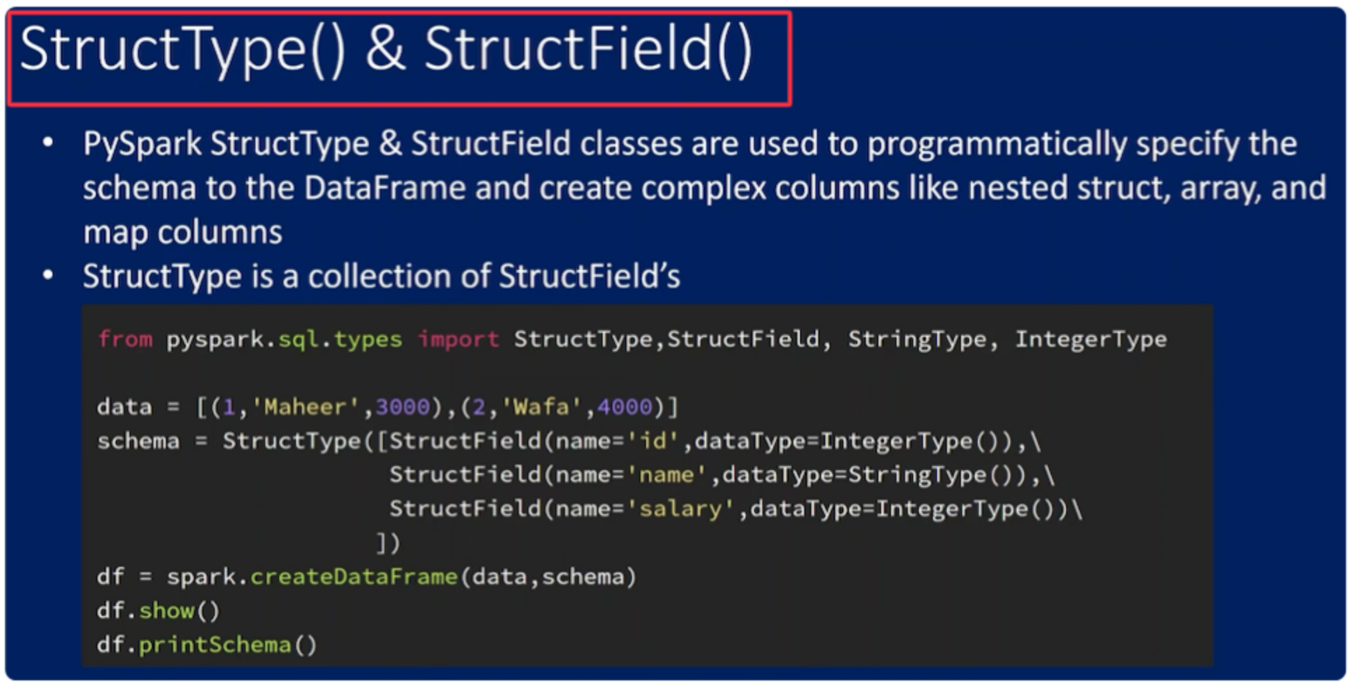
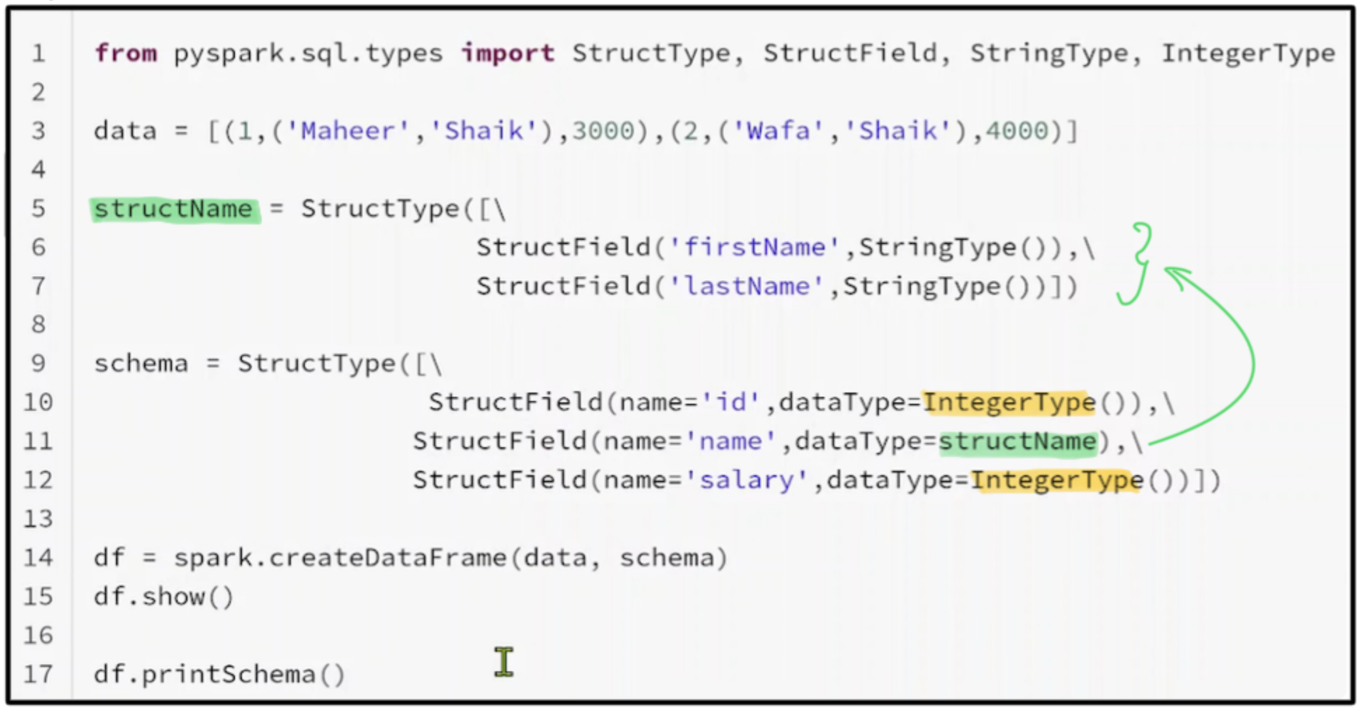
Struct type & Struct field, Array type, Array type Column, Map



