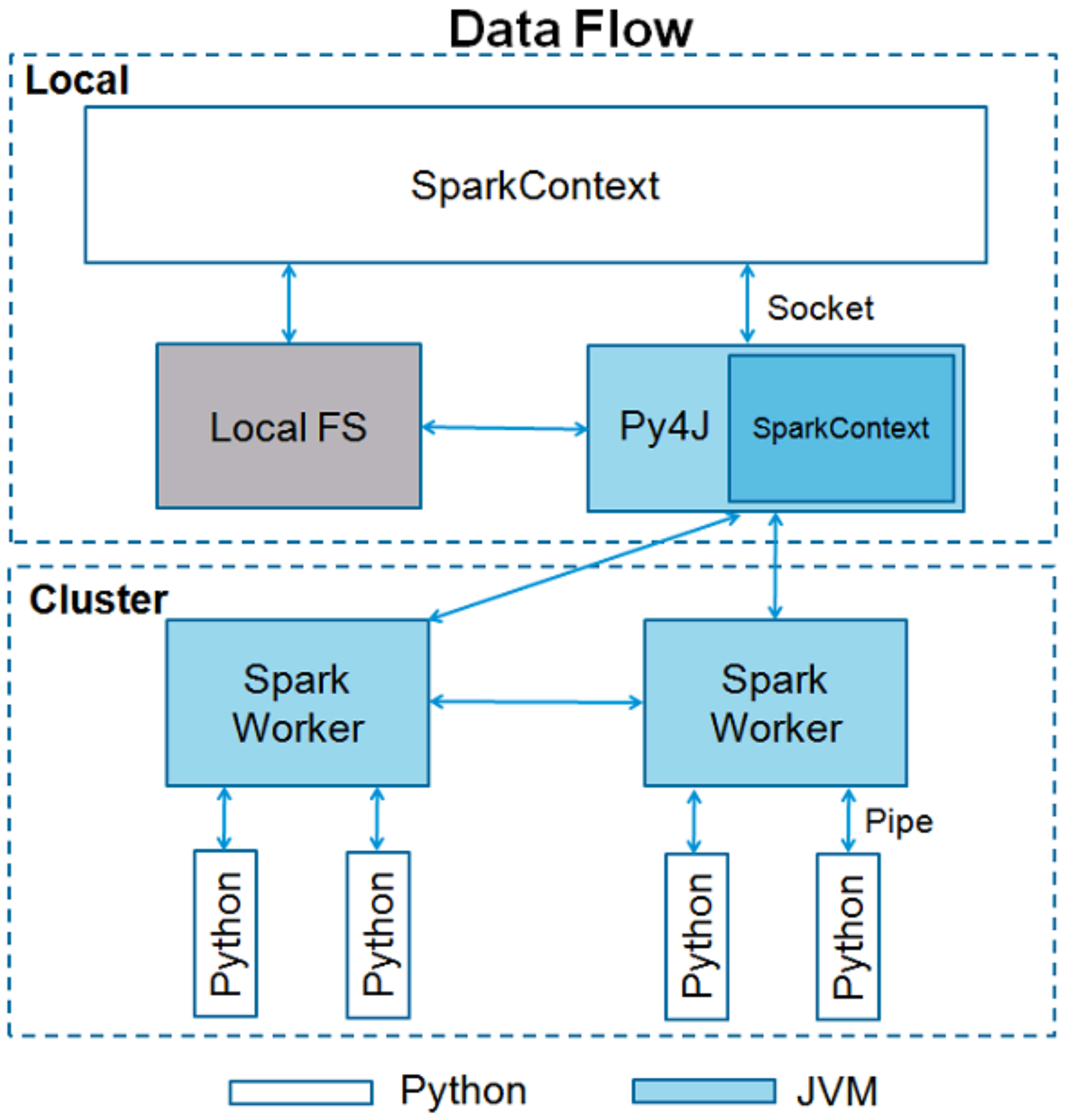
What is Spark, anyway?

Big Data & Predictive Analysis Lanjut

Arif Laksito, M. Kom

Spark?

- Spark is a platform for cluster computing.
- Spark lets you spread data and computations over *clusters* with multiple *nodes* (think of each node as a separate computer).
- Splitting up your data makes it easier to work with very large datasets because each node only works with a small amount of data.
- Deciding whether or not Spark is the best solution for your problem takes some experience, but you can consider questions like:
  1. Is my data too big to work with on a single machine?
  2. Can my calculations be easily parallelized?

