* **val** conf = **new** **SparkConf**().setAppName("wiki\_test")
* *// create a spark config object*
* **val** sc = **new** **SparkContext**(conf)
* *// Create a spark context*
* **val** data = sc.textFile("/path/to/somedir")
* *// Read files from "somedir" into an RDD of (filename, content)pairs.*
* **val** tokens = data.flatMap(\_.split(" "))
* *// Split each file into a list of tokens (words).*
* **val** wordFreq = tokens.map((\_, 1)).reduceByKey(\_ + \_)
* *// Add a count of one to each token, then sum the counts per word type.*
* wordFreq.sortBy(s => -s.\_2).map(x => (x.\_2, x.\_1)).top(10)
* *// Get the top 10 words. Swap word and count to sort by count.*

### SparkSession

[SparkSession](https://sparkbyexamples.com/spark/sparksession-explained-with-examples/) introduced in version 2.0, is an entry point to underlying Spark functionality in order to programmatically use Spark RDD, DataFrame, and Dataset. It’s object *spark* is default available in spark-shell.

Creating a SparkSession instance would be the first statement you would write to the program with [RDD](https://sparkbyexamples.com/spark-rdd-tutorial/), [DataFrame](https://sparkbyexamples.com/spark/different-ways-to-create-a-spark-dataframe/) and Dataset. SparkSession will be created using SparkSession.builder() builder pattern.

// Create SparkSession

import org.apache.spark.sql.SparkSession

val spark:SparkSession = SparkSession.builder()

.master("local[1]")

.appName("SparkByExamples.com")

.getOrCreate()

RDD creation

RDDs are created primarily in two different ways, first parallelizing an existing collection and secondly referencing a dataset in an external storage system (HDFS, HDFS, S3 and many more).

**sparkContext.parallelize()**

sparkContext.parallelize is used to parallelize an existing collection in your driver program. This is a basic method to create RDD.

//Create RDD from parallelize

val dataSeq = Seq(("Java", 20000), ("Python", 100000), ("Scala", 3000))

val rdd=spark.sparkContext.parallelize(dataSeq)

**sparkContext.textFile()**

Using textFile() method we can read a text (.txt) file from many sources like HDFS, S#, Azure, local e.t.c into RDD.

//Create RDD from external Data source

val rdd2 = spark.sparkContext.textFile("/path/textFile.txt")

RDD Operations

On Spark RDD, you can perform two kinds of operations.

**RDD Transformations**

[Spark RDD Transformations](https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-transformations/) are lazy operations meaning they don’t execute until you call an action on RDD. Since RDDs are immutable, When you run a transformation(for example map()), instead of updating a current RDD, it returns a new RDD.

Some transformations on RDDs are flatMap(), map(), reduceByKey(), filter(), sortByKey() and all these return a new RDD instead of updating the current.

**RDD Actions**

[RDD Action operation](https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-actions/) returns the values from an RDD to a driver node. In other words, any RDD function that returns non RDD[T] is considered as an action. RDD operations trigger the computation and return RDD in a List to the driver program.

Some actions on RDDs are count(),  collect(),  first(),  max(),  reduce()  and more.

RDD Examples

* [Read CSV file into RDD](https://sparkbyexamples.com/apache-spark-rdd/spark-load-csv-file-into-rdd/)
* [RDD Pair Functions](https://sparkbyexamples.com/apache-spark-rdd/spark-pair-rdd-functions/)
* [Generate DataFrame from RDD](https://sparkbyexamples.com/apache-spark-rdd/convert-spark-rdd-to-dataframe-dataset/)