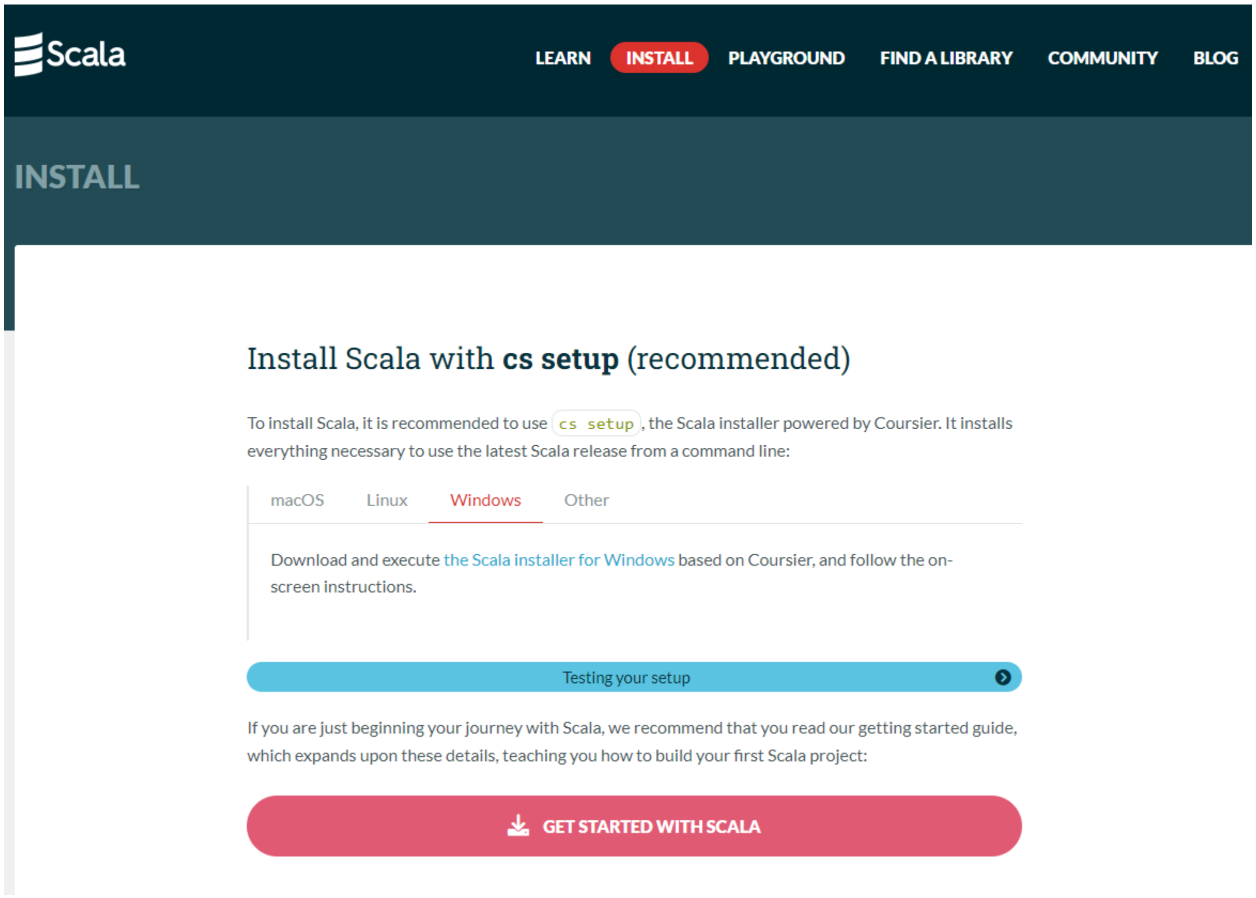


INSTALL

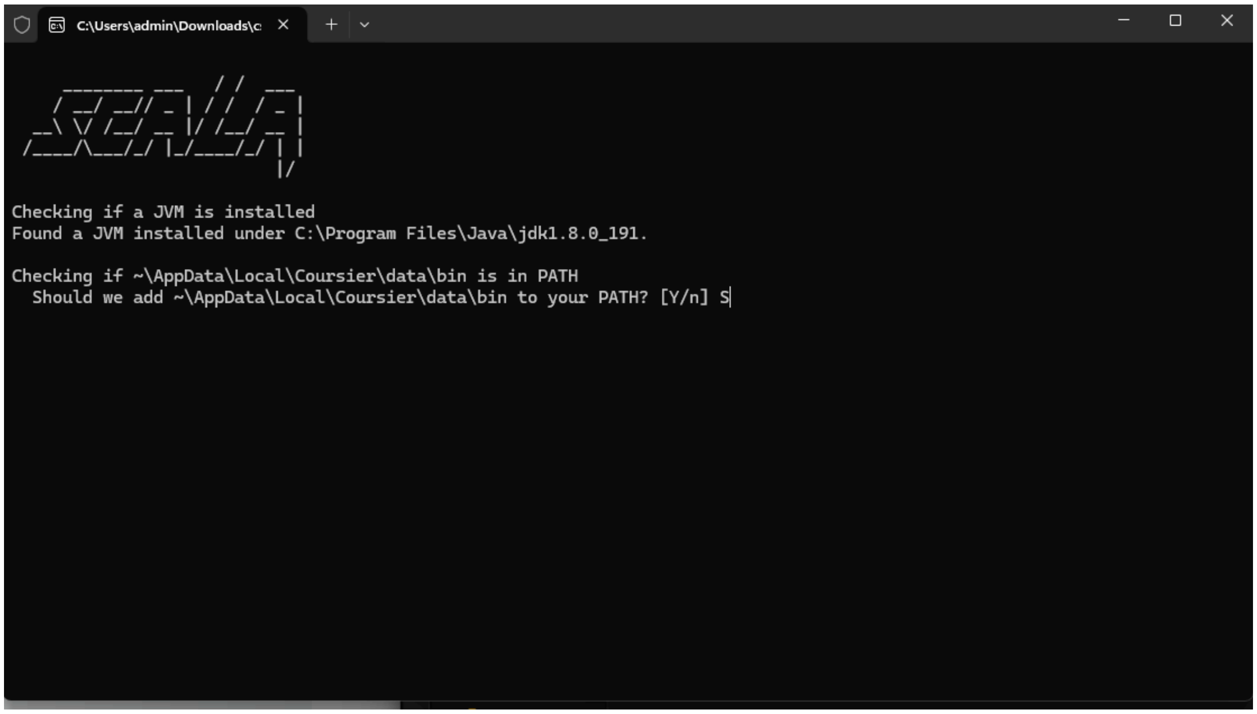
Install Scala with **cs setup** (recommended)

To install Scala, it is recommended to use `cs setup`, the Scala installer powered by Coursier. It installs everything necessary to use the latest Scala release from a command line:

Download and execute the Scala installer for Windows based on Coursier, and follow the on-screen instructions.

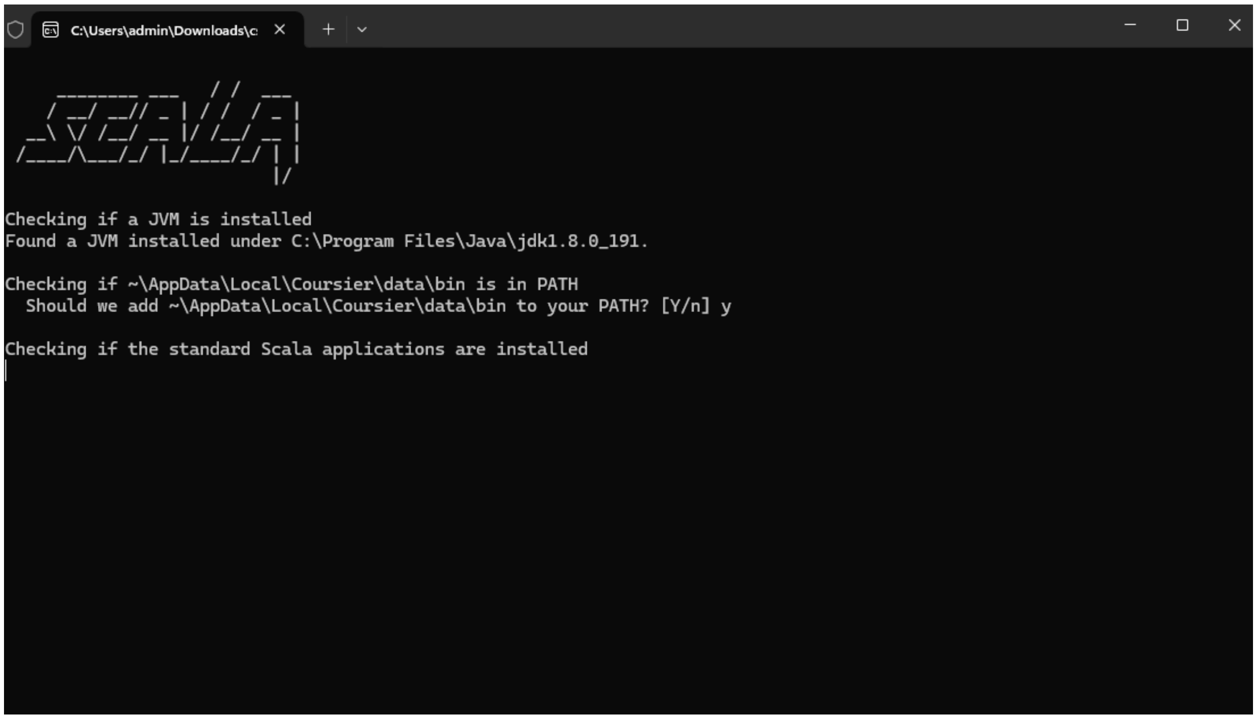
If you are just beginning your journey with Scala, we recommend that you read our getting started guide, which expands upon these details, teaching you how to build your first Scala project:

2-



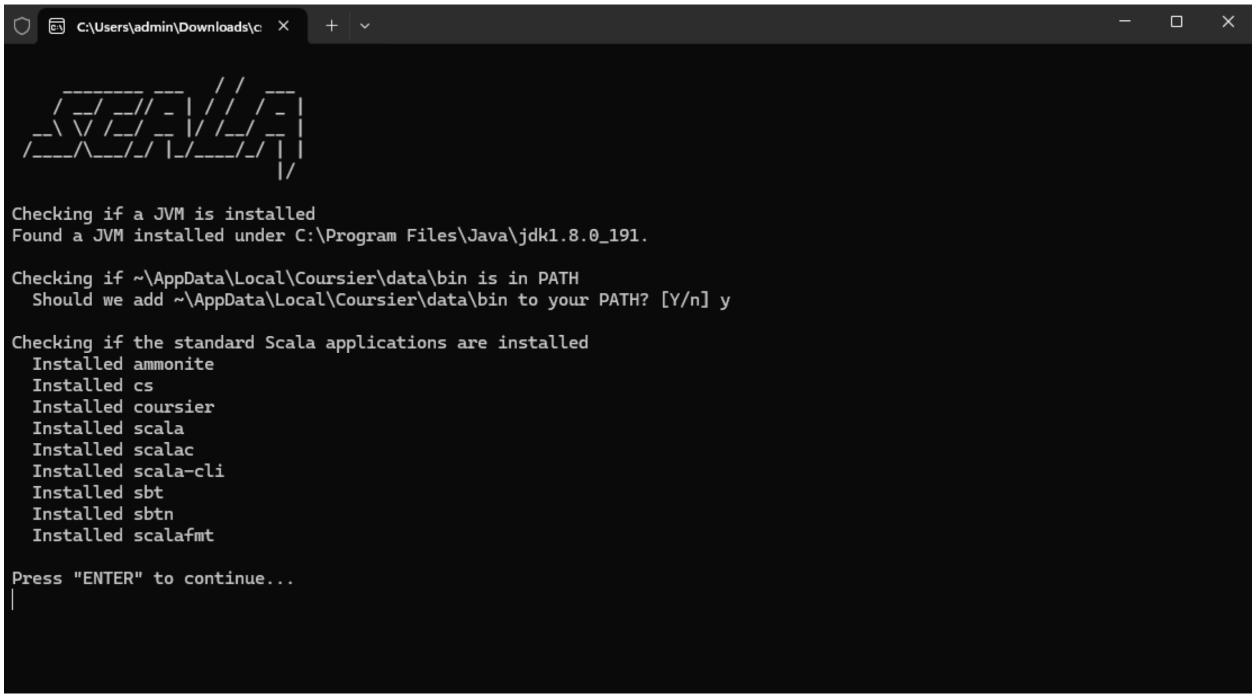
```
     ________ ___   / /  ___
    / __/ __// _ | / /  / _ |
  __\ \/ /__/ __ |/ /__/ __ |
 /____/\___/_/ |_/____/_/ | |
                          |/

Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
  Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] S
```