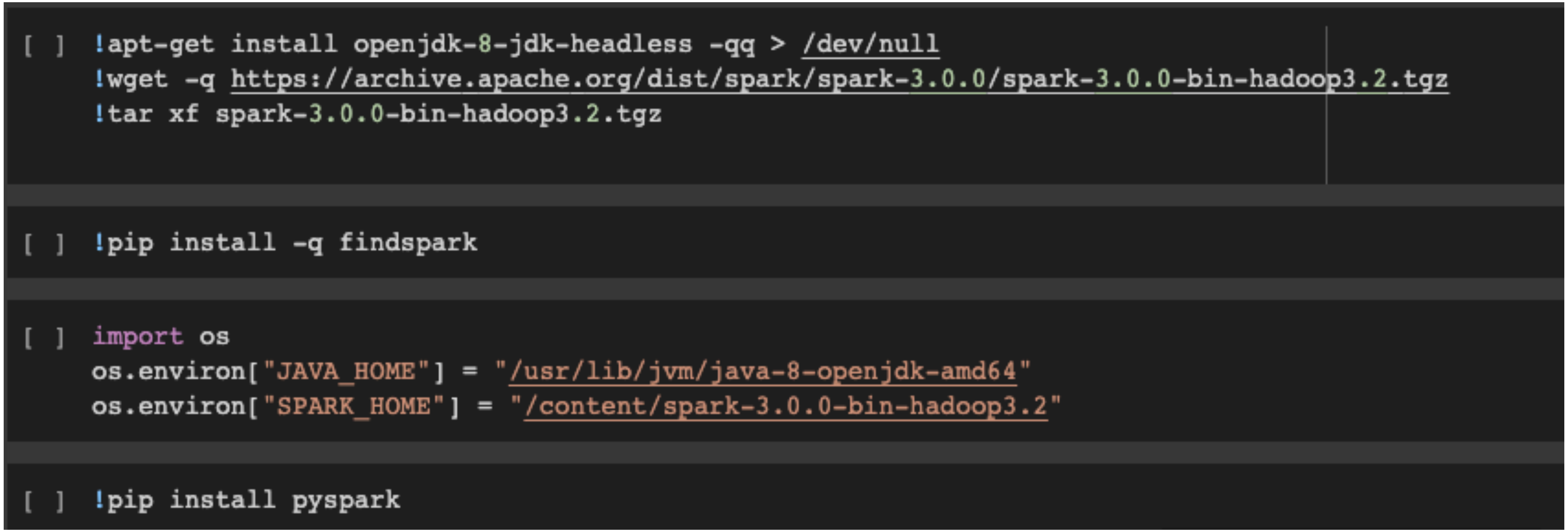


Using Spark in Python

- Apache Spark is written in Scala programming language.
- To support Python with Spark, Apache Spark Community released a tool, PySpark.
- Using PySpark, you can work with RDDs in Python programming language also. It is because of a library called Py4j that they are able to achieve this.

Environment Setup

- In this practice, we will use Google Colab and PySpark.
- Before starting PySpark, you need to set following command in Colab.



```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz
!tar xf spark-3.0.0-bin-hadoop3.2.tgz
```

```
!pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"
```

```
!pip install pyspark
```

SparkSession

- SparkSession was introduced in version 2.0, It is an entry point to underlying PySpark functionality in order to programmatically create PySpark RDD, DataFrame.
- It's object Spark is default available in `pyspark-shell` and it can be created programmatically using SparkSession.
- SparkSession also includes all the APIs available in different contexts.
  1. SparkContext
  2. SQLContext
  3. StreamingContext
  4. HiveContext

SparkContext

- SparkContext is the entry point to any spark functionality.
- When we run any Spark application, a driver program starts, which has the main function and your SparkContext gets initiated here.
- The driver program then runs the operations inside the executors on worker nodes.

