PySpark Cheat Sheet

Draphony

1. Basic Operations

```
# Create a SparkSession
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('your_app_name').getOrCreate()

# Load Data
df = spark.read.csv('/path/to/input.csv', header=True)  # CSV
df = spark.read.json('/path/to/input.json')             # JSON
df = spark.read.parquet('/path/to/input.parquet')       # Parquet

# Show data
df.show(5)    # Show the first 5 rows
df.collect()  # Collect all rows (use with caution)
```

2. Selecting and Filtering Data

```
# Select columns
df = df.select('column1', 'column2')

# Filter rows
df = df.filter(df.column > 10)
df = df.filter(df.column.isNull())
df = df.filter(df.column.isNotNull())

# Drop duplicates
df = df.dropDuplicates()

# Drop a column
df = df.drop('column_name')

# Sort by a column
df = df.orderBy(df.column.asc())
df = df.orderBy(df.column.desc())
```

3. Joining DataFrames

```
# Inner join
df = df1.join(df2, df1.id == df2.id, 'inner')

# Left join
df = df1.join(df2, df1.id == df2.id, 'left')

# Full outer join
df = df1.join(df2, df1.id == df2.id, 'outer')
```

4. Aggregations

```
from pyspark.sql import functions as F

# Aggregations
df = df.groupBy('column').agg(F.count('*').alias('count'))
df = df.groupBy('column').agg(F.sum('amount').alias('totalAmount'))

# Basic Aggregations
df = df.agg(F.max('column').alias('max'))
df = df.agg(F.min('column').alias('min'))
```

5. Window Functions

```
from pyspark.sql.window import Window

# Define a window
windowSpec = Window.partitionBy('column').orderBy('column2')

# Row number within window
df = df.withColumn('row_number', F.row_number().over(windowSpec))

# Running total within window
df = df.withColumn('running_total', F.sum('amount').over(windowSpec))
```

6. Conditional Statements

```
df = df.withColumn('status', F.when(df['column'] > 50, 'High')
                             .when(df['column'] > 20, 'Medium')
                             .otherwise('Low'))
```

7. String Functions

```
df = df.withColumn('upper_col', F.upper('column'))
df = df.withColumn('lower_col', F.lower('column'))
df = df.withColumn('substr_col', F.substring('column', 1, 3))
df = df.withColumn('trim_col', F.trim('column'))
df = df.withColumn('concat_col', F.concat(F.col('column1'), F.col('column2')))
```

8. Number Functions

```
df = df.withColumn('round_col', F.round('column', 0))
df = df.withColumn('floor_col', F.floor('column'))
df = df.withColumn('ceil_col', F.ceil('column'))
df = df.withColumn('abs_col', F.abs('column'))
df = df.withColumn('sqrt_col', F.sqrt('column'))
```

9. Date & Time Functions

```
# Add a column with the current date and time
df = df.withColumn('current_date', F.current_date())
df = df.withColumn('current_timestamp', F.current_timestamp())

# Convert a string to a date
df = df.withColumn('date_col', F.to_date('string_col', 'yyyy-MM-dd'))

# Convert a string to a timestamp
df = df.withColumn('ts_col', F.to_timestamp('string_col','yyyy-MM-dd HH:mm:ss'))

# Get number of days between two dates
df = df.withColumn('date_diff', F.datediff('end_date', 'start_date'))

# Get number of months between two dates
df = df.withColumn('months_between', F.months_between('date1', 'date2'))
```

10. Column Operations

```
# Add or rename a column
df = df.withColumn('new_column', F.lit('value'))
df = df.withColumnRenamed('old_name', 'new_name')

# Mathematical operations
df = df.withColumn('rounded', F.round('column', 2))
df = df.withColumn('absolute', F.abs('column'))

# Replace values in a column
df = df.replace(float("nan"), None)
```

11. Write data

```
# Write DataFrame to different formats
df.write.csv('/path/to/output.csv', header=True) # CSV
df.write.json('/path/to/output.json')            # JSON
df.write.parquet('/path/to/output.parquet')      # Parquet
```

12. Ending Spark

```
# Cache DataFrame in memory
df.cache()

# Remove DataFrame from memory
df.unpersist()

# Stop SparkSession
spark.stop()
```

https://draphony.de/