What is PySpark :-

PySpark is an interface for Apache Spark in Python. It is a library.

Setting Up :-

- Create a virtual environment.
- Install pyspark in it :-

```
pip install pyspark
```

It is also install py4j automatically.

SparkSession :-

When we want to work with PySpark, we always need to start a Spark Session.

To create a Spark Session, we need to code as follows :-

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Prac...
...tise").getOrCreate()
```

Name of the Spark session that we create

This function actually creates the session, and returns the session object.

We save the session in a variable (spark here)

We can run this code multiple times, and it creates a session (takes a bit of time when 1st time executed).

The Spark session gets created in memory/RAM.

When running in local, there will be 1 cluster.

On cloud, there will be Master & multiple clusters.

Reading data with PySpark :-

```
df_spark = spark.read.csv("filename.extension")
```

- `df_spark` : storing the data that is being read
- `spark` : Spark session object
- `read.csv("filename.extension")` : file name/path that is to be read.
  - here, we have many options like csv, json, .. Depending on the file to be read, that function should be used appropriately.

This method will take all the data as raw data, and will store them as column data (even the column headings will be considered data), and will by itself give arbitary naming to column as headings.

E:- _c0, _c1.

To read by taking the 1st row as row header names, we need to write as follows:-

```
df_spark = spark.read.option('header', 'true').csv("filename")
```

Due to this `option()` method, we can set the 1st row as header row.

- 1st parameter : Key of option, (which is 'header' here)
- 2nd param: Value of the key.

Displaying the data being read :-

```
df_spark.show()
```

- show() method displays the data stored in the Pyspark dataframe object, in tabular format.

- `type(df_spark)` ⇒ pyspark.sql.dataframe.DataFrame
- This shows what type of object, in which the read data is being stored.
- Thus, we see, it is a **Pyspark SQL Dataframe**

---

head() method :-

Takes one integer parameter (n) :-

```
df_spark.head(n)
```

- n > 1 :- Returns a list of first (n) rows.
- n = 1 :- Returns a single row object.

---

printSchema() method :-

- This method actually prints the schema of the

tabular format in which data is stored.

* Schema here means that the column names, and the data types of each column.

```
df_spark.printSchema()
```

* This will show result similar to desc table_name in MySQL