3-



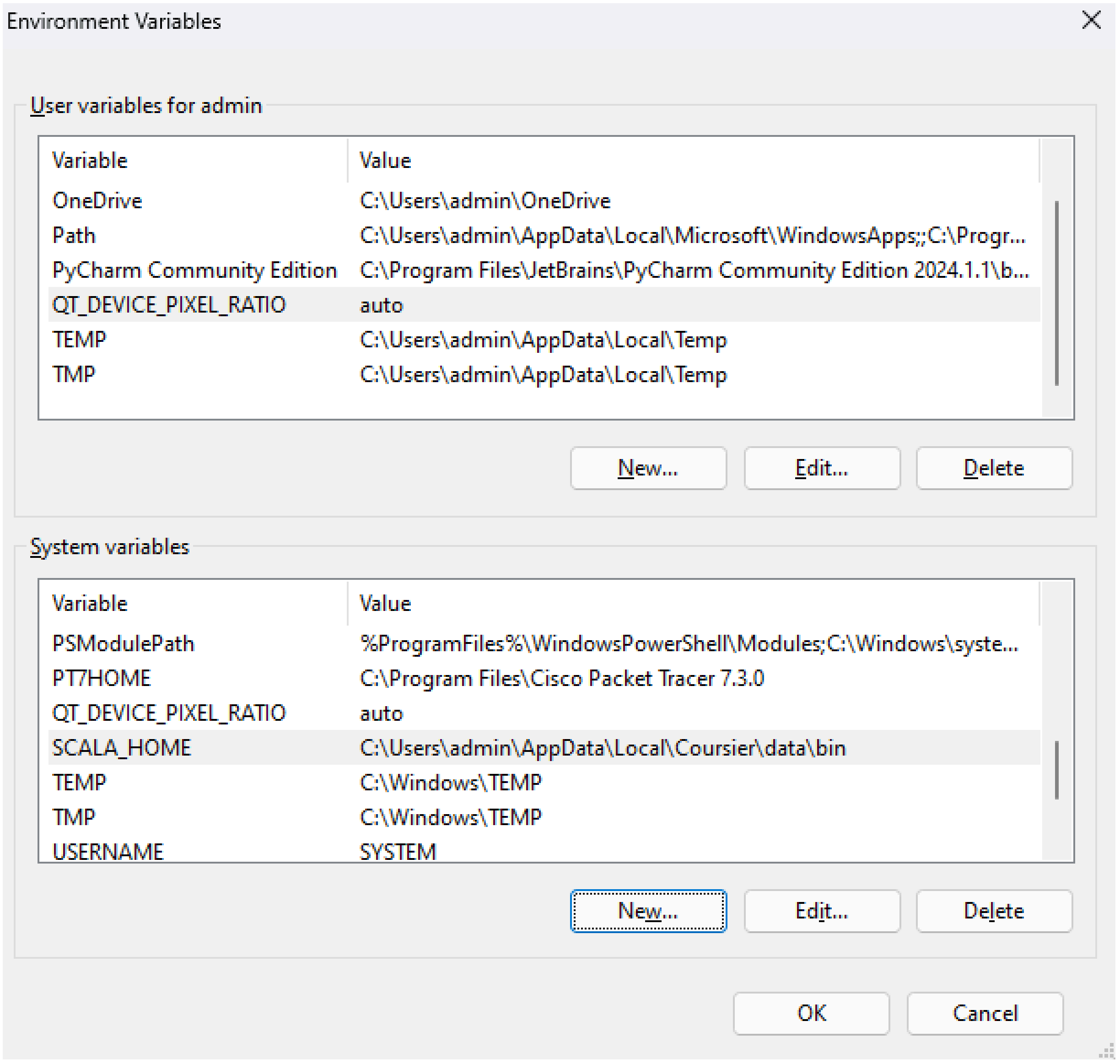
```
     ________ ___   / /  ___
    / __/ __// _ | / /  / _ |
  __\ \/ /__/ __ |/ /__/ __ |
 /____/\___/_/ |_/____/_/ | |
                          |/

Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
  Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] y

Checking if the standard Scala applications are installed
```

4-



```
     ________ ___   / /  ___
    / __/ __// _ | / /  / _ |
  __\ \/ /__/ __ |/ /__/ __ |
 /____/\___/_/ |_/____/_/ | |
                          |/

Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
  Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] y

Checking if the standard Scala applications are installed
  Installed ammonite
  Installed cs
  Installed coursier
  Installed scala
  Installed scalac
  Installed scala-cli
  Installed sbt
  Installed sbtn
  Installed scalafmt

Press "ENTER" to continue...
```

5-Now, make sure to set the environmental path variables



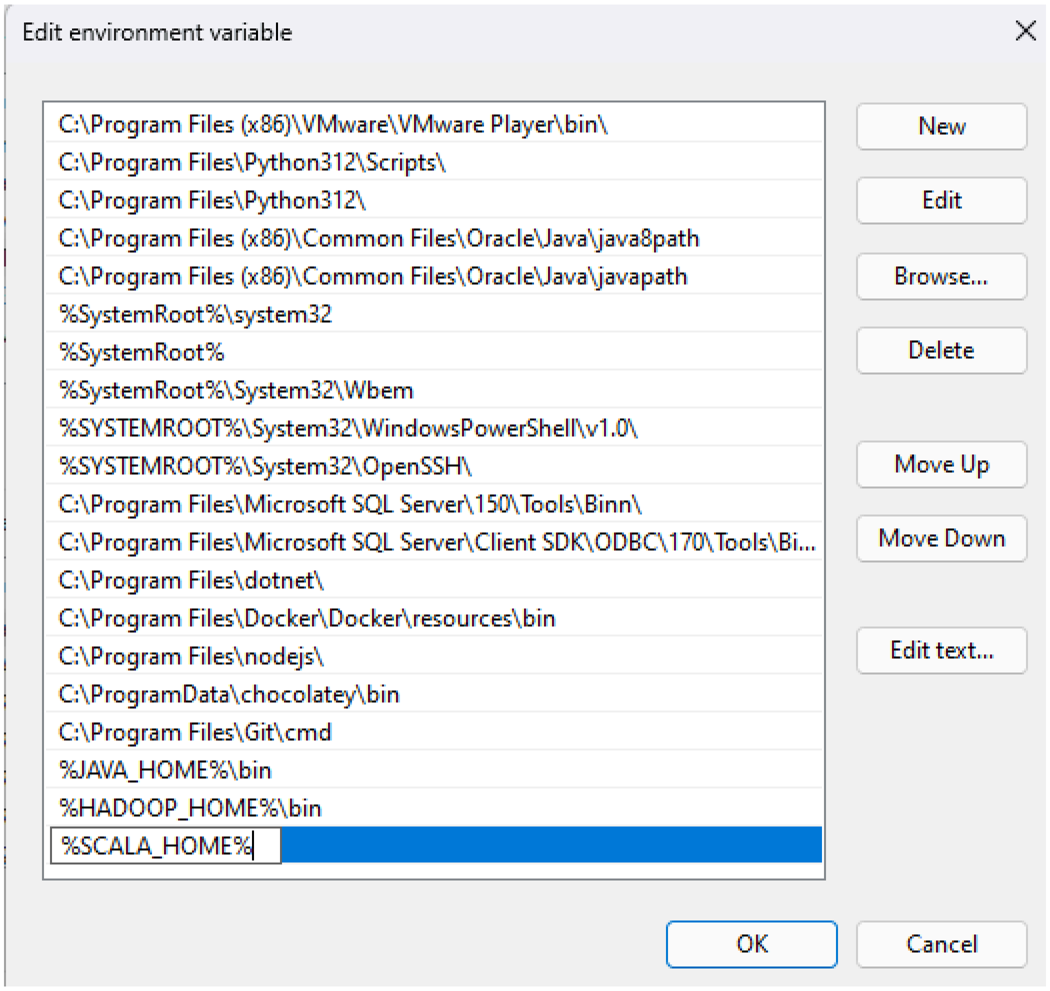
User variables for admin

Variable	Value
OneDrive	C:\Users\admin\OneDrive
Path	C:\Users\admin\AppData\Local\Microsoft\WindowsApps;;C:\Progr...
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.1\b...
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\admin\AppData\Local\Temp
TMP	C:\Users\admin\AppData\Local\Temp

System variables

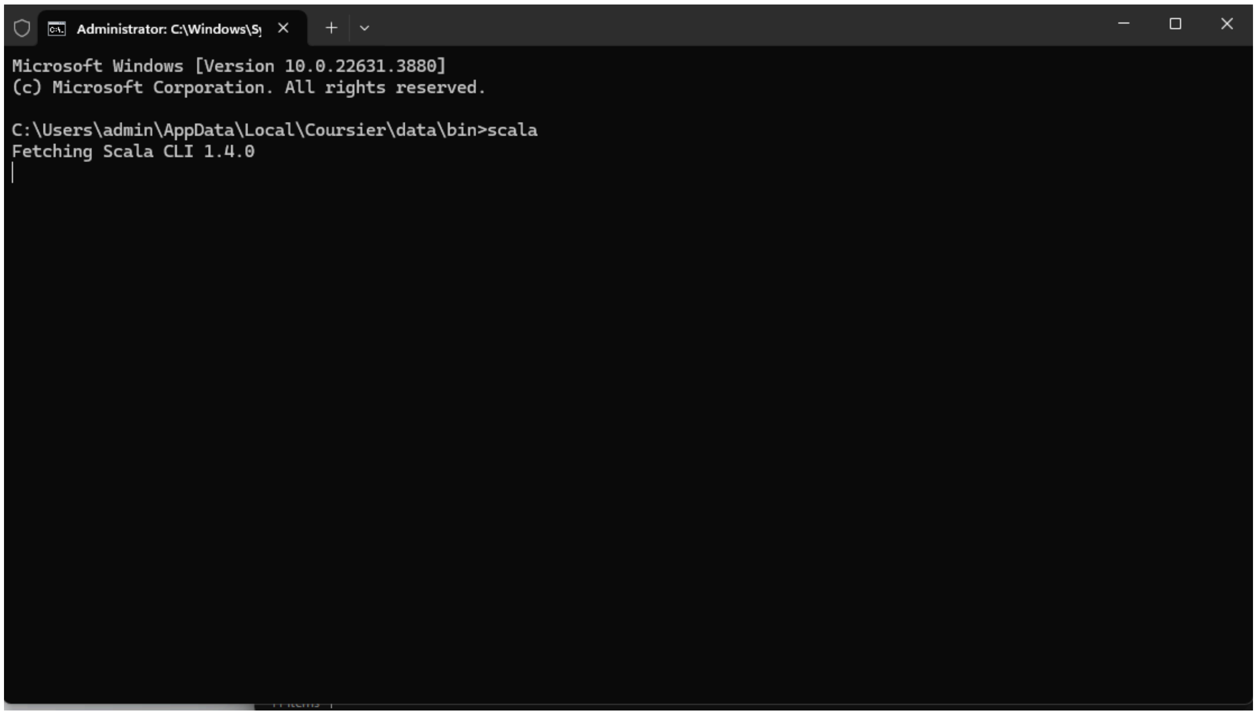
Variable	Value
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\Windows\syste...
PT7HOME	C:\Program Files\Cisco Packet Tracer 7.3.0
QT_DEVICE_PIXEL_RATIO	auto
SCALA_HOME	C:\Users\admin\AppData\Local\Coursier\data\bin
TEMP	C:\Windows\TEMP
TMP	C:\Windows\TEMP
USERNAME	SYSTEM

