```
# #Running on Colab
!pip install pyspark
!pip install -U -q PyDrive
!apt install openjdk-8-jdk-headless -qq
import os
os.environ['JAVA_HOME'] = '/usr/lib/jvm/java-8-openjdk-amd64'
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 2.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=51434d7aee17c9e69f7266858ac0c8a2e9f7811d66f065a13cfa7fcc3813d10d
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
The following additional packages will be installed:
  libxtst6 openjdk-8-jre-headless
Suggested packages:
  openjdk-8-demo openjdk-8-source libnss-mdns fonts-dejavu-extra fonts-nanum fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  libxtst6 openjdk-8-jdk-headless openjdk-8-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 39 not upgraded.
Need to get 39.7 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Selecting previously unselected package libxtst6:amd64.
(Reading database ... 121753 files and directories currently installed.)
Preparing to unpack .../libxtst6_2%3a1.2.3-1build4_amd64.deb ...
Unpacking libxtst6:amd64 (2:1.2.3-1build4) ...
Selecting previously unselected package openjdk-8-jre-headless:amd64.
Preparing to unpack .../openjdk-8-jre-headless_8u402-ga-2ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-8-jre-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
Selecting previously unselected package openjdk-8-jdk-headless:amd64.
Preparing to unpack .../openjdk-8-jdk-headless_8u402-ga-2ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-8-jdk-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
Setting up libxtst6:amd64 (2:1.2.3-1build4) ...
Setting up openjdk-8-jre-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to provide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/servertool to provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv to provide /usr/bin/tnameserv (tnameserv) in auto mode
Setting up openjdk-8-jdk-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/clhsdb to provide /usr/bin/clhsdb (clhsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/extcheck to provide /usr/bin/extcheck (extcheck) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/hsdb to provide /usr/bin/hsdb (hsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/idlj to provide /usr/bin/idlj (idlj) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javah to provide /usr/bin/javah (javah) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jhat to provide /usr/bin/jhat (jhat) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jsadebugd to provide /usr/bin/jsadebugd (jsadebugd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/native2ascii to provide /usr/bin/native2ascii (native2ascii) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/schemagen to provide /usr/bin/schemagen (schemagen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen (wsgen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsimport to provide /usr/bin/wsimport (wsimport) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/xjc to provide /usr/bin/xjc (xjc) in auto mode
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link
```

```
#  Import the libraries we will need
import pyspark
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark import SparkContext, SparkConf

# Create Spark session and ContextRun PySpark.
# create the session
conf = SparkConf().set("spark.ui.port","4050")
# create the context
sc = pyspark.SparkContext(conf=conf)
spark = SparkSession.builder.appName("DataFrame").config('spark.ui.port', '4050').getOrCreate()
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell

```
rdd1 = sc.parallelize([(0,2), (3,4), (5,6), (20,8), (2,2), (3,4), (6,6), (8,8),(4,2), (12,4), (15,6), (7,8)], 3)
print('Number of partitions:{} '.format(rdd1.getNumPartitions()))
print('Partitioner: {}'.format(rdd1.partitioner))
print('Partitions structure: {}'.format(rdd1.glom().collect()))
```

```
Number of partitions:3
Partitioner: None
Partitions structure: [[(0, 2), (3, 4), (5, 6), (20, 8)], [(2, 2), (3, 4), (6, 6), (8, 8)], [(4, 2), (12, 4), (15, 6), (7, 8)]]
```

```
rdd = sc.parallelize([(0,2), (3,4), (5,6), (20,8), (2,2), (3,4), (6,6), (8,8),
(4,2), (12,4), (15,6), (7,8)], 3)
rdd1 = rdd.partitionBy(5)
print('Number of partitions: {}'.format(rdd1.getNumPartitions()))
print('Partitioner: {}'.format(rdd1.partitioner))
print('Partitions structure: {}'.format(rdd1.glom().collect()))
```

```
Number of partitions: 5
Partitioner: <pyspark.rdd.Partitioner object at 0x78cf0414ab00>
Partitions structure: [[(0, 2), (5, 6), (20, 8), (15, 6)], [(6, 6)], [(2, 2), (12, 4), (7, 8)], [(3, 4), (3, 4), (8, 8)], [(4, 2)]]
```

```
# Custome Partitioner
def partFunc(k):
  if(k % 2 == 0 ):
    return 0
  else:
    return 1
rdd = sc.parallelize([(0,2), (3,4), (5,6), (20,8), (2,2), (3,4),
(6,6), (8,8), (4,2), (12,4), (15,6), (7,8)])
rdd1 = rdd.partitionBy(2,lambda x: partFunc(x)).persist()
print('Number of partitions: {}'.format(rdd1.getNumPartitions()))
print('Partitioner: {}'.format(rdd1.partitioner))
print('Partitions structure: {}'.format(rdd1.glom().collect()))
```

