PySpark Fingertip Commands Cheat Sheet

Reading data from a file

```
df = spark.read.csv("file.csv", header=True)
df =
spark.r ea d.p arq uet ("fi le.p ar que
t", header =True)
```

Casting a column to a different data type

```
df.withColumn("col1", col("col1").cast("double"))
```

Displaying the schema of a Dataframe

```
df.printSchema()
```

Get distinct count of columns

```
df.select("col").distinct().count()
```

Filtering rows based on a condition

```
# Filter entries of age, only keep those records
of
   which the values are >24
     df.f il ter (df ["ag e"]> 24).show()
```

Renaming columns of DataFrame

```
#Syntax
df.wit hCo lum nRe nam ed( " old _na me",
  " new -
_na me")
#Example
df =
df.wit hCo lum nRe nam ed( 'Ca llN umber
', 'Phone Num ber')
```

Inspect Data

```
# Return first n rows
     df.h ead()
# Return first row
     df.f irst()
# Return the first n rows
     df.t ake(2)
# Print the schema of df
     df.p ri ntS chema()
# Print the (logical and physical) plans
   df.e xp lain()
#Get All column names from DataFrame
     df.c olumns
```

Get count

```
# Get_row count
   rows = empDF.c ount()
# Get_co lumns count
   cols = len(em pDF.co lumns)
```

Selecting specific columns of a Dataframe

```
df.select("col1","col2").show()
 # Select All columns
df.sel ect ("*").sho
w()
```

Full content of the columns without truncation

```
df. show(t run cat e=F alse)
```

Handling missing or null values

```
# Fill all null values with 0
     df.f il lna(0)
  #Fill specific columns with specified values
     df.f il lna ({col1: 0,
   " col 2":"m iss -
ing "})
```

Joining two dataframes

```
#syntax:
     joi ned_df = df1.jo in(df2,
   on=" key _co lum -
n", how="in ner ")
#example
     joi ned_df =
   empDF.j oi n(d ept DF, emp DF.e -
mp _de pt_id == deptDF.de pt_ id, " inn er")
\
          .sh ow( tru nca te= False)
```

Adding a new column to a DataFrame

```
df.withColumn("new_col", col("col1")
+ col("col2"))
```

Dropping columns from a Dataframe

```
df.drop("col1")
```

PySpark Fingertip Commands Cheat Sheet

Grouping data by a colm and agg. with a function

```
#syntax
df.gro upB y("c ol1 " ).a gg( {"co l2": " mea n"})
#examples
df.gro upB y("d epa rtm ent ") \
       .ag g(s um( " sal ary " ).a lia s("s um_ -
sal ary "), \
                  avg ("sa lar y").a li as( " -
avg _sa lar y"), \
                  sum ("bo nus " ).a lia s("s um_ -
bon us"), \
                  max ("bo nus " ).a lia s("m ax_ -
bon us") \
       )\
       .sh ow( tru nca te= False)
```

Stopping the SparkS ession

```
spark.stop()
```