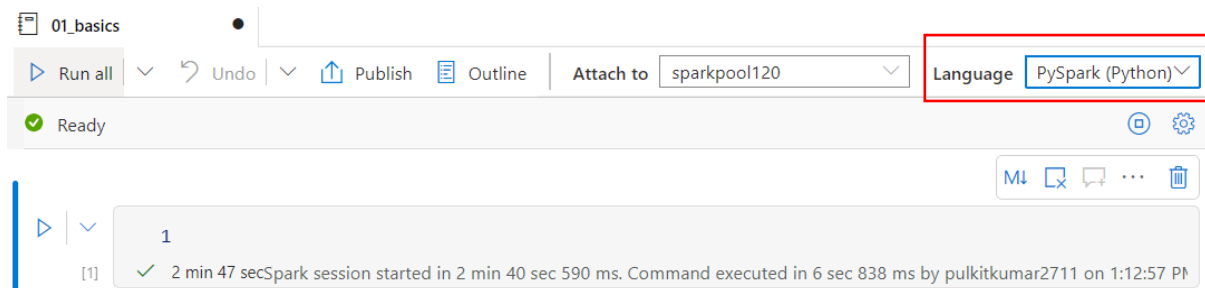


😊 Spark Pool with Python

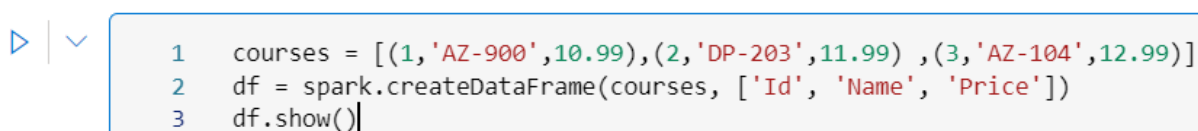
In this lab, we're focusing on utilizing our Spark pool with Python instead of Scala. We start by switching the language to Python in our notebook and then proceed to write and execute Python code for our data analysis tasks. After running the code, we observe the output displayed in the form of a table, showing the results of our analysis. Specifically, we demonstrate how to convert course information into a DataFrame and then sort the data to obtain it in a sorted manner for better analysis and presentation.

1. In this lab we are going to work on our spark pool but we use Python as our language instead of Scala.
2. Now delete your previous code and choose your language as Python.



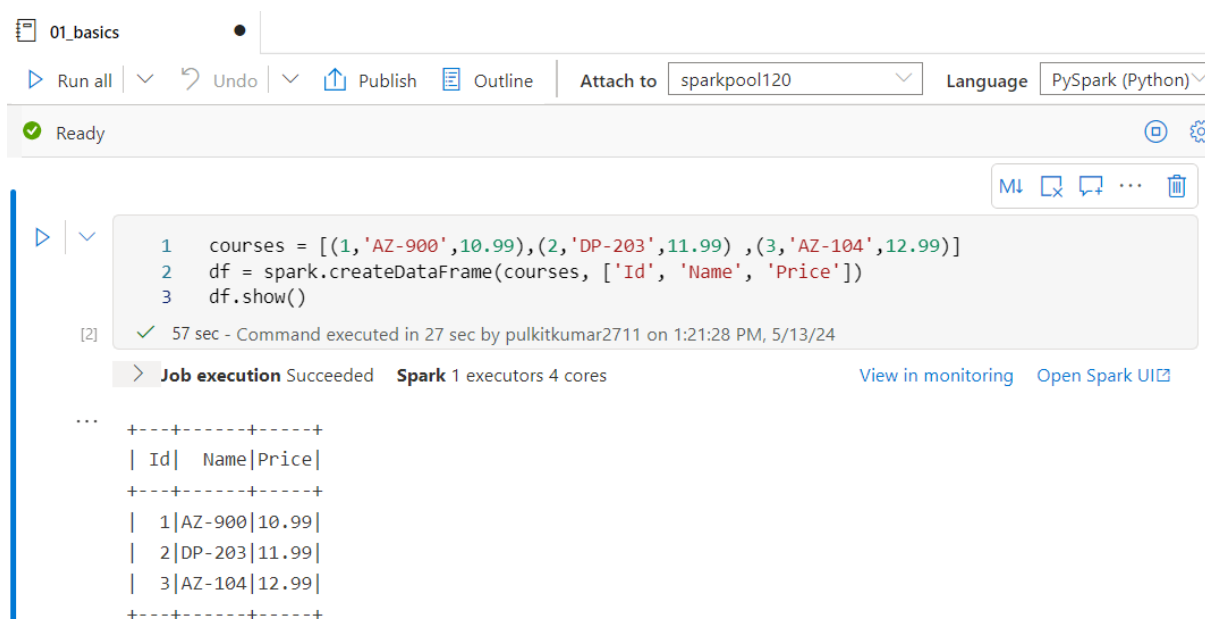
The screenshot shows the top toolbar of a Databricks notebook. The 'Language' dropdown menu is highlighted with a red box and set to 'PySpark (Python)'. Below the toolbar, a code cell is shown with a single line of code: `1`. The output of the cell is a green checkmark and the text: '2 min 47 sec Spark session started in 2 min 40 sec 590 ms. Command executed in 6 sec 838 ms by pulkitkumar2711 on 1:12:57 PM'.

3. After that you need to write your code and run it.



```
1 courses = [(1, 'AZ-900', 10.99), (2, 'DP-203', 11.99), (3, 'AZ-104', 12.99)]
2 df = spark.createDataFrame(courses, ['Id', 'Name', 'Price'])
3 df.show()
```

4. Below you can see that you got the output of your code in form of a table.



The screenshot shows the same Databricks notebook interface. The code cell is now executed, and the output is displayed as a table. The table has three columns: 'Id', 'Name', and 'Price'. The data is as follows:

Id	Name	Price
1	AZ-900	10.99
2	DP-203	11.99
3	AZ-104	12.99

Below the table, the status 'Job execution Succeeded' is shown, along with 'Spark 1 executors 4 cores'. There are also links to 'View in monitoring' and 'Open Spark UI'.

5. Our course, list of data or our course information in Python has now been converted onto a data frame.
6. Now you are going to add another cell to write more code but this time to sort your data. Below you can see that we ran our code and we got the data in sorted manner.

▶

▼

```
1 from pyspark.sql.functions import col
2
3 sortedddf=df.sort(col("Price").desc())
4 sortedddf.show()
```

[3] ✓ 1 sec - Command executed in 1 sec 852 ms by pulkitkumar2711 on 1:26:30 PM, 5/13/24

> Job execution Succeeded Spark 1 executors 4 cores [View in monitoring](#) [Open Spark UI](#)

...

	Id	Name	Price
	3	AZ-104	12.99
	2	DP-203	11.99
	1	AZ-900	10.99