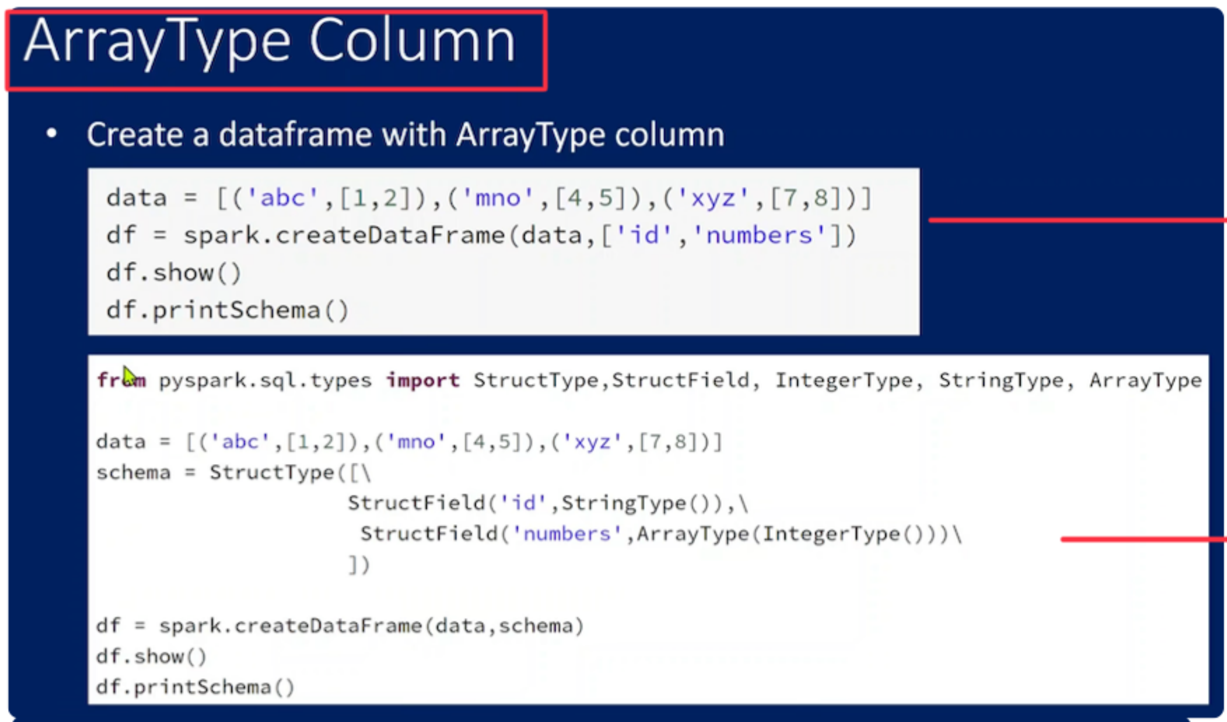
StructType() & StructField()

