Show() Function

from pyspark.sql import \*  
  
spark = SparkSession.builder.appName('show').getOrCreate()  
  
data = [(1,'PySpark is the Python API for Apache Spark'),  
 (2,'Allows Python developers to write Spark applications'),  
 (3,'Enables distributed computing'),  
 (4,'PySpark introduces Resilient Distributed Datasets (RDDs)'),  
 (5,'PySpark also offers a higher-level DataFrame API'),  
 (6,'DataFrames are distributed collections of data'),  
 (7,'PySpark supports real-time data processing')]  
  
schema = ['id','comments']  
  
df = spark.createDataFrame(data = data, schema = schema)  
  
*#=====================================================================  
# By default show() displays the first 20 rows of the DataFrame or DataSet*df.show()

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

| id| comments|

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

| 1|PySpark is the Py...|

| 2|Allows Python dev...|

| 3|Enables distribut...|

| 4|PySpark introduce...|

| 5|PySpark also offe...|

| 6|DataFrames are di...|

| 7|PySpark supports ...|

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

*#=====================================================================  
# Customizing the number of rows to be displayed*df.show(n=3)

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

| id| comments|

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

| 1|PySpark is the Py...|

| 2|Allows Python dev...|

| 3|Enables distribut...|

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

only showing top 3 rows

*#=====================================================================  
# Truncating long strings*'''  
To control the truncation of the long strings in the displayed table, we can use the truncate parameter. By default it is set to true and truncates strings longer than 20 char's.  
'''  
df.show(n=5, truncate=False)  
*# This will display the first 5 rows of the DataFrame df without truncating long string values.*+---+--------------------------------------------------------+

|id |comments |

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

|1 |PySpark is the Python API for Apache Spark |

|2 |Allows Python developers to write Spark applications |

|3 |Enables distributed computing |

|4 |PySpark introduces Resilient Distributed Datasets (RDDs)|

|5 |PySpark also offers a higher-level DataFrame API |

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

*#==================================================================  
# Changing Column Alignment*'''  
You can modify the alignment of the columns in the displayed table using the vertical parameter. By default, it is set to False and aligns columns horizontally.   
Setting it to True aligns columns vertically.   
'''  
  
df.show(n=5,vertical=True)  
  
*# This will display the first 5 rows of the DataFrame df with vertically aligned columns.*-RECORD 0------------------------

id | 1

comments | PySpark is the Py...

-RECORD 1------------------------

id | 2

comments | Allows Python dev...

-RECORD 2------------------------

id | 3

comments | Enables distribut...

-RECORD 3------------------------

id | 4

comments | PySpark introduce...

-RECORD 4------------------------

id | 5

comments | PySpark also offe...

only showing top 5 rows

*#=================================================================  
# Chaining DataFrame Operations*'''  
show() can be used in conjunction with other DataFrame operations like we can apply filtering or aggregation before displaying the results.  
'''  
  
df.filter(df.age > 30).groupBy('gender').count().show()  
  
'''  
This code filters the DataFrame df to keep only rows where the age is greater than 30, then groups the data by the 'gender' column and counts the occurrences.   
Finally, show() is called to display the result.  
'''