6-



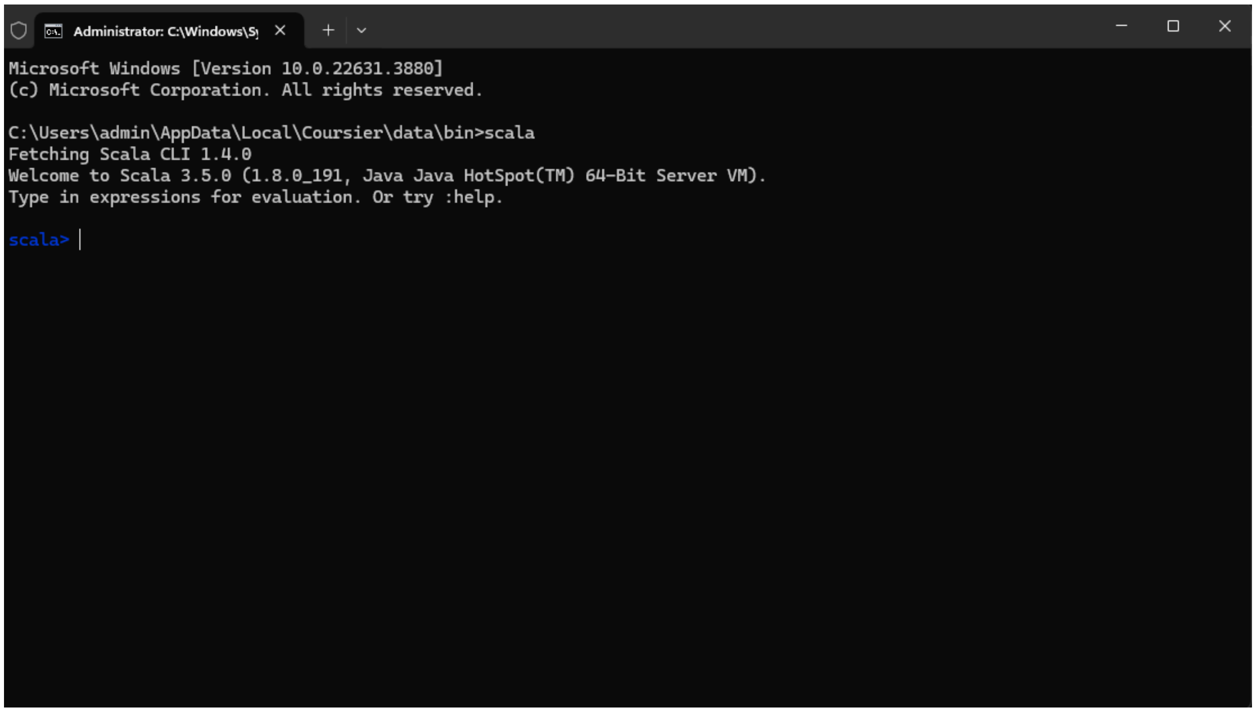
```
C:\Program Files (x86)\VMware\VMware Player\bin\
C:\Program Files\Python312\Scripts\
C:\Program Files\Python312\
C:\Program Files (x86)\Common Files\Oracle\Java\java8path
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Bi...
C:\Program Files\dotnet\
C:\Program Files\Docker\Docker\resources\bin
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin
C:\Program Files\Git\cmd
%JAVA_HOME%\bin
%HADOOP_HOME%\bin
%SCALA_HOME%
```

7-Now, open the following with the help of cmd (enable hidden files if unable to find it as desired) and the following path C:\Users\admin\AppData\Local\Coursier\data\bin to access scala



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
```

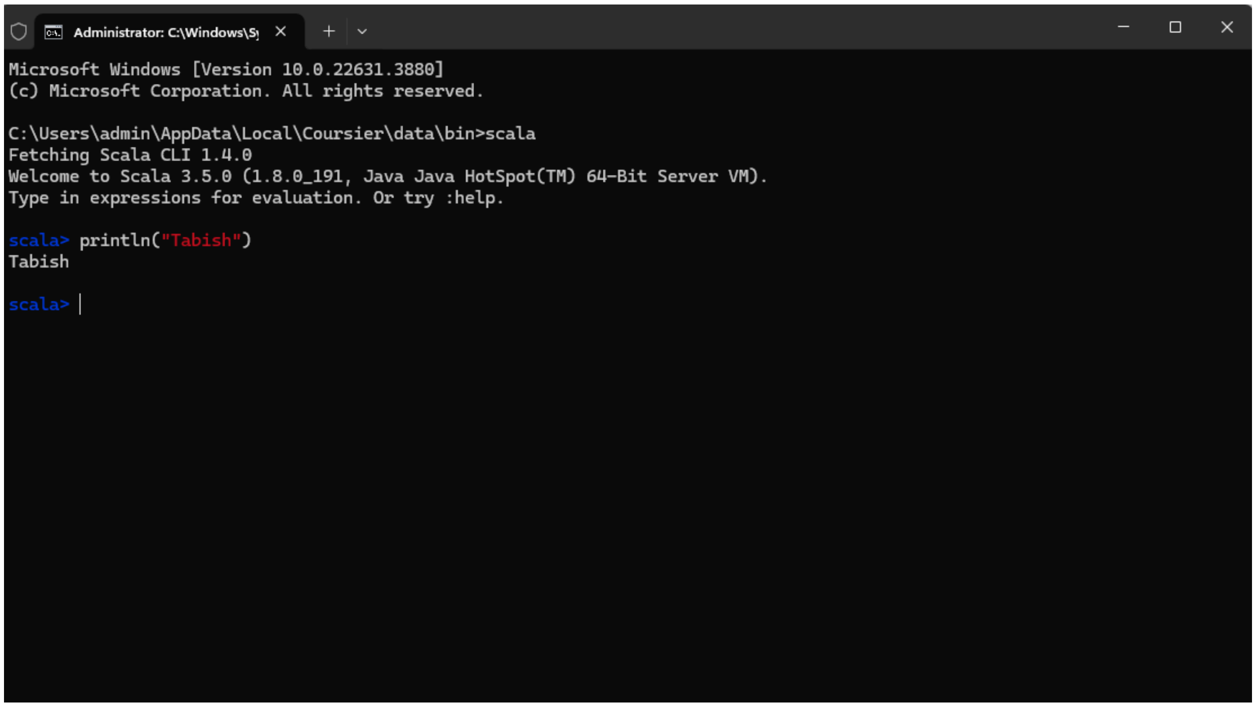
8-Successfully accessed the desired interface



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala>
```

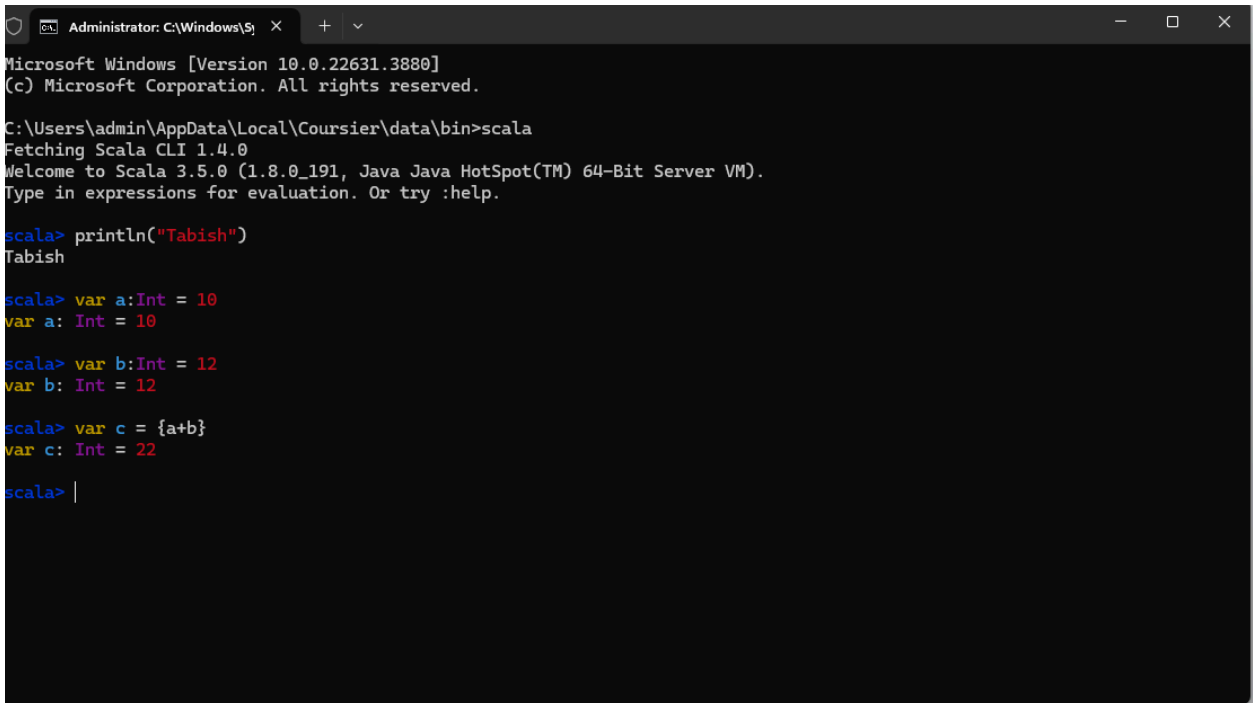
9-Perform the following programs:

A-



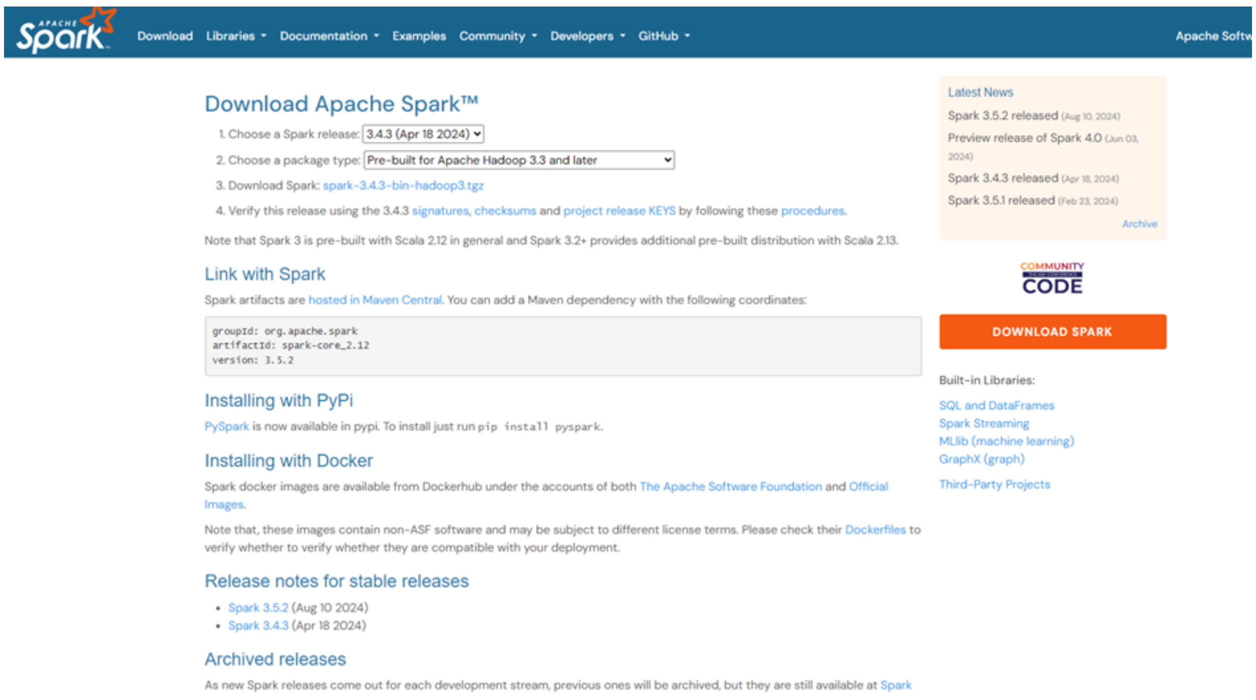
```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> println("Tabish")
Tabish

scala> |
```

B-



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> println("Tabish")
Tabish

scala> var a:Int = 10
var a: Int = 10

scala> var b:Int = 12
var b: Int = 12

scala> var c = {a+b}
var c: Int = 22

scala> |
```

10-Install Spark



Download Apache Spark™

1. Choose a Spark release: 3.4.3 (Apr 18 2024)
2. Choose a package type: Pre-built for Apache Hadoop 3.3 and later
3. Download Spark: spark-3.4.3-bin-hadoop3.tgz
4. Verify this release using the 3.4.3 signatures, checksums and project release KEYS by following these procedures.

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Link with Spark

Spark artifacts are hosted in Maven Central. You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark
artifactId: spark-core_2.12
version: 3.5.2
```

Installing with PyPi

PySpark is now available in pypi. To install just run `pip install pyspark`.

Installing with Docker

Spark docker images are available from Dockerhub under the accounts of both The Apache Software Foundation and Official Images.

Note that, these images contain non-ASF software and may be subject to different license terms. Please check their Dockerfiles to verify whether to verify whether they are compatible with your deployment.

Release notes for stable releases

- Spark 3.5.2 (Aug 10 2024)
- Spark 3.4.3 (Apr 18 2024)

Archived releases

As new Spark releases come out for each development stream, previous ones will be archived, but they are still available at Spark

Latest News

Spark 3.5.2 released (Aug 10, 2024)

Preview release of Spark 4.0 (Jun 03, 2024)

Spark 3.4.3 released (Apr 18, 2024)

Spark 3.5.1 released (Feb 23, 2024)

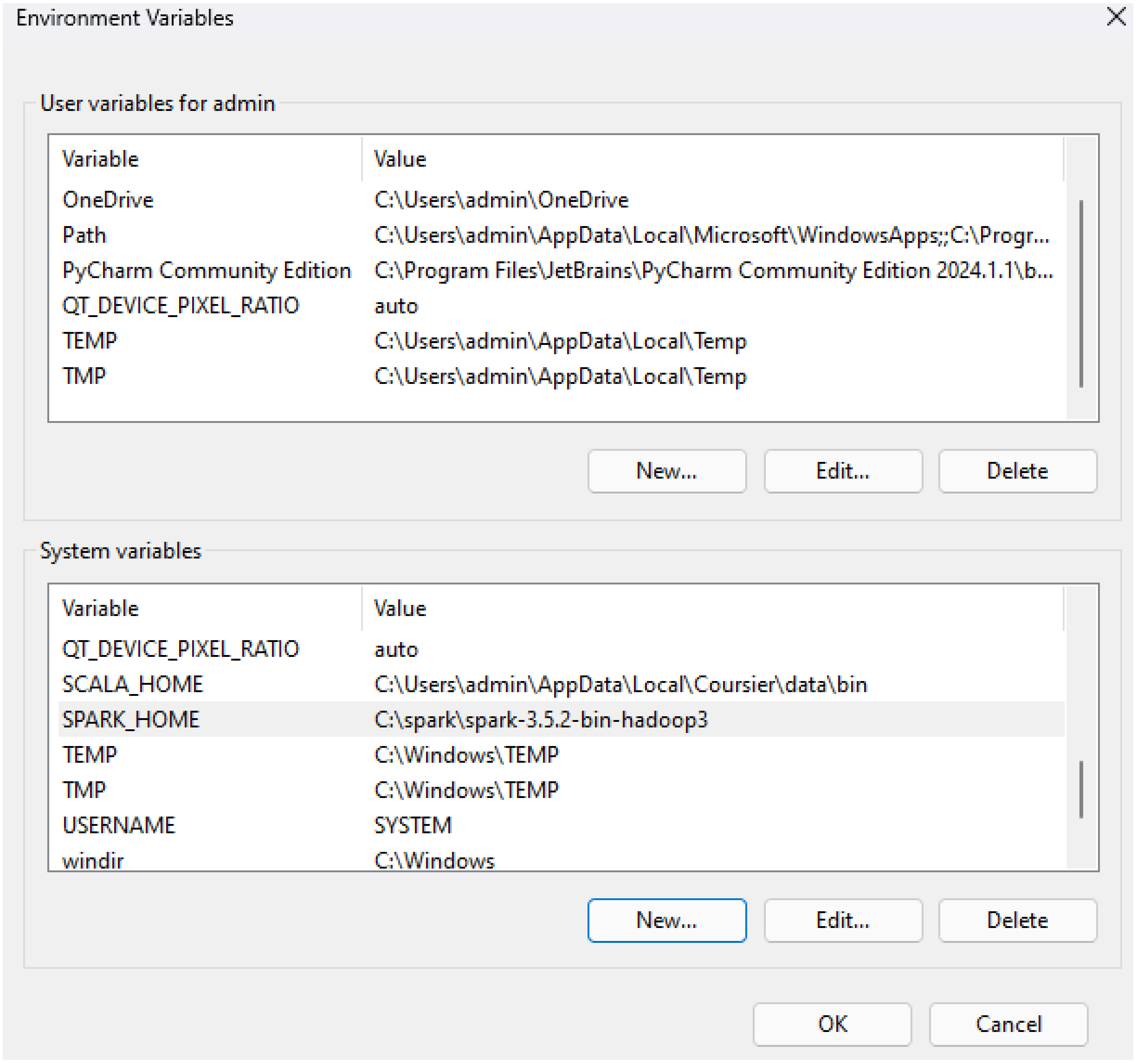
Archive

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Projects

11-Set the environmental variables accordingly (using C:\spark\spark-3.5.2-bin-hadoop3\bin)



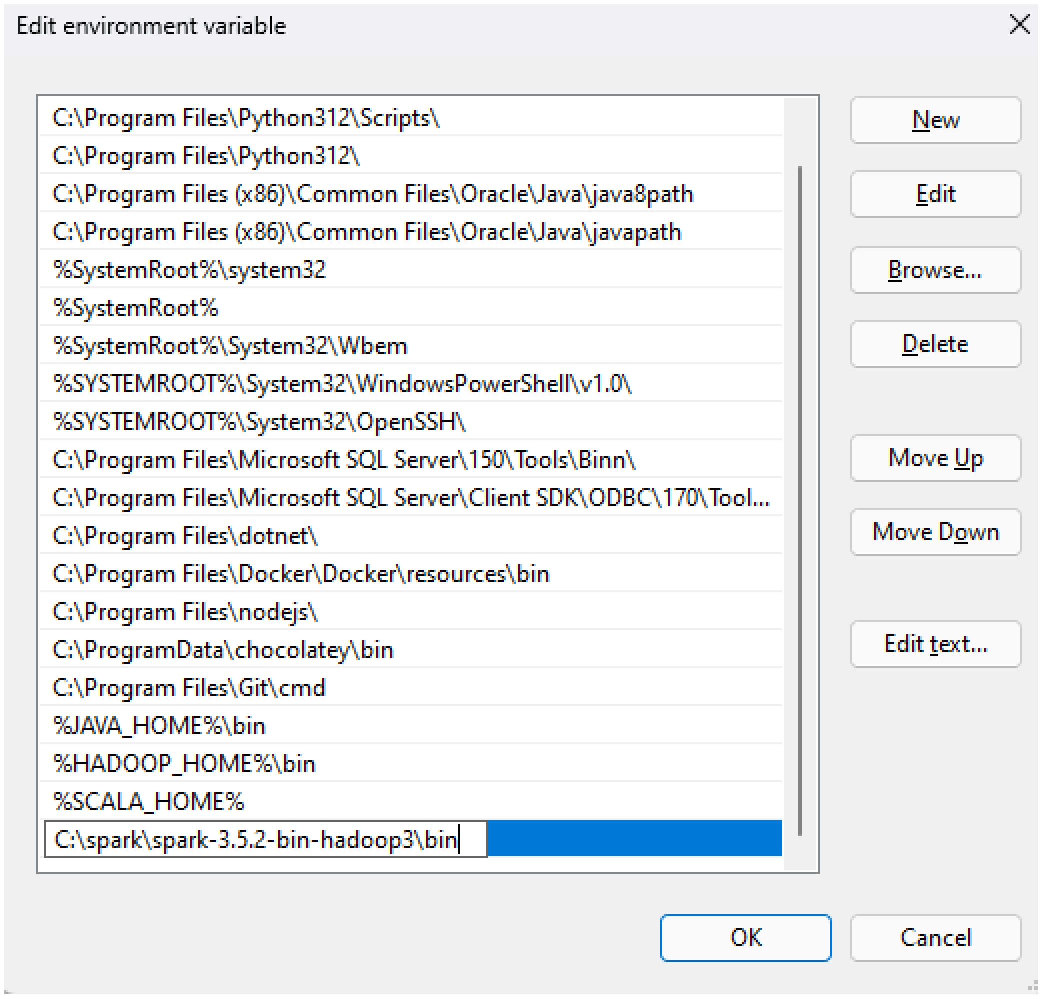
Environment Variables

User variables for admin

Variable	Value
OneDrive	C:\Users\admin\OneDrive
Path	C:\Users\admin\AppData\Local\Microsoft\WindowsApps;;C:\Progr...
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.1\b...
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\admin\AppData\Local\Temp
TMP	C:\Users\admin\AppData\Local\Temp