- PySpark StructType & StructField classes are used to programmatically specify the schema to the DataFrame and create complex columns like nested struct, array, and map columns
- StructType is a collection of StructField's

```
from pyspark.sql.types import StructType,StructField, StringType, IntegerType

data = [(1,'Maheer',3000),(2,'Wafa',4000)]
schema = StructType([StructField(name='id',dataType=IntegerType()),\
                     StructField(name='name',dataType=StringType()),\
                     StructField(name='salary',dataType=IntegerType())\
                    ])
df = spark.createDataFrame(data,schema)
df.show()
df.printSchema()
```



```
from pyspark.sql.types import StructType, StructField, StringType, IntegerType

data = [(1,('Maheer','Shaik'),3000),(2,('Wafa','Shaik'),4000)]

structName = StructType([\
                        StructField('firstName',StringType()),\
                        StructField('lastName',StringType())])

schema = StructType([\
                     StructField(name='id',dataType=IntegerType()),\
                    StructField(name='name',dataType=structName),\
                    StructField(name='salary',dataType=IntegerType())])

df = spark.createDataFrame(data, schema)
df.show()

df.printSchema()
```



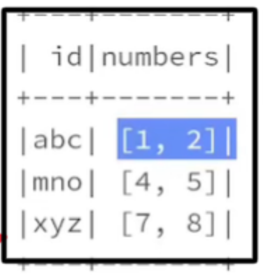
ArrayType Column

- Create a dataframe with ArrayType column

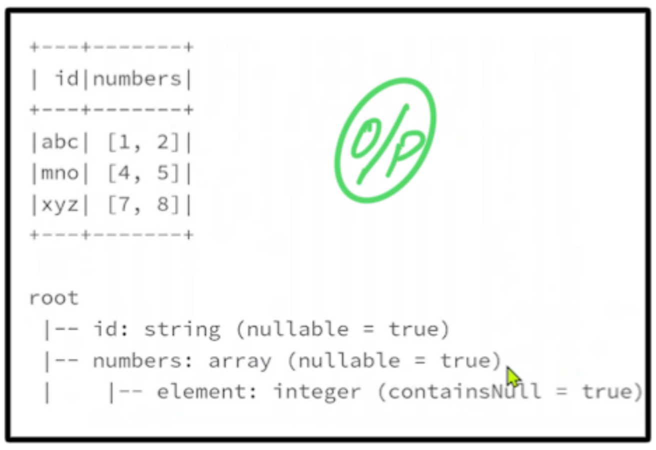
```
data = [('abc',[1,2]),('mno',[4,5]),('xyz',[7,8])]
df = spark.createDataFrame(data,['id','numbers'])
df.show()
df.printSchema()
```

```
from pyspark.sql.types import StructType,StructField, IntegerType, StringType, ArrayType

data = [('abc',[1,2]),('mno',[4,5]),('xyz',[7,8])]
schema = StructType([\
                     StructField('id',StringType()),\
                      StructField('numbers',ArrayType(IntegerType()))\
                     ])

df = spark.createDataFrame(data,schema)
df.show()
df.printSchema()
```

