# PKA-WORDCOUNT

January 11, 2024

[1]:
```
import pyspark
import os
import sys
from pyspark import SparkContext
from pyspark import SparkConf
os.environ['PYSPARK_PYTHON'] = sys.executable
os.environ['PYSPARK_DRIVER_PYTHON'] = sys.executable
```

[4]:
```
# Create a SparkSession

sc = SparkContext.getOrCreate(SparkConf().setMaster("local[*]"))
# Load the text file
lines =  sc.textFile("sentence.txt")
counts = lines.flatMap(lambda line: line.split(" ")) \
                            .map(lambda word: (word, 1)) \
                           .reduceByKey(lambda x, y: x + y)
output = counts.collect()
for (word, count) in output:
    print("%s: %i" % (word, count))

# Split the lines into words
```

rdd

```
to: 4
be: 4
or: 2
not: 2
```

[11]:
```
from pyspark.sql import SparkSession
from pyspark.sql import functions as f
# Create a SparkSession
spark = SparkSession.builder.getOrCreate()

# Load the text file
lines = spark.read.text("sentence.txt")

# Split the lines into words
words = lines.withColumn('word', f.explode(f.split(f.col('value'), ' ')))\
    .groupBy('word')\
```

data frame

```
    .count()\
    .sort('count', ascending=False)\
    .show()



# Stop the SparkSession
spark.stop()
```

```
+----+-----+
|word|count|
+----+-----+
|  be|    4|
|  to|    4|
| not|    2|
|  or|    2|
+----+-----+
```