Master PySpark: From Zero to Big Data Hero!!

Trim Function in Dataframe

Let's create a new sample dataset for employees and demonstrate the usage of string trimming and padding functions (ltrim, rtrim, trim, lpad, and rpad) in PySpark.

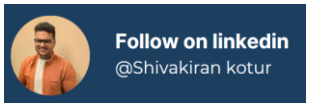
Steps:

1. Create sample employee data.
2. Demonstrate the usage of ltrim(), rtrim(), trim(), lpad(), and rpad() on string columns.

Sample Data Creation for Employees

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import lit, ltrim, rtrim, rpad, lpad,
trim, col

# Sample employee data with leading and trailing spaces in the
'Name' column
data = [
    (1, " Alice   ", "HR"),
    (2, "  Bob", "IT"),
    (3, "Charlie  ", "Finance"),
    (4, "  David ", "HR"),
    (5, "Eve  ", "IT")
]

# Define the schema for the DataFrame
columns = ["EmployeeID", "Name", "Department"]

# Create DataFrame
df = spark.createDataFrame(data, columns)

# Show the original DataFrame
df.show(truncate=False)
```

```
+----------+---------+----------+
|EmployeeID|Name     |Department|
+----------+---------+----------+
|1         | Alice   |HR        |
|2         |  Bob    |IT        |
|3         |Charlie  |Finance   |
|4         |  David  |HR        |
|5         |Eve      |IT        |
+----------+---------+----------+
```



Follow me on LinkedIn – Shivakiran kotur

Applying Trimming and Padding Functions

1. ltrim(), rtrim(), and trim():

- **ltrim()**: Removes leading spaces.
- **rtrim()**: Removes trailing spaces.
- **trim()**: Removes both leading and trailing spaces.

2. lpad() and rpad():

- **lpad()**: Pads the left side of a string with a specified character up to a certain length.
- **rpad()**: Pads the right side of a string with a specified character up to a certain length.

Example:

```
# Apply trimming and padding functions
result_df = df.select(
    col("EmployeeID"),
    col("Department"),
    ltrim(col("Name")).alias("ltrim_Name"),  # Remove leading spaces
    rtrim(col("Name")).alias("rtrim_Name"),  # Remove trailing spaces
    trim(col("Name")).alias("trim_Name"),    # Remove both leading and trailing spaces
    lpad(col("Name"), 10, "X").alias("lpad_Name"),  # Left pad with "X" to make the string length 10
    rpad(col("Name"), 10, "Y").alias("rpad_Name")   # Right pad with "Y" to make the string length 10
)

# Show the resulting DataFrame
result_df.show(truncate=False)
```

```
+----------+----------+----------+----------+---------+----------+----------+
|EmployeeID|Department|ltrim_Name|rtrim_Name|trim_Name|lpad_Name |rpad_Name |
+----------+----------+----------+----------+---------+----------+----------+
|1         |HR        |Alice     | Alice    |Alice    |X Alice   | Alice   Y|
|2         |IT        |Bob       |  Bob     |Bob      |XXXXX  Bob|  BobYYYYY|
|3         |Finance   |Charlie   |Charlie   |Charlie  |XCharlie  |Charlie  Y|
|4         |HR        |David     |  David   |David    |XX  David |  David YY|
|5         |IT        |Eve       |Eve       |Eve      |XXXXXEve  |Eve  YYYYY|
+----------+----------+----------+----------+---------+----------+----------+
```

Output Explanation:

- **ltrim_Name**: The leading spaces from the Name column are removed.
- **rtrim_Name**: The trailing spaces from the Name column are removed.
- **trim_Name**: Both leading and trailing spaces are removed from the Name column.
- **lpad_Name**: The Name column is padded on the left with "X" until the string length becomes 10.
- **rpad_Name**: The Name column is padded on the right with "Y" until the string length becomes 10.