1. These are similar to functions in SQL. We define some logic in functions and store them in database and use them in queries.
2. Similar to that we can write our own custom logic in python function and register it with Pyspark using udf() function.

Example-1

from pyspark.sql import \*  
from pyspark.sql.functions import \*  
from pyspark.sql.types import \*  
  
*# Create a Spark session*spark = SparkSession.builder.appName("udf").getOrCreate()  
  
myData = [(1,'Harsha',2000,500),(2,'Mokshit',3000,500)]  
  
mySchema = ['id','name','salary','bonus']  
  
df = spark.createDataFrame(myData,mySchema)  
  
def totalPay(s,b):  
 return s+b  
  
totalPayment = udf(lambda s,b:totalPay(s,b),IntegerType())  
  
df.withColumn('totPay',totalPayment(df.salary,df.bonus)).show()

**Output:**

+---+-------+------+-----+------+

| id| name|salary|bonus|totPay|

+---+-------+------+-----+------+

| 1| Harsha| 2000| 500| 2500|

| 2|Mokshit| 3000| 500| 3500|

+---+-------+------+-----+------+

**Example-2**

from pyspark.sql import \*  
from pyspark.sql.functions import \*  
from pyspark.sql.types import \*  
  
*# Create a Spark session*spark = SparkSession.builder.appName("udf").getOrCreate()  
  
myData = [(1,'Harsha',2000,500),(2,'Mokshit',3000,500)]  
  
mySchema = ['id','name','salary','bonus']  
  
df = spark.createDataFrame(myData,mySchema)  
  
@udf(returnType=IntegerType())  
def totalPay(s,b):  
 return s+b  
  
df.select("\*",totalPay(df.salary,df.bonus).alias('Total\_paid')).show()

**Output:**

+---+-------+------+-----+----------+

| id| name|salary|bonus|Total\_paid|

+---+-------+------+-----+----------+

| 1| Harsha| 2000| 500| 2500|

| 2|Mokshit| 3000| 500| 3500|

+---+-------+------+-----+----------+

**Example-3**

from pyspark.sql import \*  
from pyspark.sql.functions import \*  
from pyspark.sql.types import \*  
  
*# Create a Spark session*spark = SparkSession.builder.appName("udf").getOrCreate()  
  
myData = [(1,'Harsha',2000,500),(2,'Mokshit',3000,500)]  
  
mySchema = ['id','name','salary','bonus']  
  
df = spark.createDataFrame(myData,mySchema)  
  
df.createOrReplaceTempView('emps')  
  
def Total\_pay(s,b):  
 return s+b  
  
spark.udf.register(name='Total\_pay\_sql',f=Total\_pay, returnType=IntegerType())  
  
spark.sql("SELECT id, name, Total\_pay\_sql(salary,bonus) As TotPay FROM emps").show()

**Output:**

+---+-------+------+

| id| name|TotPay|

+---+-------+------+

| 1| Harsha| 2500|

| 2|Mokshit| 3500|

+---+-------+------+