS . Is there any difference in term of speed and architecture . Please assist me ? Thanks.

[Reply](#)



**Shiva Reddy** September 10, 2013 at 4:16 AM

Really good piece of knowledge, I had come back to understand regarding your website from my friend Sumit, Hyderabad And it is very useful for who is looking for [Hadoop Online Training](#). I found a best [Hadoop Online Training](#) demo class.

[Reply](#)



**Shehar bano Shafqat** September 19, 2013 at 11:03 PM

perfectarticle to take a start/

[Reply](#)



**shyamala** October 23, 2013 at 5:54 AM

Thank a lot.It is really a kick starter.

[Reply](#)**Jana KS** October 30, 2013 at 10:23 PM

Very useful Thank You

[Reply](#)**Jana KS** October 30, 2013 at 11:59 PM

Hello Dude

I am Fresher in Hadoop. What about Future Vacancies for Hadoop Technology? Reply Must

[Reply](#)**Ashish Chaudhary** February 6, 2014 at 9:50 PMHey friends here are some good tutorials on hadoop 2.2.0 <http://www.javatute.com/javatute/ViewPostByLabel?label=hadoop>[Reply](#)**tehniat mirza** March 16, 2014 at 9:14 AM

'This is really very nice tutorial to have the basic understanding of map reduce function.Thanks a lot.

[Reply](#)[Replies](#)**Content Dev II** May 6, 2014 at 3:30 AM

Nice

**besantvignesh M** July 3, 2014 at 9:47 PMThanks.Recording to your posts very useful for my project.  
[Hadoop Training in Chennai](#)[Reply](#)**Jijo John** July 1, 2014 at 5:12 PM

Very good document for reference for a newbie in hadoop world. Counts words using unix scripts are not fun any more :P

Expecting more and more illustrative examples.

[Reply](#)**Sandy** July 17, 2014 at 6:47 AMFor the latest api, a working example with complete source code and explanation can be found at <http://hadooptuts.com><http://hadooptuts.com> is great resource for BigData Hadoop newbies[Reply](#)**besantvignesh M** July 21, 2014 at 7:45 PM

thanks a lot.

[Hadoop Training in Chennai](#)[Reply](#)**sundara rami reddy** July 22, 2014 at 8:39 PM

This is a great inspiring tutorials on hadoop.I am pretty much pleased with your good work.You put really very helpful information. Keep it up

[Hadoop Training in hyderabad](#)[Reply](#)**Dosi Sagar** September 27, 2014 at 7:09 PM

This comment has been removed by the author.

[Reply](#)





**VMD** October 7, 2014 at 1:59 PM

Nice Explanation,Excellent details,solve some doubts,thanks.  
Keep it up. :)

[Reply](#)



**Kristem Adam** October 18, 2014 at 4:59 AM

Hadoop is an open source tool, so it has multiple benefits for developers and corporate as well.Anobody intrested in Hadoop Training so please check <https://intellipaat.com/>

[Reply](#)



**Joseph A. Wallace** November 5, 2014 at 5:52 AM

Thanks for InformationHadoop Course will provide the basic concepts of MapReduce applications developed using Hadoop, including a close look at framework components, use of Hadoop for a variety of data analysis tasks, and numerous examples of Hadoop in action. This course will further examine related technologies such as Hive, Pig, and Apache Accumulo. [HADOOP Online Training](#)

[Reply](#)



**Madrid Software** November 6, 2014 at 1:29 AM

Nice blog,  
you have explained map reduce in very nice way, it helps most of the students who wants to learn big data hadoop.  
We are also providing [Hadoop training in Delhi](#) and our trainers are working professionals having approx 4 to 5 year experience.

[Reply](#)



**Tejasvi Gaurav** December 2, 2014 at 7:14 AM

You didn't explain the driver class properly. I'm surprised no one else has said anything about it. Please add some more information about that.

[Reply](#)



**ANIL PATEL** December 3, 2014 at 6:16 AM

Please explain the run method used in Driver class, How is the flow ?

[Reply](#)



**Ashish Dixit** January 21, 2015 at 4:16 AM

This is what I am looking for. Thanks a lot.

[Reply](#)



**vignesh m** February 12, 2015 at 3:32 AM

<http://www.besanttechnologies.com/training-courses/data-warehousing-training/hadoop-training-institute-in-chennai>

[Reply](#)



**arvind saxena** March 27, 2015 at 5:15 AM

Great article! Map-Reduce has served a great purpose, though: many, many companies, research labs and individuals are successfully bringing Map-Reduce to bear on problems to which it is suited: brute-force processing with an optional aggregation. But more important in the longer term, to my mind, is the way that Map-Reduce provided the justification for re-evaluating the ways in which large-scale data processing platforms are built (and purchased!). Learn more at <https://intellipaat.com/hadoop-online-training/>

[Reply](#)



**harika goud** April 9, 2015 at 4:25 AM

Hi  
Really very nice blog.Very informative.Thanks for sharing the valuable information.Recently I visited [www.hadooponlinetutor.com](http://www.hadooponlinetutor.com).They are offering hadoop videos at \$20 only.The videos are really awesome.

[Reply](#)



**Niranjan Patil** April 10, 2015 at 8:39 PM

hello

Very nice explanation thanks

I am new to map reduce programming. How to calculate total count of words. I want output as sum/total\_count.

here in this example total sum is 12 so for each key we divide it with respective sum of that word.