o/p

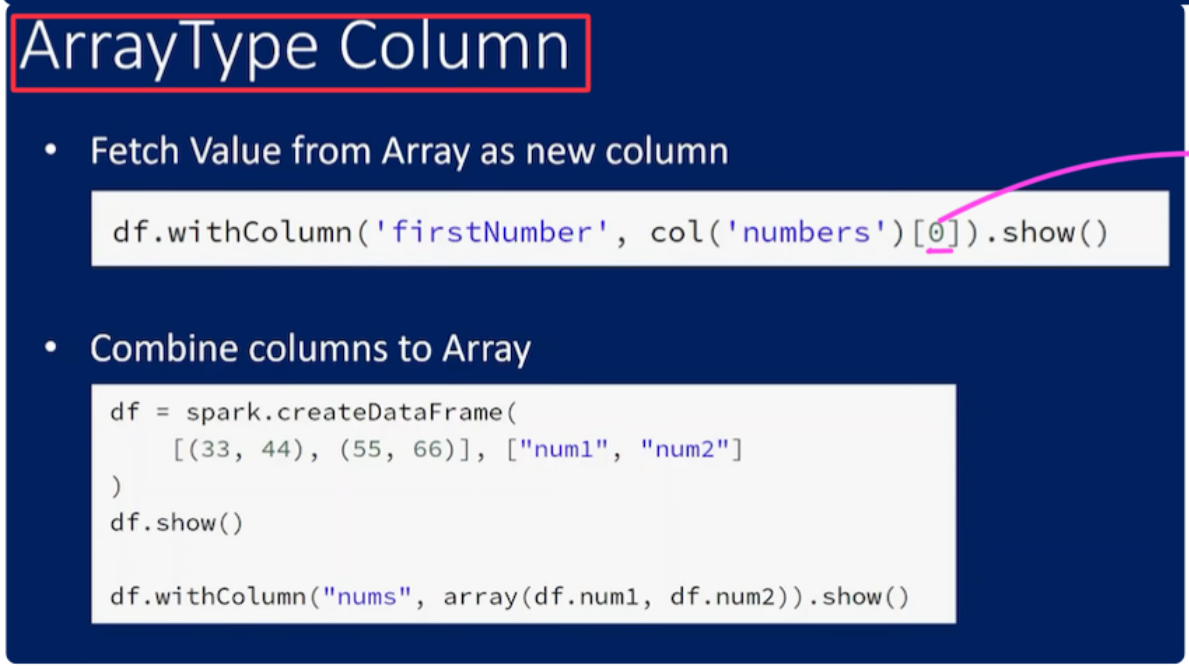


```
+---+--------+
| id|numbers|
+---+--------+
|abc| [1, 2]|
|mno| [4, 5]|
|xyz| [7, 8]|
+---+--------+
```



o/p

```
+---+--------+
| id|numbers|
+---+--------+
|abc| [1, 2]|
|mno| [4, 5]|
|xyz| [7, 8]|
+---+--------+

root
 |-- id: string (nullable = true)
 |-- numbers: array (nullable = true)
 |    |-- element: integer (containsNull = true)
```



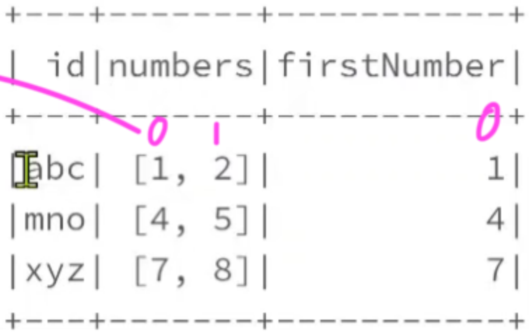
ArrayType Column

- Fetch Value from Array as new column

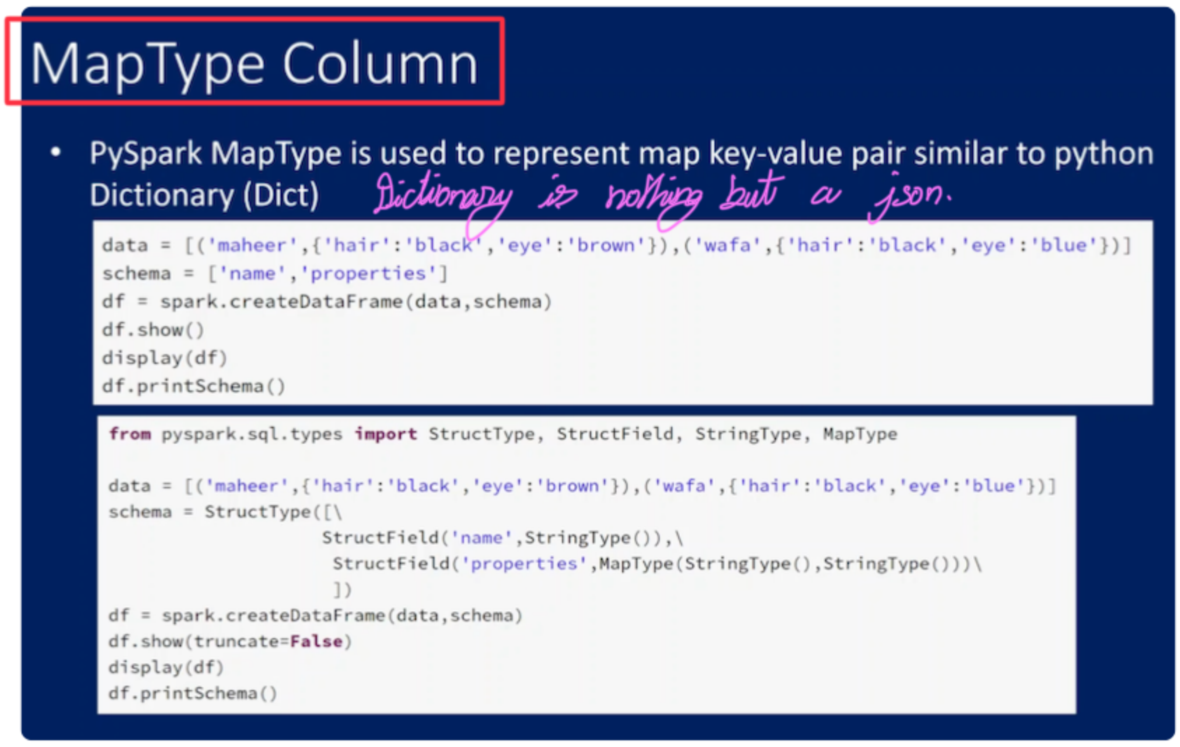
```
df.withColumn('firstNumber', col('numbers')[0]).show()
```