System variables

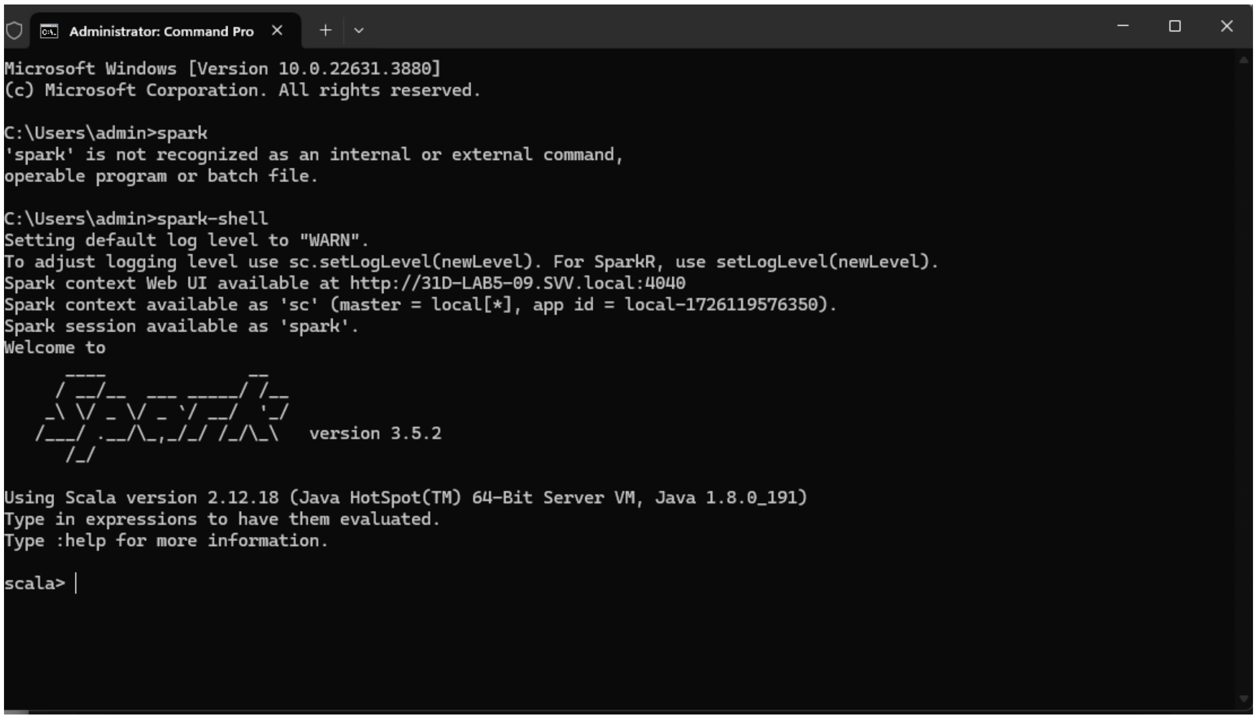
Variable	Value
QT_DEVICE_PIXEL_RATIO	auto
SCALA_HOME	C:\Users\admin\AppData\Local\Coursier\data\bin
SPARK_HOME	C:\spark\spark-3.5.2-bin-hadoop3
TEMP	C:\Windows\TEMP
TMP	C:\Windows\TEMP
USERNAME	SYSTEM
windir	C:\Windows



```
C:\Program Files\Python312\Scripts\
C:\Program Files\Python312\
C:\Program Files (x86)\Common Files\Oracle\Java\java8path
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tool...
C:\Program Files\dotnet\
C:\Program Files\Docker\Docker\resources\bin
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin
C:\Program Files\Git\cmd
%JAVA_HOME%\bin
%HADOOP_HOME%\bin
%SCALA_HOME%
C:\spark\spark-3.5.2-bin-hadoop3\bin
```

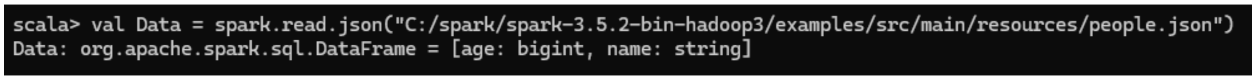
12-Run spark (spark-shell) and perform the following:

A-



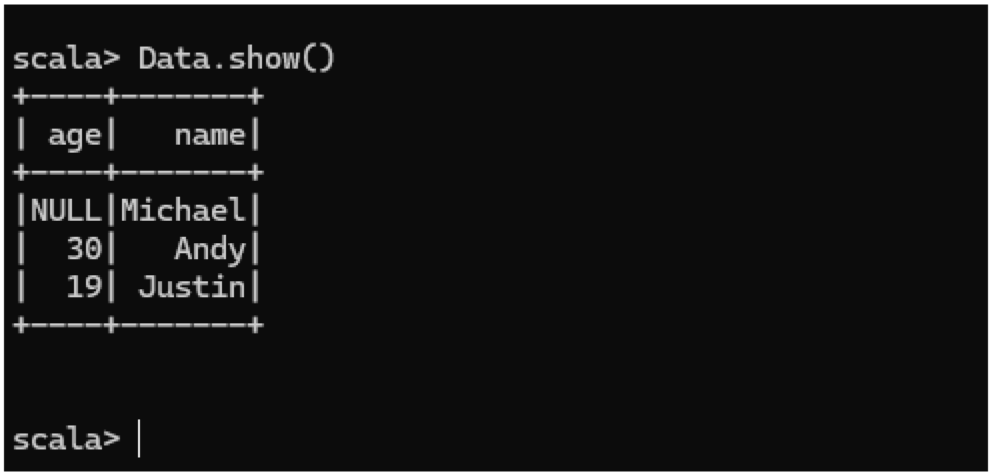
```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>spark
'spark' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\admin>spark-shell
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://31D-LAB5-09.SVV.local:4040
Spark context available as 'sc' (master = local[*], app id = local-1726119576350).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.2
      /_/

Using Scala version 2.12.18 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_191)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

B-



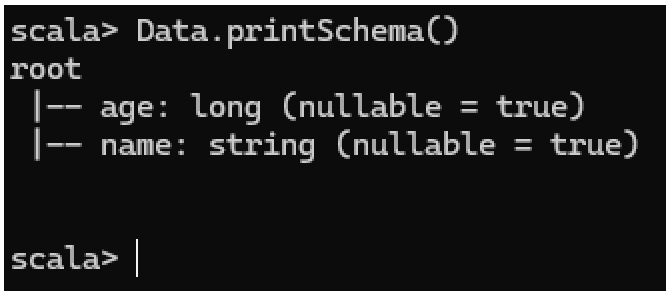
```
scala> val Data = spark.read.json("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json")
Data: org.apache.spark.sql.DataFrame = [age: bigint, name: string]
```

C-



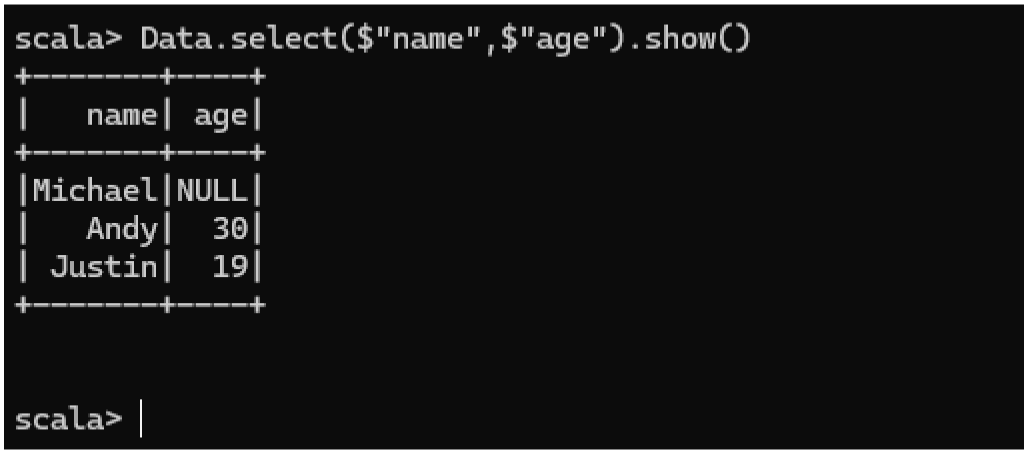
```
scala> Data.show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+


scala>
```

D-



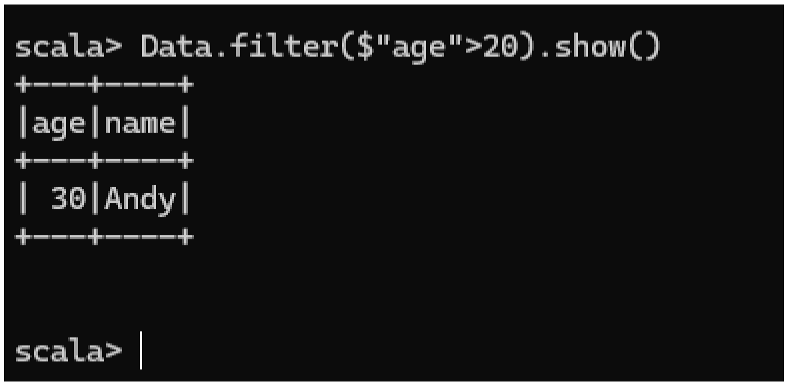
```
scala> Data.printSchema()
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)


scala>
```

E-



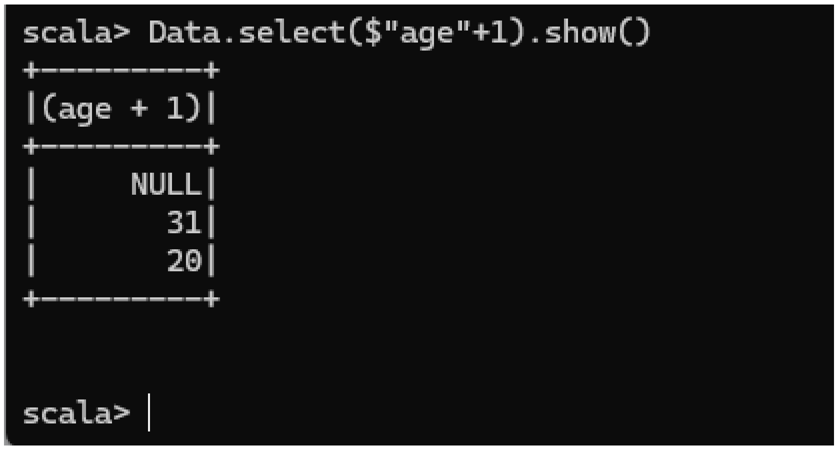
```
scala> Data.select($"name",$"age").show()
+-------+----+
|   name| age|
+-------+----+
|Michael|NULL|
|   Andy|  30|
| Justin|  19|
+-------+----+


