

```

bmscscse@bmscscse-HP-Elite-Tower-600-G9-Desktop-PC: $ spark-shell
25/05/20 11:28:13 WARN Utils: Your hostname, bmscscse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.80 instead (on interface eno1)
25/05/20 11:28:13 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.0.3.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/05/20 11:28:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.124.3.80:4040
Spark context available as 'sc' (master = local[*], app id = local-1747720695950).
Spark session available as 'spark'.
Welcome to

  ____  __
 / ___/ /_
/ /   / __ \
/ /___/ /_/ /
\____/____/

version 3.0.3

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.26)
Type in expressions to have them evaluated.
Type :help for more information.

scala> for (i <- 1 to 100) print(i + " ")
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
5 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
scala> val rdd = spark.sparkContext.textFile("file:/home/bmscscse/Desktop/scala")
rdd: org.apache.spark.rdd.RDD[String] = file:/home/bmscscse/Desktop/scala MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = rdd.flatMap(_.split("\\s+")).map(word => (word.toLowerCase, 1)).reduceByKey(_ + _).filter(_._2 > 4)
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:25

scala> counts.collect().foreach{ case (word, count) => println(s"$word $count") }
spark 6

```