- Combine columns to Array

```
df = spark.createDataFrame(
    [(33, 44), (55, 66)], ["num1", "num2"]
)
df.show()

df.withColumn("nums", array(df.num1, df.num2)).show()
```



```
+---+--------+-----------+
| id|numbers|firstNumber|
+---+--------+-----------+
|abc| [1, 2]|          1|
|mno| [4, 5]|          4|
|xyz| [7, 8]|          7|
+---+--------+-----------+
```

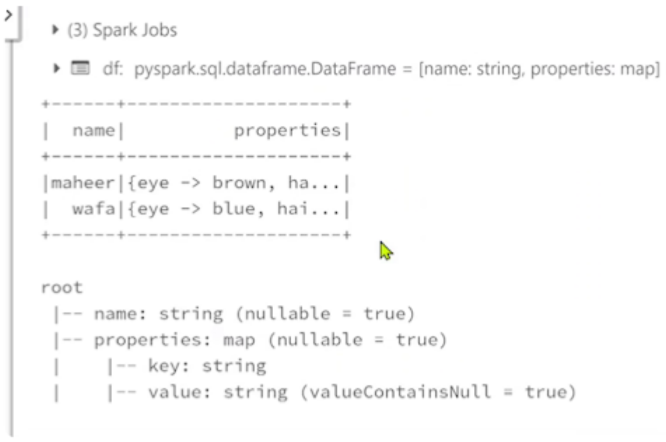


MapType Column

- PySpark MapType is used to represent map key-value pair similar to python Dictionary (Dict) Dictionary is nothing but a json.

```
data = [('maheer',{'hair':'black','eye':'brown'}),('wafa',{'hair':'black','eye':'blue'})]
schema = ['name','properties']
df = spark.createDataFrame(data,schema)
df.show()
display(df)
df.printSchema()
```