scala>
```

F-



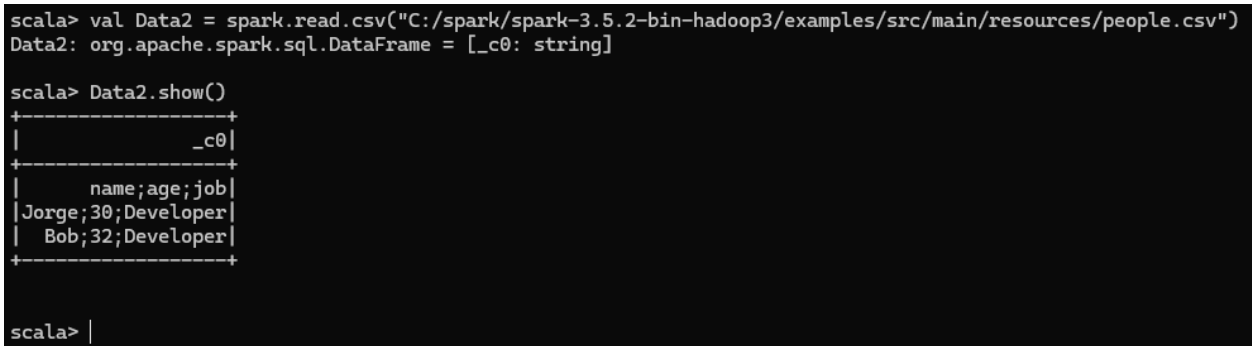
```
scala> Data.filter($"age">20).show()
+---+----+
|age|name|
+---+----+
| 30|Andy|
+---+----+


scala>
```

G-



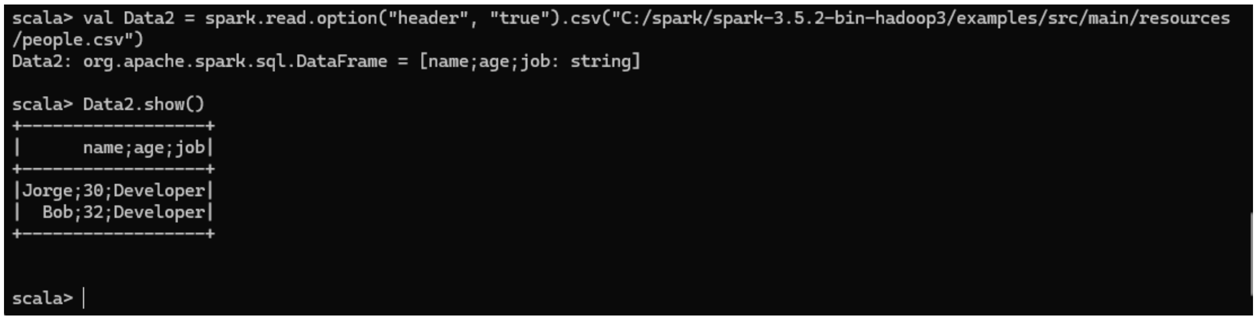
```
scala> Data.select($"age"+1).show()
+---------+
|(age + 1)|
+---------+
|     NULL|
|       31|
|       20|
+---------+


scala>
```

H-



```
scala> val Data2 = spark.read.csv("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.csv")
Data2: org.apache.spark.sql.DataFrame = [_c0: string]

scala> Data2.show()
+------------------+
|               _c0|
+------------------+
|      name;age;job|
|Jorge;30;Developer|
|  Bob;32;Developer|
+------------------+


scala> |
```

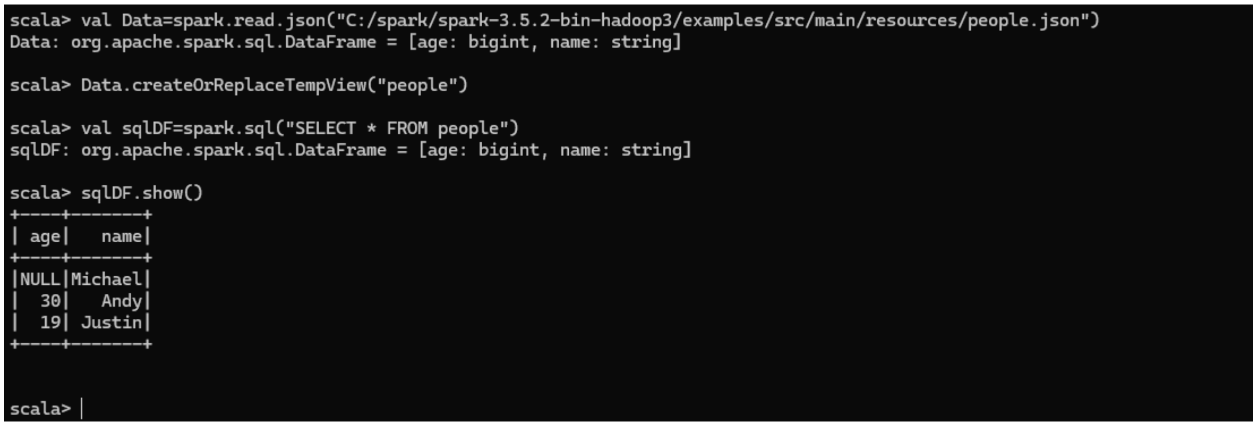
I-



```
scala> val Data2 = spark.read.option("header", "true").csv("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources
/people.csv")
Data2: org.apache.spark.sql.DataFrame = [name;age;job: string]

scala> Data2.show()
+------------------+
|      name;age;job|
+------------------+
|Jorge;30;Developer|
|  Bob;32;Developer|
+------------------+


scala> |
```

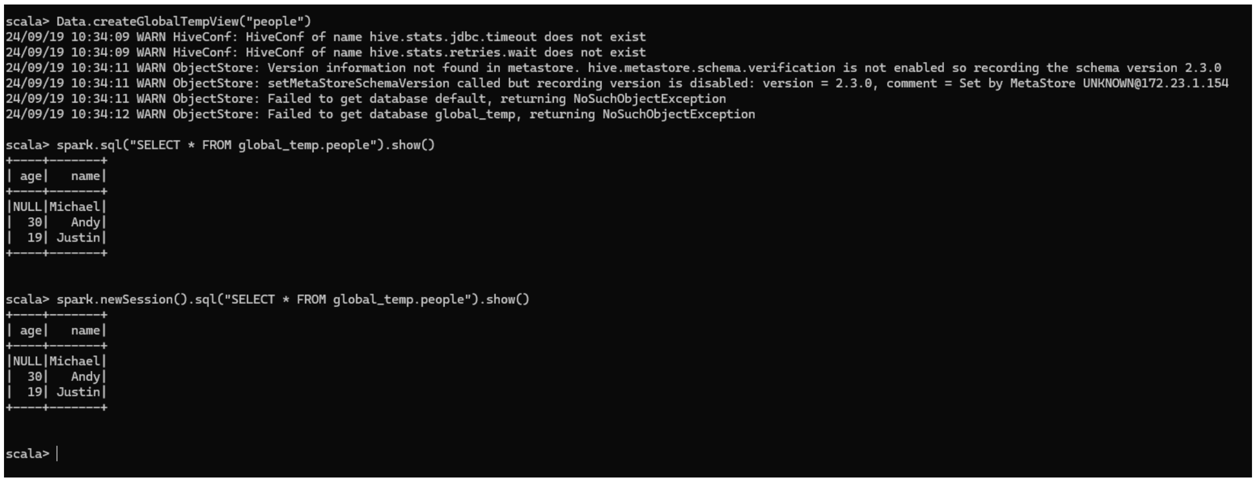
B-Now, further perform the following (refer: [Getting Started - Spark 3.5.2 Documentation (apache.org)]()):

A-



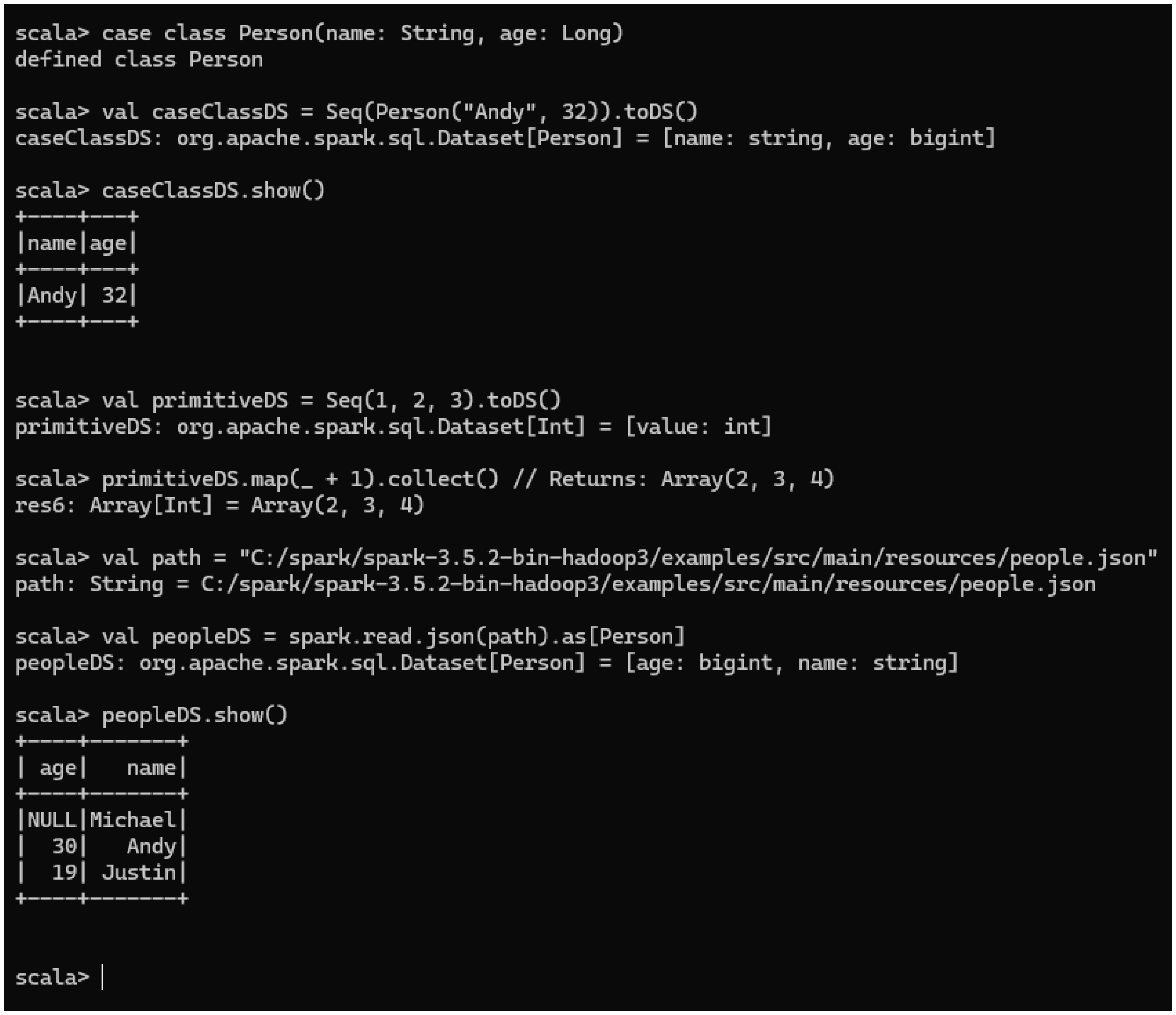
```
scala> val Data=spark.read.json("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json")
Data: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

scala> Data.createOrReplaceTempView("people")

scala> val sqlDF=spark.sql("SELECT * FROM people")
sqlDF: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