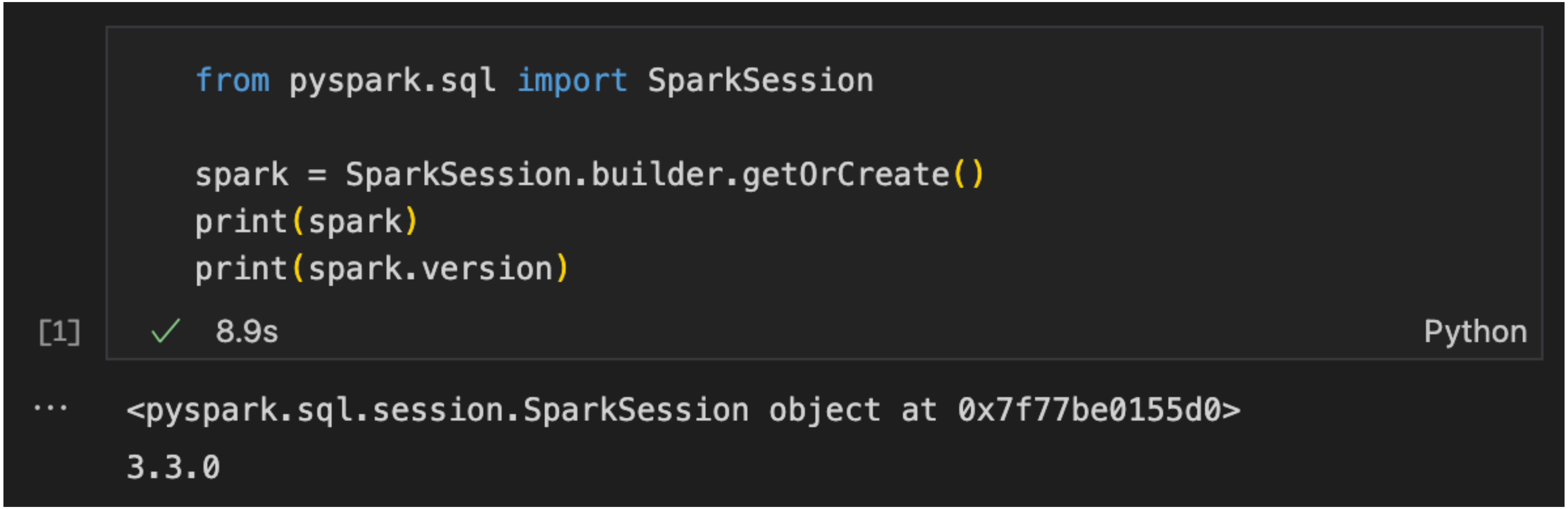
The following code block has the details of a PySpark class and the parameters, which a SparkContext can take.

```
class pyspark.SparkContext (
   master = None,
   appName = None,
   sparkHome = None,
   pyFiles = None,
   environment = None,
   batchSize = 0,
   serializer = PickleSerializer(),
   conf = None,
   gateway = None,
   jsc = None,
   profiler_cls = <class
'pyspark.profiler.BasicProfiler'>
)
```