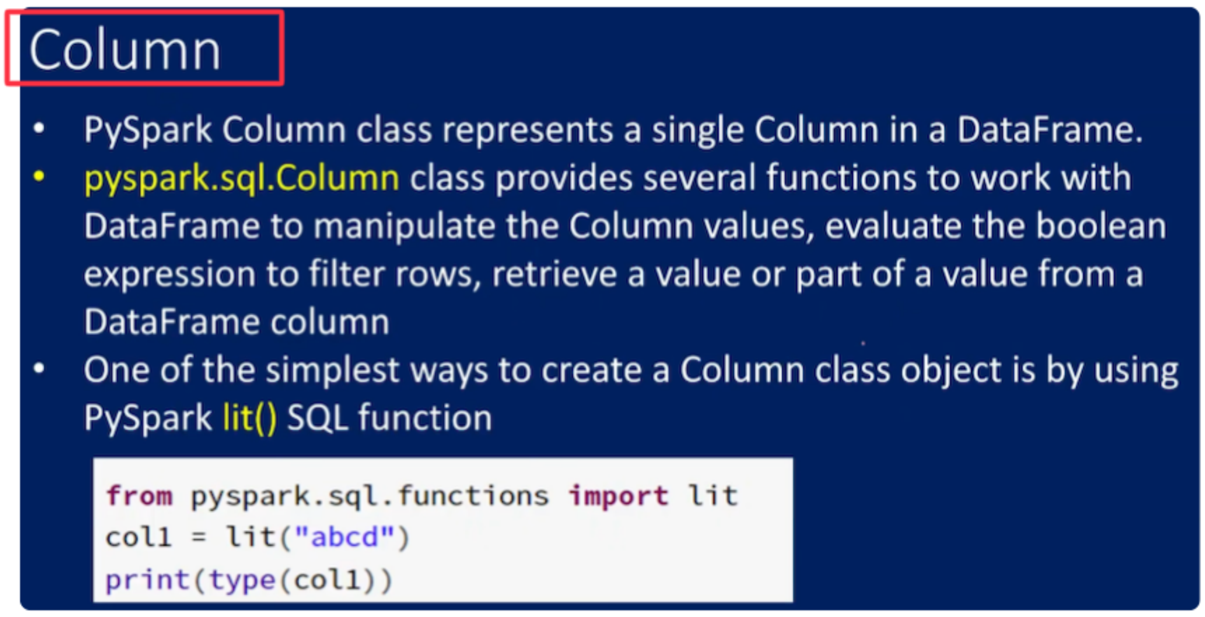
```
from pyspark.sql.types import StructType, StructField, StringType, MapType

data = [('maheer',{'hair':'black','eye':'brown'}),('wafa',{'hair':'black','eye':'blue'})]
schema = StructType([\
                     StructField('name',StringType()),\
                      StructField('properties',MapType(StringType(),StringType()))\
                      ])
df = spark.createDataFrame(data,schema)
df.show(truncate=False)
display(df)
df.printSchema()
```



▸ (3) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [name: string, properties: map]

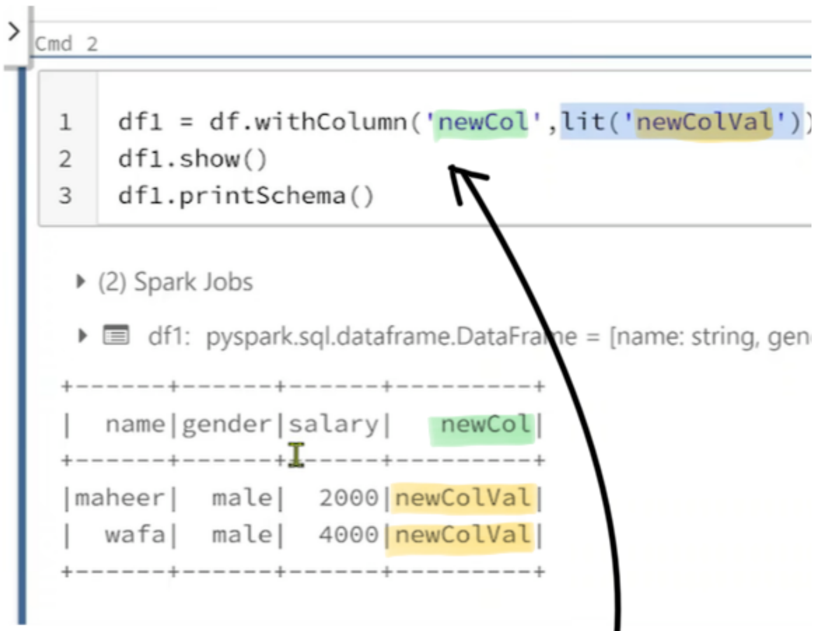
```
+------+--------------------+
|  name|          properties|
+------+--------------------+
|maheer|{eye -> brown, ha...|
|  wafa|{eye -> blue, hai...|
+------+--------------------+

root
 |-- name: string (nullable = true)
 |-- properties: map (nullable = true)
 |    |-- key: string
 |    |-- value: string (valueContainsNull = true)
```



Column

- PySpark Column class represents a single Column in a DataFrame.
- pyspark.sql.Column class provides several functions to work with DataFrame to manipulate the Column values, evaluate the boolean expression to filter rows, retrieve a value or part of a value from a DataFrame column
- One of the simplest ways to create a Column class object is by using PySpark lit() SQL function

```
from pyspark.sql.functions import lit
col1 = lit("abcd")
print(type(col1))
```