scala> sqlDF.show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+


scala> |
```

B-



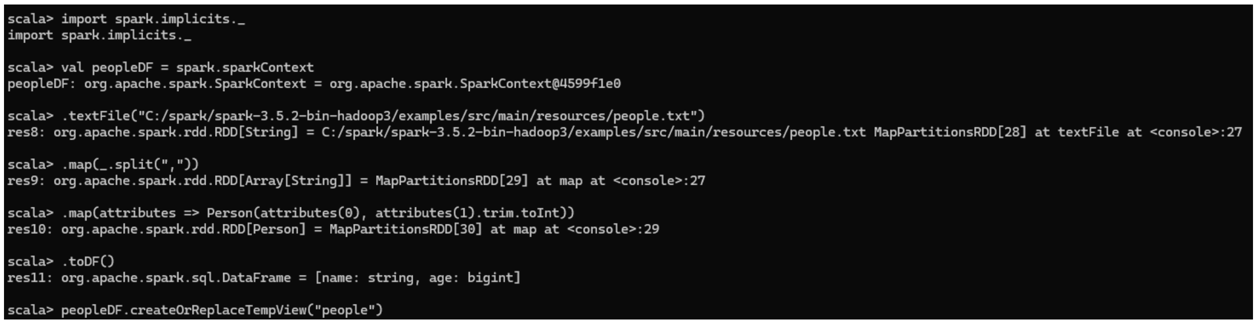
```
scala> Data.createGlobalTempView("people")
24/09/19 10:34:09 WARN HiveConf: HiveConf of name hive.stats.jdbc.timeout does not exist
24/09/19 10:34:09 WARN HiveConf: HiveConf of name hive.stats.retries.wait does not exist
24/09/19 10:34:11 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 2.3.0
24/09/19 10:34:11 WARN ObjectStore: setMetaStoreSchemaVersion called but recording version is disabled: version = 2.3.0, comment = Set by MetaStore UNKNOWN@172.23.1.154
24/09/19 10:34:11 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
24/09/19 10:34:12 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException

scala> spark.sql("SELECT * FROM global_temp.people").show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+


scala> spark.newSession().sql("SELECT * FROM global_temp.people").show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+


scala> |
```

C-

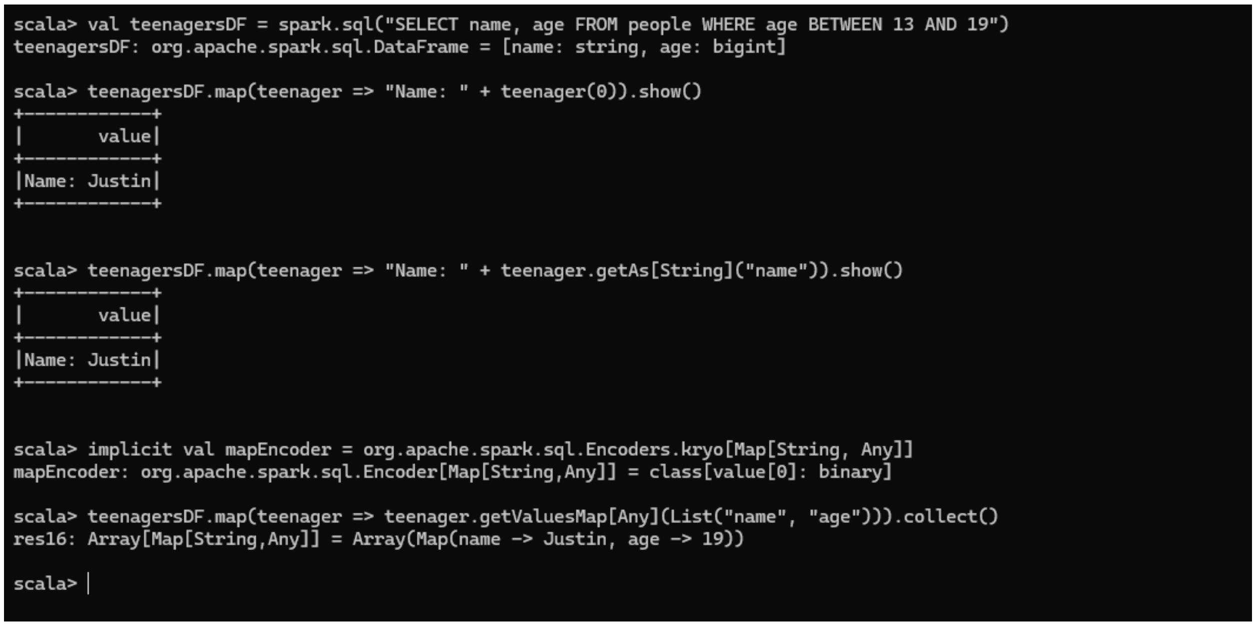


```
scala> case class Person(name: String, age: Long)
defined class Person

scala> val caseClassDS = Seq(Person("Andy", 32)).toDS()
caseClassDS: org.apache.spark.sql.Dataset[Person] = [name: string, age: bigint]

scala> caseClassDS.show()
+----+---+
|name|age|
+----+---+
|Andy| 32|
+----+---+


scala> val primitiveDS = Seq(1, 2, 3).toDS()
primitiveDS: org.apache.spark.sql.Dataset[Int] = [value: int]

scala> primitiveDS.map(_ + 1).collect() // Returns: Array(2, 3, 4)
res6: Array[Int] = Array(2, 3, 4)

scala> val path = "C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json"
path: String = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json

scala> val peopleDS = spark.read.json(path).as[Person]
peopleDS: org.apache.spark.sql.Dataset[Person] = [age: bigint, name: string]

scala> peopleDS.show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+


scala> |
```

D-

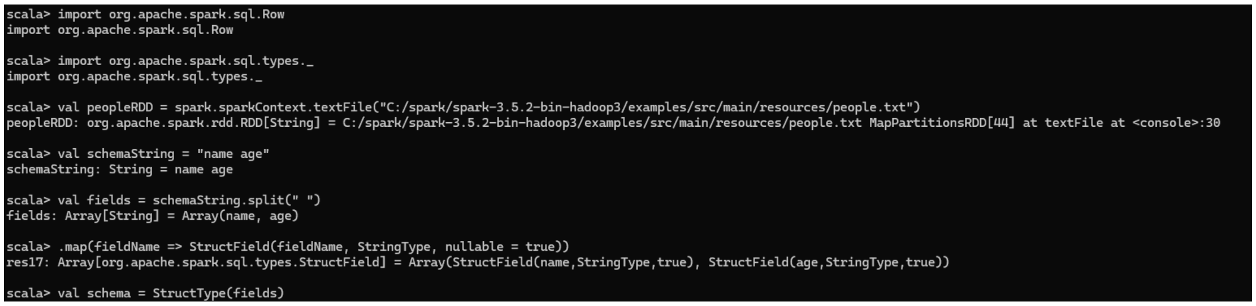


```
scala> import spark.implicits._
import spark.implicits._

scala> val peopleDF = spark.sparkContext
peopleDF: org.apache.spark.SparkContext = org.apache.spark.SparkContext@4599f1e0

scala> .textFile("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt")
res8: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[28] at textFile at <console>:27

scala> .map(_.split(","))
res9: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[29] at map at <console>:27

scala> .map(attributes => Person(attributes(0), attributes(1).trim.toInt))
res10: org.apache.spark.rdd.RDD[Person] = MapPartitionsRDD[30] at map at <console>:29

scala> .toDF()
res11: org.apache.spark.sql.DataFrame = [name: string, age: bigint]

scala> peopleDF.createOrReplaceTempView("people")
```

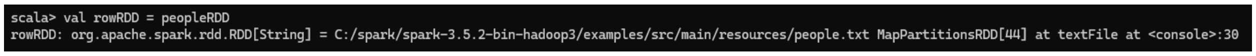


```
scala> val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")
teenagersDF: org.apache.spark.sql.DataFrame = [name: string, age: bigint]

scala> teenagersDF.map(teenager => "Name: " + teenager(0)).show()
+------------+
|       value|
+------------+
|Name: Justin|
+------------+


scala> teenagersDF.map(teenager => "Name: " + teenager.getAs[String]("name")).show()
+------------+
|       value|
+------------+
|Name: Justin|
+------------+


scala> implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]
mapEncoder: org.apache.spark.sql.Encoder[Map[String,Any]] = class[value[0]: binary]

scala> teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name", "age"))).collect()
res16: Array[Map[String,Any]] = Array(Map(name -> Justin, age -> 19))

scala> |
```

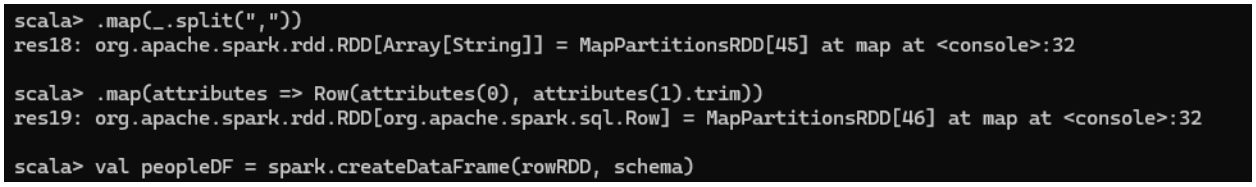
E-



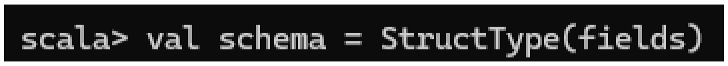
```
scala> import org.apache.spark.sql.Row
import org.apache.spark.sql.Row

scala> import org.apache.spark.sql.types._
import org.apache.spark.sql.types._

scala> val peopleRDD = spark.sparkContext.textFile("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt")
peopleRDD: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[44] at textFile at <console>:30

scala> val schemaString = "name age"
schemaString: String = name age

scala> val fields = schemaString.split(" ")
fields: Array[String] = Array(name, age)

scala> .map(fieldName => StructField(fieldName, StringType, nullable = true))
res17: Array[org.apache.spark.sql.types.StructField] = Array(StructField(name,StringType,true), StructField(age,StringType,true))

scala> val schema = StructType(fields)
```



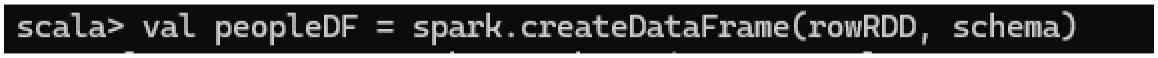
```
scala> val rowRDD = peopleRDD
rowRDD: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[44] at textFile at <console>:30
```



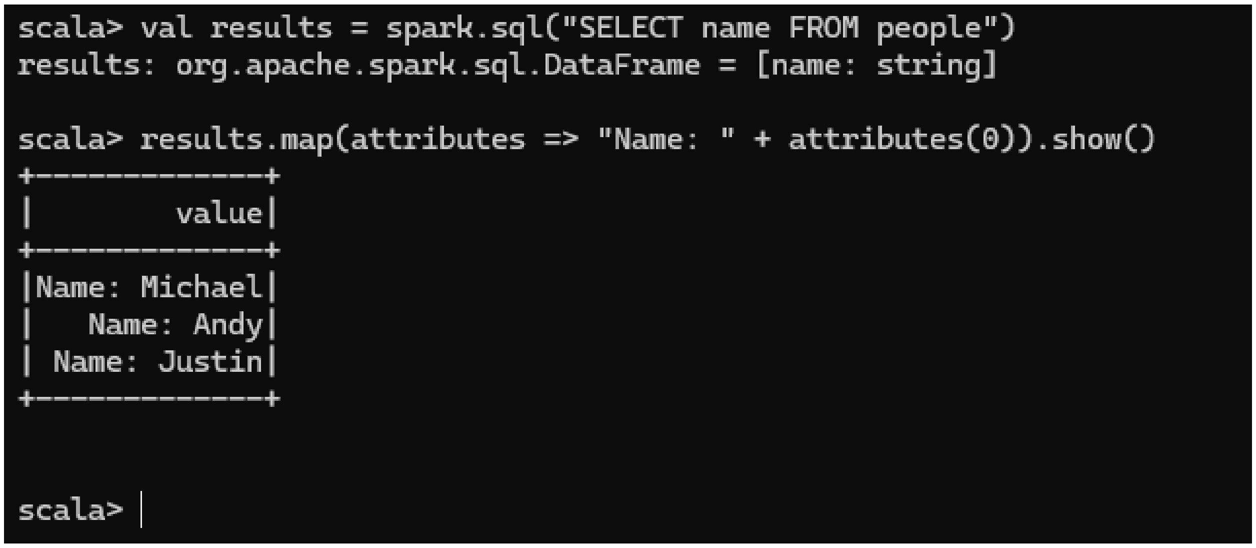
```
scala> .map(_.split(","))
res18: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[45] at map at <console>:32