Let's code

Beginning of PySpark

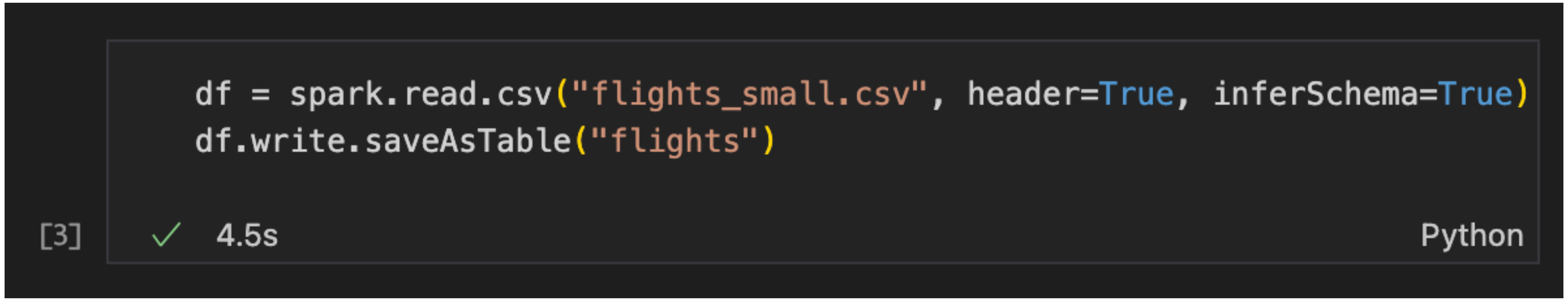
Creating SparkSession



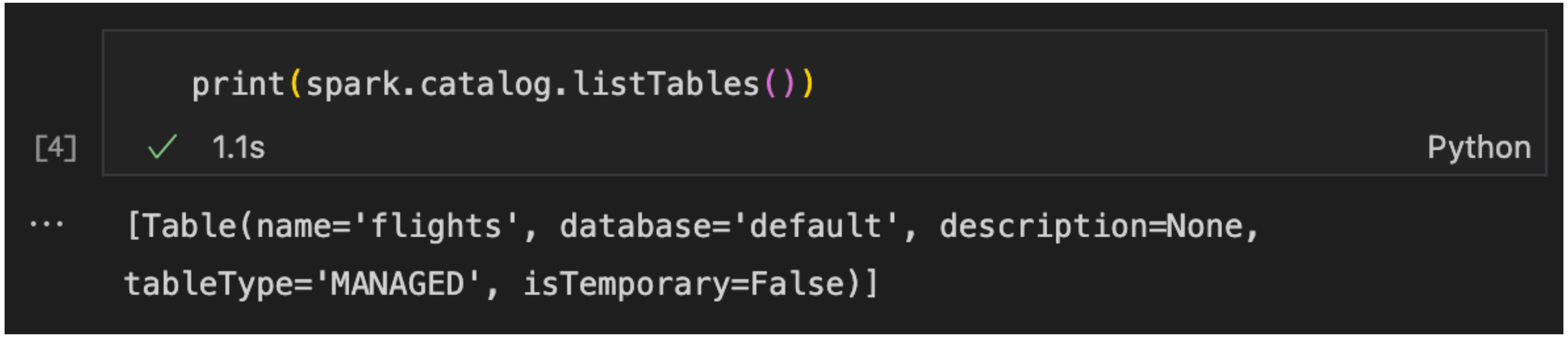
```python
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()
print(spark)
print(spark.version)
```

```
<pyspark.sql.session.SparkSession object at 0x7f77be0155d0>
3.3.0
```