Example:

for apple 4/12  
for mango 2/12  
etc

so in output i want like this  
apple 0.33  
mango 0.16  
etc

could any one tell me how should i achieve this i am really struggling a lot.

thanks

[Reply](#)



**venki mosha** April 23, 2015 at 10:23 PM

Giving good information. it will help me lot visualpath is one of the best training institute in hyderabad ameerpeth. [lombardi bpm](#)

[Reply](#)



**Kalyan Hadoop** May 1, 2015 at 5:45 PM

Best Big Data Hadoop Training in Hyderabad @ Kalyan Orient

Follow the below links to know more knowledge on Hadoop

WebSites:

=====

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

Videos:

=====

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

[Hadoop Training in Hyderabad](#)

Best Big Data [Hadoop Training in Hyderabad](#) @ Kalyan Orient

[Reply](#)



**Eduwizz Online Trainings** May 7, 2015 at 6:48 AM

EDUWIZZ provides an excellent job opportunity in [Hybris Training](#) for JAVA professionals who are seeking for job or looking to change to latest and advanced technologies.

[Reply](#)



**JpaSolutions.in** May 22, 2015 at 3:31 AM

As CEOs across the globe grapple with issues from talent acquisition and retention to the need for greater employee productivity, a study by KPMG shows that HR has a massive opportunity to drive significant business value. To know more about , Visit [Big Data training in chennai](#)

[Reply](#)



**Virat Harish** July 26, 2015 at 10:03 PM

hello plz assist me how to print a particular word in op file

[Reply](#)



**peterjohn** August 3, 2015 at 3:12 AM

I really enjoy the blog.Much thanks again. Really Great.  
Very informative article post.Really looking forward to read more. Will read on...

[sap online training](#)  
[sap sd online training](#)  
[hadoop online training](#)  
[sap-crm-online-training](#)

[Reply](#)



**peterjohn** August 5, 2015 at 7:05 AM

I really enjoy the blog.Much thanks again. Really Great.  
Very informative article post.Really looking forward to read more. Will read on...

[oracle online training](#)  
[sap fico online training](#)  
[dotnet online training](#)  
[qa-qtp-software-testing-training-tutorial](#)

[Reply](#)



**syed s** August 8, 2015 at 9:55 AM

I was reading your blog this morning and noticed that you have a awesome resource page. I actually have a similar blog that might be helpful or useful to your audience.

Regards

[sap sd and crm online training](#)  
[sap online tutorials](#)  
[sap sd tutorial](#)  
[sap sd training in ameerpet](#)

[Reply](#)



**Bay Max** August 10, 2015 at 2:31 AM

There are lots of information about latest technology and how to get trained in them, like [Hadoop Training Chennai](#) have spread around the web, but this is a unique one according to me. The strategy you have updated here will make me to get trained in future technologies([Hadoop Training in Chennai](#)). By the way you are running a great blog. Thanks for sharing this.

[Reply](#)



**Venu -blogs** August 28, 2015 at 1:30 PM

Great example, tthanks to explain about word count. This post shows you are too [experienced bigdata analyst](#), please share more tips like this.

[Reply](#)



**Shashaa Tirupati** September 1, 2015 at 6:34 AM

That is an informative post. Thank you so much.

Shashaa

[HTML5 Training in Chennai](#) | [HTML5 Training in Chennai](#)

[Reply](#)



**Ravindra Reddy** September 21, 2015 at 12:48 AM

Best kits online trainings,thanks for sharing

[Oracle DBA Online Training institute](#)

[Oracle SOA Online Training institute](#)

[SalesForce Online Training institute](#)

[SAP ABAP Online Training institute](#)

[SAP Basis Online Training institute](#)

[SAP Bw Hana Online Training institute](#)

[Reply](#)



**Jannik Andrew** September 23, 2015 at 10:02 AM

The Hadoop tutorial you have explained is most useful for begineers who are taking [Hadoop Administrator Online Training](#)  
Thank you for sharing Such a good tutorials on Hadoop

[Reply](#)



**Raksha** September 24, 2015 at 3:58 AM

This informative post helped me a lot in training my students. Thanks so much.  
[HTML5 Training in T Nagar](#) | [HTML5 Training in T Nagar](#)

[Reply](#)



**kits online** September 24, 2015 at 5:00 AM

I was reading your blog and it gives a lot of information to us.Thanks for sharing the post..!

[QTP &QC Training](#)

[Microstrategy Training](#)

[MSBI Training](#)

[Oracle 11g RAC Training](#)

[Reply](#)



**TEK CLASSES** September 29, 2015 at 3:54 AM

plenty of new stuff proper here!

<http://www.tekclasses.com/>

[Reply](#)



**TEK CLASSES** September 29, 2015 at 4:01 AM

Very thoughtful information for freshers and starters

<http://www.tekclasses.com/>

[Reply](#)



**Shashaa Tirupati** October 1, 2015 at 4:35 AM

This is exactly what I was searching for. Awesome post. Thanks a bunch. Helped me in taking class for my students. Wish to follow your posts, keep writing! God Bless!

Shashaa

[Dot Net training](#) | [Dot Net training](#) | [Dot Net training](#)

[Reply](#)



**James Brown** October 12, 2015 at 9:04 AM

This is just the information I am finding everywhere. Thanks for your blog, I just subscribe your blog. This is a nice blog..

[online word count](#)

[Reply](#)



**Raksha** November 3, 2015 at 1:52 AM

Hello Admin, thank you for the article. It has helped me during my [Java training in Chennai](#). Fita academy is a [Java training institutes in Chennai](#) that provides training for interested students. So feel free to contact us to join our [Java J2EE training institutes in Chennai](#).

[Reply](#)



**Michael Bucceri** November 3, 2015 at 2:54 PM

Is there a way of doing this without using imports? Have to use IO inputs.

[Reply](#)

Enter your comment...

Comment as:

Select profile...

Publish

Preview

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Simple template. Powered by [Blogger](#).