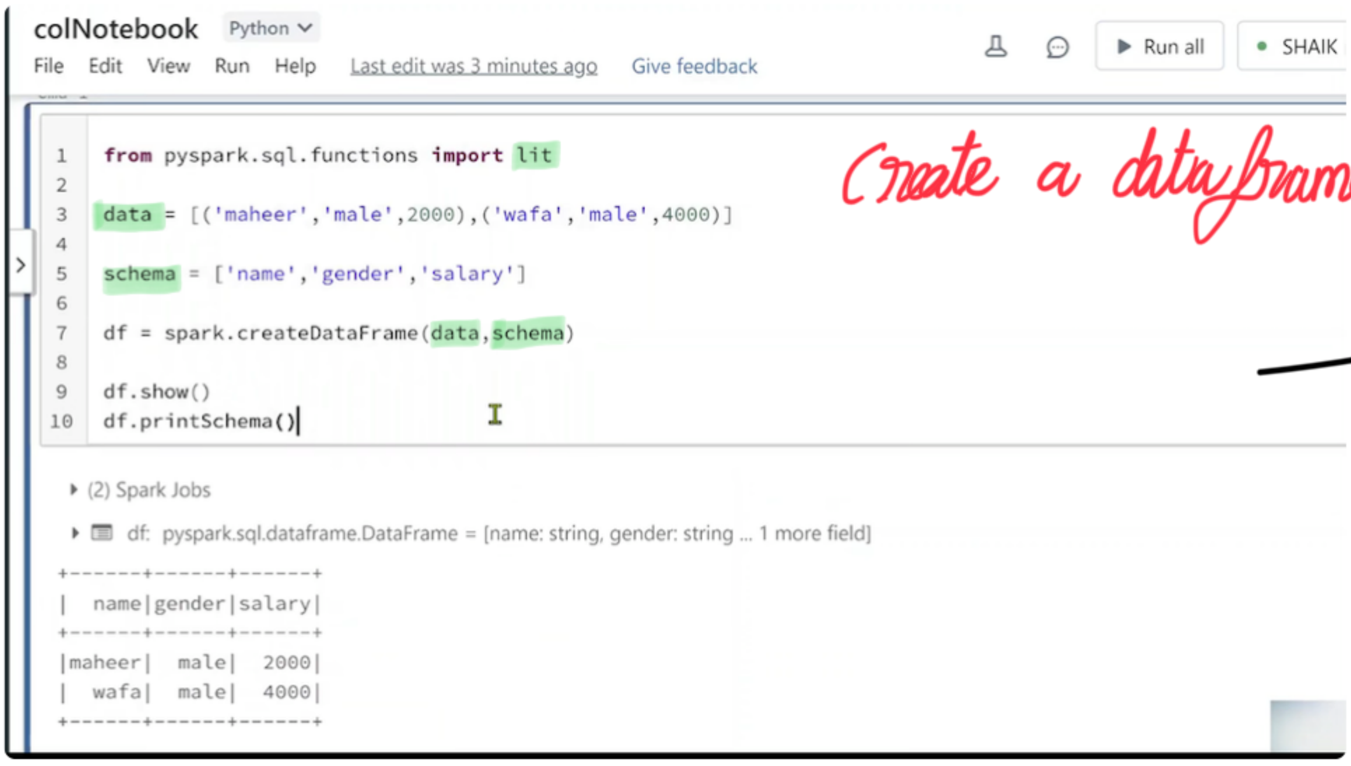
Cmd 2

```
df1 = df.withColumn('newCol',lit('newColVal'))
df1.show()
df1.printSchema()
```

▸ (2) Spark Jobs

▸ df1: pyspark.sql.dataframe.DataFrame = [name: string, gen

```
+------+------+------+---------+
|  name|gender|salary|   newCol|
+------+------+------+---------+
|maheer|  male|  2000|newColVal|
|  wafa|  male|  4000|newColVal|
+------+------+------+---------+
```



```
from pyspark.sql.functions import lit

data = [('maheer','male',2000),('wafa','male',4000)]

schema = ['name','gender','salary']

df = spark.createDataFrame(data,schema)

df.show()
df.printSchema()
```

Create a dataframe

▸ (2) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [name: string, gender: string ... 1 more field]

```
+------+------+------+
|  name|gender|salary|
+------+------+------+
|maheer|  male|  2000|
|  wafa|  male|  4000|
+------+------+------+
```