Creating & Viewing Table



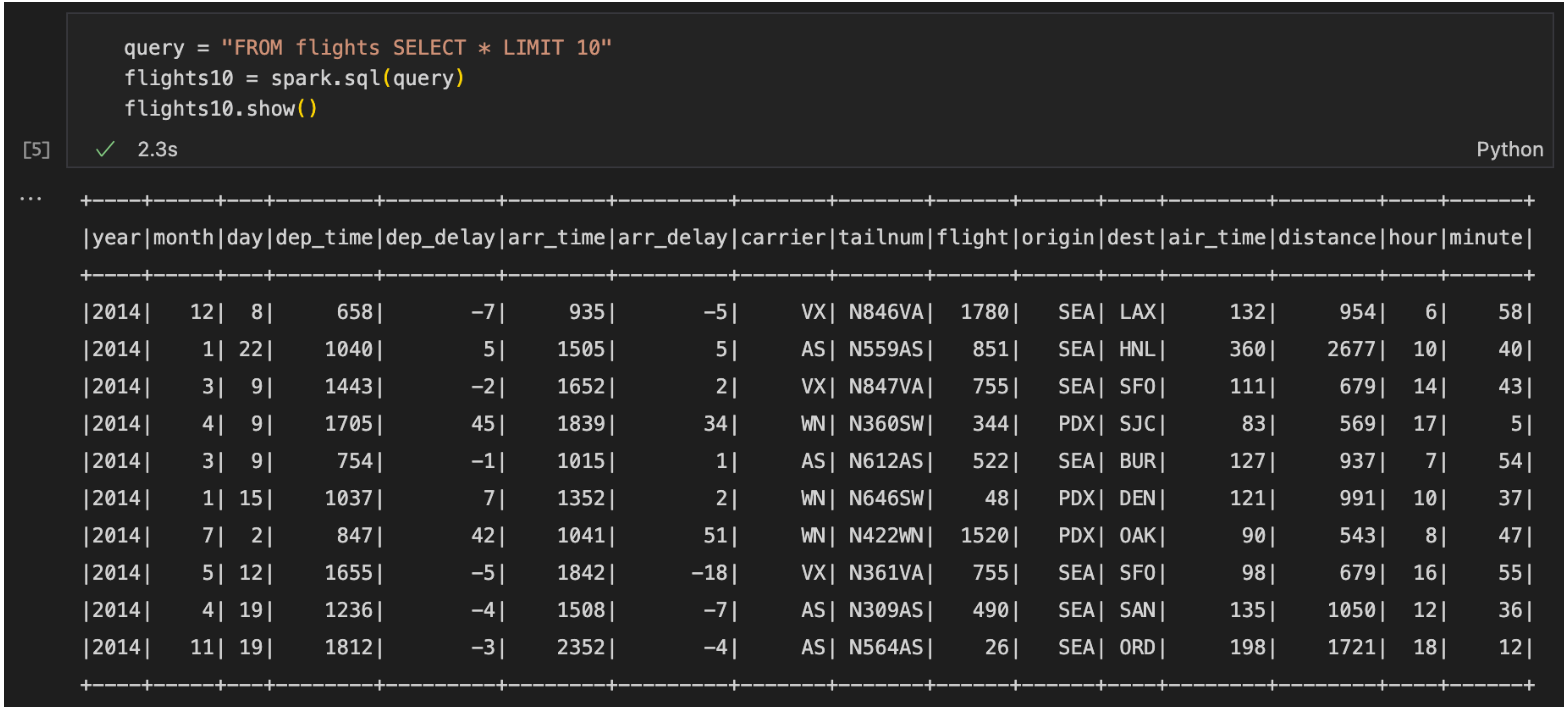
```
df = spark.read.csv("flights_small.csv", header=True, inferSchema=True)
df.write.saveAsTable("flights")
```



```
print(spark.catalog.listTables())
```

```
[Table(name='flights', database='default', description=None,
tableType='MANAGED', isTemporary=False)]
```