scala> .map(attributes => Row(attributes(0), attributes(1).trim))
res19: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[46] at map at <console>:32

scala> val peopleDF = spark.createDataFrame(rowRDD, schema)
```



```
scala> val schema = StructType(fields)
```

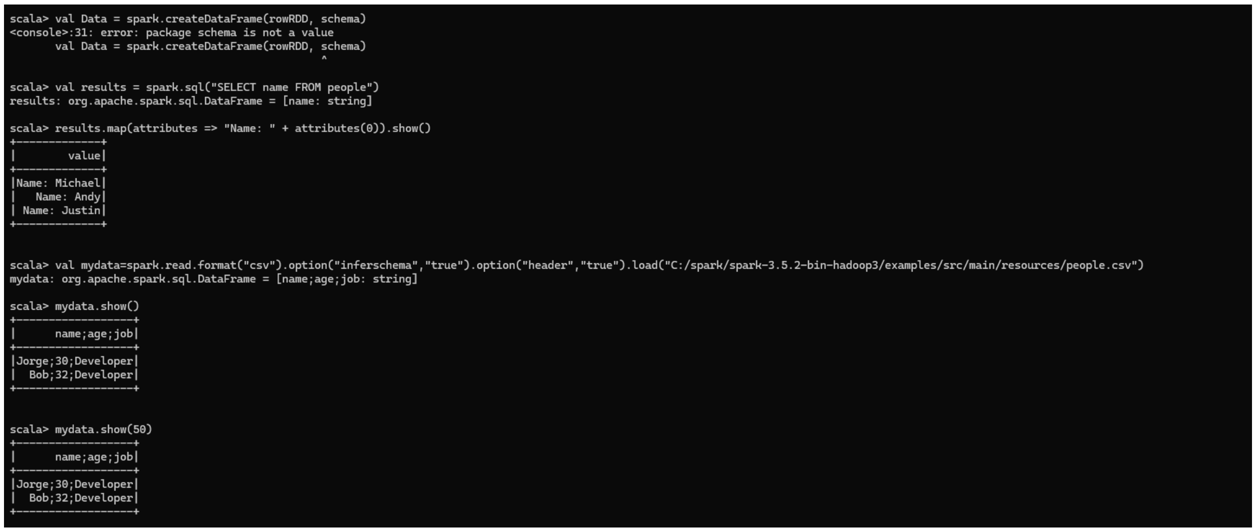


```
scala> val peopleDF = spark.createDataFrame(rowRDD, schema)
```

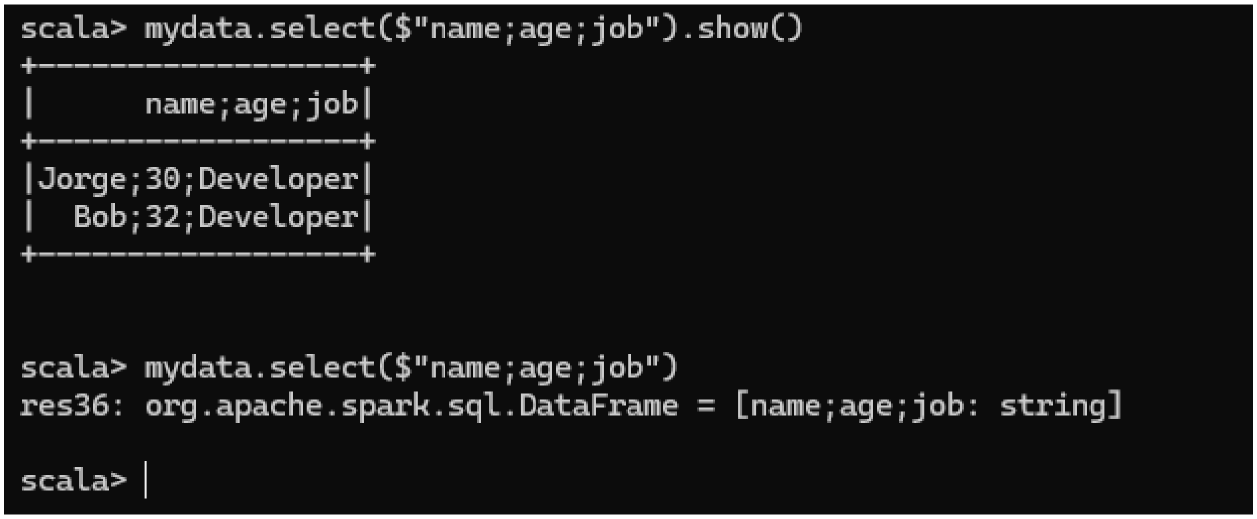


```
scala> val results = spark.sql("SELECT name FROM people")
results: org.apache.spark.sql.DataFrame = [name: string]

scala> results.map(attributes => "Name: " + attributes(0)).show()
+-------------+
|        value|
+-------------+
|Name: Michael|
|   Name: Andy|
| Name: Justin|
+-------------+


scala>
```

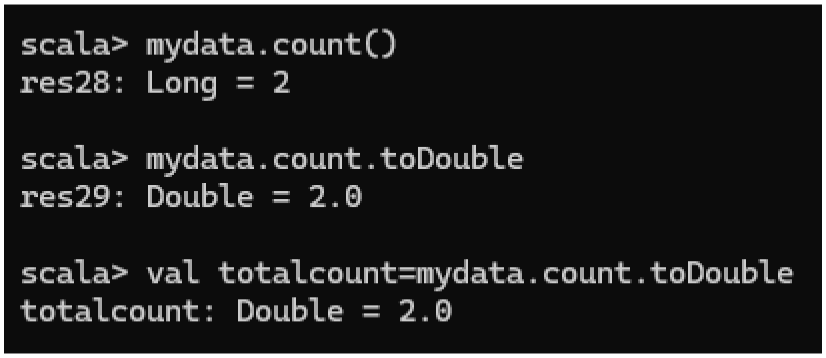
F-



```
scala> val Data = spark.createDataFrame(rowRDD, schema)
<console>:31: error: package schema is not a value
       val Data = spark.createDataFrame(rowRDD, schema)
                                                ^

scala> val results = spark.sql("SELECT name FROM people")
results: org.apache.spark.sql.DataFrame = [name: string]

scala> results.map(attributes => "Name: " + attributes(0)).show()
+-------------+
|        value|
+-------------+
|Name: Michael|
|   Name: Andy|
| Name: Justin|
+-------------+


scala> val mydata=spark.read.format("csv").option("inferschema","true").option("header","true").load("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.csv")
mydata: org.apache.spark.sql.DataFrame = [name;age;job: string]

scala> mydata.show()
+------------------+
|      name;age;job|
+------------------+
|Jorge;30;Developer|
|  Bob;32;Developer|
+------------------+


scala> mydata.show(50)
+------------------+
|      name;age;job|
+------------------+
|Jorge;30;Developer|
|  Bob;32;Developer|
+------------------+
```

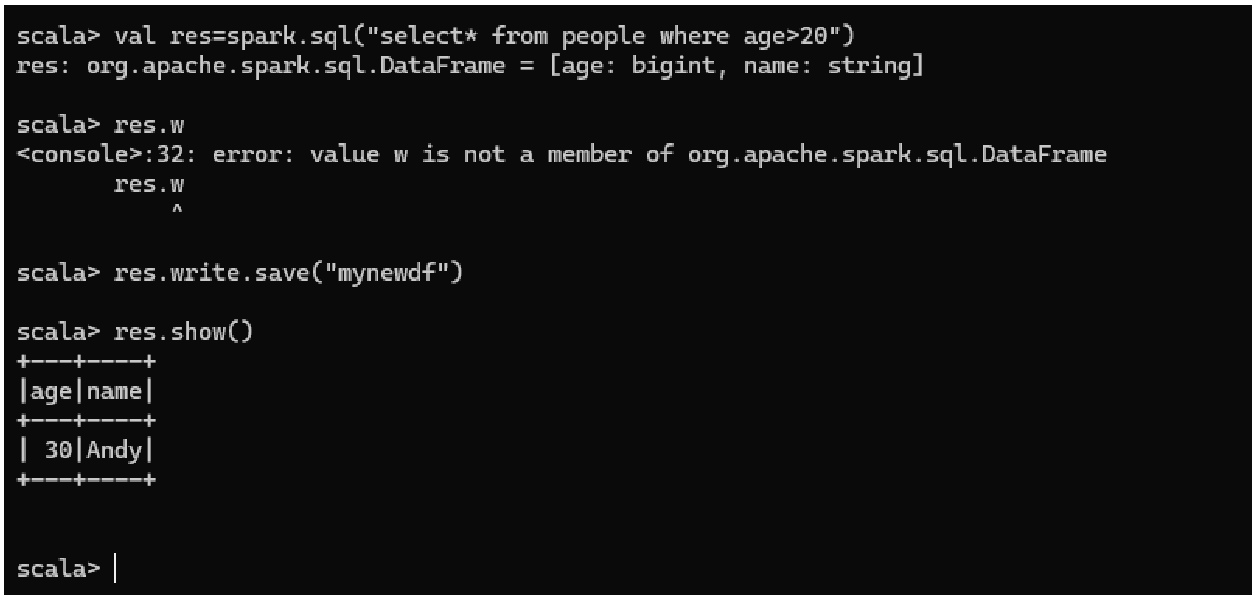


```
scala> mydata.select($"name;age;job").show()
+------------------+
|      name;age;job|
+------------------+
|Jorge;30;Developer|
|  Bob;32;Developer|
+------------------+


scala> mydata.select($"name;age;job")
res36: org.apache.spark.sql.DataFrame = [name;age;job: string]

scala>
```



```
scala> mydata.count()
res28: Long = 2

scala> mydata.count.toDouble
res29: Double = 2.0

scala> val totalcount=mydata.count.toDouble
totalcount: Double = 2.0
```

G-



```
scala> val res=spark.sql("select* from people where age>20")
res: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

scala> res.w
<console>:32: error: value w is not a member of org.apache.spark.sql.DataFrame
       res.w
           ^

scala> res.write.save("mynewdf")

scala> res.show()
+---+----+
|age|name|
+---+----+
| 30|Andy|
+---+----+


scala>
```