First install SCALA :

Go to <https://www.scala-lang.org/download/2.13.14.html>

And install the scala msi installer.

Make sure to install java 11.0.23 , in the environment variable paste java path and do not include bin in the JAVA\_HOME .

Now include the java bin path in the path variable of system variable and make sure it it on top of paths.

On cmd , check java -version

scala - version

After getting the scala version , type scala on cmd to run scala.

To check if the scala is working type : println("Hello");

Now type this one by one : var a: Int = 12;

var b: Int = 10;

var c: Int = a + b;

val k: Int = 23;

k = 21;

Scala is installed .

Now SPARK:

Go to <https://spark.apache.org/downloads.html> And download 

Now create a folder named ‘spark’ and in that extract the zip file.

In environment variable , under SPARK\_HOME paste path without zip file

(C:\SPARK\spark-3.5.3-bin-hadoop3-scala2.13)

And in the path variable paste the path along with the zip file.

Navigate the bin path to cmd and type spark-shell.

After, run the following command:

val x = spark.read.json("C:\\SPARK\\spark-3.5.3-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json");

x.show()

x.printSchema()

x.select($"name",$"age").show()

x.filter($"age">20).show()

x.select($"age"+1).show()

x.createOrReplaceTempView("people")

val sqlDF = spark.sql("Select \* from people")

spark.sql("SELECT \* FROM global\_temp.people").show()

spark.newSession.sql("SELECT \* FROM global\_temp.people ").show()

case class Person(name: String, age: Long)

val caseClassDS = Seq(Person("Andy", 32)).toDS()

val path = "C:\\SPARK\\spark-3.5.3-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json"

val peopleDS = spark.read.json(path).as[Person]

peopleDS.show()

import spark.implicits.\_

val peopleDF = spark.sparkContext.textFile("C:\\SPARK\\spark-3.5.3-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.txt").map(\_.split(",")).map(attributes => Person(attributes(0), attributes(1).trim.toInt)).toDF()

peopleDF.createOrReplaceTempView("people")

val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")

teenagersDF.map(teenager => "Name: " + teenager(0)).show()

implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]

val a = spark.read.option("header", "true").csv("C:\\SPARK\\spark-3.5.3-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.csv");

a.show()