MapReduce on Hadoop

1. Look at the file to decide on the InputFormat (TextInputFormat)
2. Decide what kind of file you want to use to store the output of MapReduce (TextInputFormat)
3. Create a Mapper Class
4. Create a Reducer Class
5. Create a Driver class to create a JOB, and set the attributes of the JOB.
6. Create a JAR file
7. Run the JAR file as follows:

hadoop jar myapp.jar package.MydriverClass /path/to/input-path/directory-in-hdfs /path/to/output-path/ directory-in-hdfs

InputFormat → FileInputFormat<LongWritable,Text>

```
Public class WordCountMapper extends Mapper<LongWritable,Text,Text,IntWritable>
{
        IntWritable one = new IntWritable(1);

        Public void map(LogWritable key, Text value, Context context)
        {
                String line = value.toString();
                String [] tokens = line.split();
                For (String s : tokens)
                        Context.write(new Text(s), one);
        }
}
```

```
Public class WordCountReducer extends Reducer< Text,IntWritable,Text, IntWritable>
{

        Public void reduce(Text key, Iterable<IntWritable> values, Context context)
        {
                int sum = 0;
                For (v : values)
                {
                        sum += v.get();
                }
                Context.write(key, new IntWritable(sum));
        }
}
```

InputFormat → FileOutputFormat<Text, IntWritable>

```
Public class MyDriverClass {
        Public void main(String [] args) {
                // Create a new Job
                Job job = Job.getInstance();
                job.setJarByClass(MyJob.class);
                // Specify various job-specific parameters
                job.setJobName("myjob");
                job.setInputPath(new Path("in"));
                job.setOutputPath(new Path("out"));
                job.setMapperClass(MyJob.MyMapper.class);
                job.setReducerClass(MyJob.MyReducer.class);
                // Submit the job, then poll for progress until the job is complete
                job.waitForCompletion(true);
        }
}
```