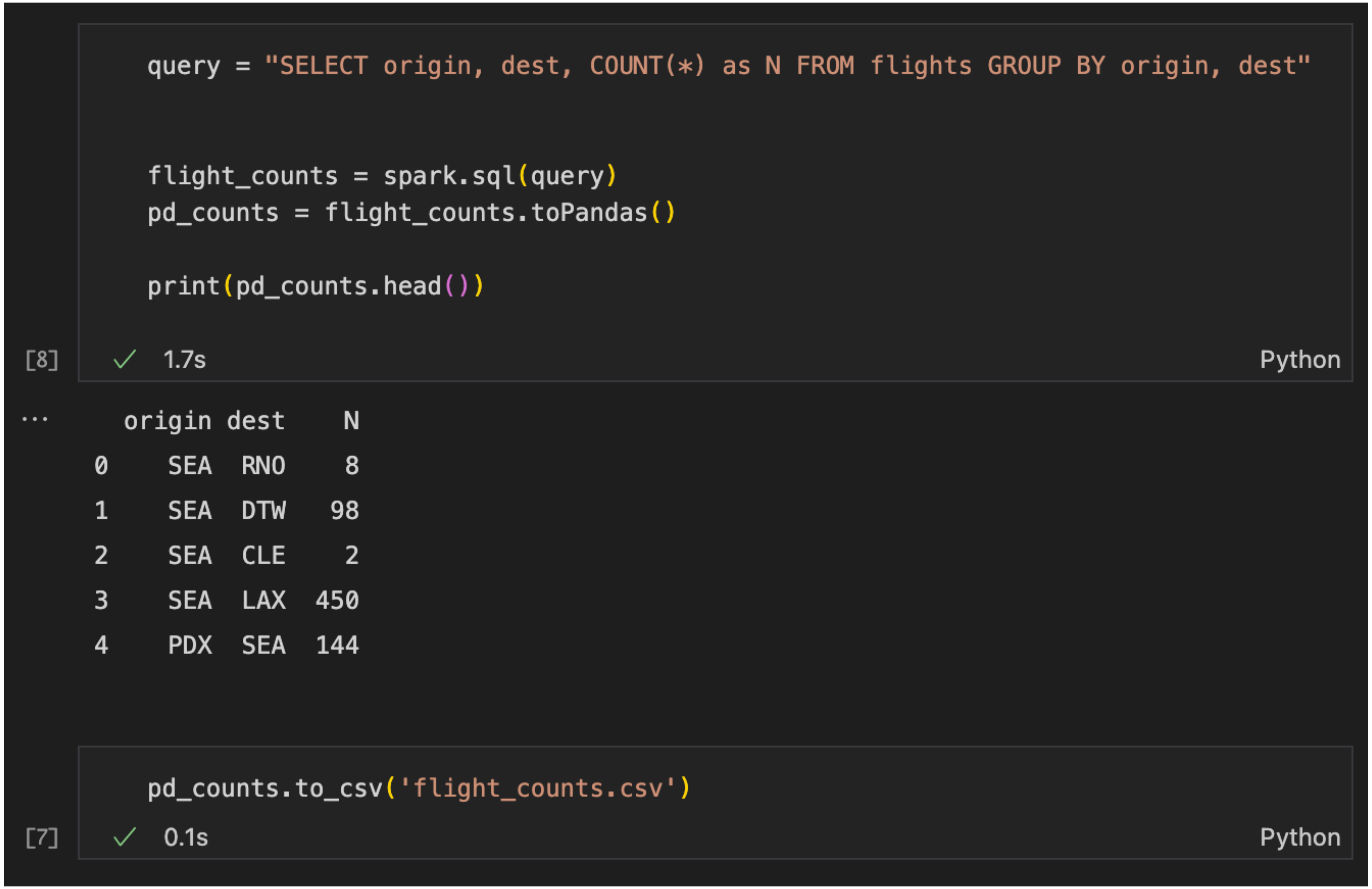
Doing query



```python
query = "FROM flights SELECT * LIMIT 10"
flights10 = spark.sql(query)
flights10.show()
```

```
+----+-----+---+--------+---------+--------+---------+-------+-------+------+------+----+--------+--------+----+------+
|year|month|day|dep_time|dep_delay|arr_time|arr_delay|carrier|tailnum|flight|origin|dest|air_time|distance|hour|minute|
+----+-----+---+--------+---------+--------+---------+-------+-------+------+------+----+--------+--------+----+------+
|2014|   12|  8|     658|       -7|     935|       -5|     VX| N846VA|  1780|   SEA| LAX|     132|     954|   6|    58|
|2014|    1| 22|    1040|        5|    1505|        5|     AS| N559AS|   851|   SEA| HNL|     360|    2677|  10|    40|
|2014|    3|  9|    1443|       -2|    1652|        2|     VX| N847VA|   755|   SEA| SFO|     111|     679|  14|    43|
|2014|    4|  9|    1705|       45|    1839|       34|     WN| N360SW|   344|   PDX| SJC|      83|     569|  17|     5|
|2014|    3|  9|     754|       -1|    1015|        1|     AS| N612AS|   522|   SEA| BUR|     127|     937|   7|    54|
|2014|    1| 15|    1037|        7|    1352|        2|     WN| N646SW|    48|   PDX| DEN|     121|     991|  10|    37|
|2014|    7|  2|     847|       42|    1041|       51|     WN| N422WN|  1520|   PDX| OAK|      90|     543|   8|    47|
|2014|    5| 12|    1655|       -5|    1842|      -18|     VX| N361VA|   755|   SEA| SFO|      98|     679|  16|    55|
|2014|    4| 19|    1236|       -4|    1508|       -7|     AS| N309AS|   490|   SEA| SAN|     135|    1050|  12|    36|
|2014|   11| 19|    1812|       -3|    2352|       -4|     AS| N564AS|    26|   SEA| ORD|     198|    1721|  18|    12|
+----+-----+---+--------+---------+--------+---------+-------+-------+------+------+----+--------+--------+----+------+
```

Pandafy a Spark Dataframe



```python
query = "SELECT origin, dest, COUNT(*) as N FROM flights GROUP BY origin, dest"


flight_counts = spark.sql(query)
pd_counts = flight_counts.toPandas()

print(pd_counts.head())
```

```
  origin dest    N
0    SEA  RNO    8
1    SEA  DTW   98
2    SEA  CLE    2
3    SEA  LAX  450
4    PDX  SEA  144
```

```python
pd_counts.to_csv('flight_counts.csv')
```

Refferences

- https://campus.datacamp.com/courses/introduction-to-pyspark
- https://www.tutorialspoint.com/pyspark/pyspark_sparkcontext.htm
- https://sparkbyexamples.com/pyspark/pyspark-what-is-sparksession/