```
Number of partitions: 2
Partitioner: <pyspark.rdd.Partitioner object at 0x78cee2aad900>
Partitions structure: [[(0, 2), (20, 8), (2, 2), (6, 6), (8, 8), (4, 2), (12, 4)], [(3, 4), (5, 6), (3, 4), (15, 6), (7, 8)]]
```

```
# Accumulators  & Boardcast
counter = sc.accumulator(0)

# Define broadcast variable
data_to_broadcast = [1, 2, 3, 4, 5]
broadcast_var = sc.broadcast(data_to_broadcast)

# Sample RDD to demonstrate accumulator and broadcast variable usage
rdd = sc.parallelize(range(10))

# Function to increment accumulator and use broadcast variable
def process_element(x):
    global counter
    global broadcast_var

    # Increment accumulator
    counter += 1

    # Access and use broadcast variable
    broadcast_data = broadcast_var.value
    return x * broadcast_data[1]

# Apply function to each element in RDD
result = rdd.map(process_element)

# Collect and print result
print(result.collect())

# Print accumulator value
print("Accumulator value:", counter.value)
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
Accumulator value: 10
```

```
# Word co-occurence in Spark
# myFile.txt is as follows:
# great first try super boy
# first try good
# super boy won first try

input = sc.textFile("myFile.txt")
co = input.map(lambda x:x.split(" "))
co.collect()
```

```
[['great', 'first', 'try', 'super', 'boy'],
 ['first', 'try', 'good'],
 ['super', 'boy', 'won', 'first', 'try']]
```

```
input = sc.textFile("myFile.txt")
co = input.flatMap(lambda x:x.split(" "))
co.collect()
```

```
['great',
 'first',
 'try',
 'super',
 'boy',
 'first',
 'try',
 'good',
 'super',
 'boy',
 'won',
 'first',
 'try']
```

```
input = sc.textFile("myFile.txt")
co = input.map(lambda x:x.split(" "))

def func(line):
  value =[]
  for i in range(len(line)-1):
    for j in range(i+1,len(line)):
      value.append(((line[i],line[j]),1))
  return value

co2 = co.flatMap(func)
co3 =co2.reduceByKey(lambda x,y:x+y)
co3.collect()
```

```
[(('great', 'first'), 1),
 (('try', 'super'), 1),
 (('try', 'boy'), 1),
 (('super', 'boy'), 2),
 (('try', 'good'), 1),
 (('super', 'try'), 1),
 (('boy', 'try'), 1),
 (('won', 'first'), 1),
 (('great', 'try'), 1),
 (('great', 'super'), 1),
 (('great', 'boy'), 1),
 (('first', 'try'), 3),
 (('first', 'super'), 1),
 (('first', 'boy'), 1),
 (('first', 'good'), 1),
 (('super', 'won'), 1),
 (('super', 'first'), 1),
 (('boy', 'won'), 1),
 (('boy', 'first'), 1),
 (('won', 'try'), 1)]
```

```
# in map reduce
from mrjob.job import MRJob

class WordPairsMRJob(MRJob):
    def mapper(self, _, line):
        words = line.split(" ")
        # Emit all pairs of words in the line
        for i in range(len(words) - 1):
            for j in range(i + 1, len(words)):
                # Ensure that the pair is emitted
                yield tuple(([words[i], words[j]])), 1

    def reducer(self, key, values):
        # Sum up the counts for each pair
        yield key, sum(values)

if __name__ == '__main__':
    WordPairsMRJob.run()
```

```
# Solution to same pair different order issue (E.g boy try and try boy are same pairs)
# IDEA: Generate pair so that words are in sorted order

input = sc.textFile("myFile.txt")
co = input.map(lambda x:x.split(" "))

def func(line):
  value =[]
  for i in range(len(line)-1):
    for j in range(i+1,len(line)):
      if(line[i]<=line[j]):
        value.append(((line[i],line[j]),1))
      else:
        value.append(((line[j],line[i]),1))
  return value

co2 = co.flatMap(func)
co3 =co2.reduceByKey(lambda x,y:x+y)
co3.collect()
```

```
[(('first', 'great'), 1),
 (('super', 'try'), 2),
 (('boy', 'try'), 2),
 (('boy', 'super'), 2),
 (('good', 'try'), 1),
 (('first', 'won'), 1),
 (('great', 'try'), 1),
 (('great', 'super'), 1),
 (('boy', 'great'), 1),
 (('first', 'try'), 3),
 (('first', 'super'), 2),
 (('boy', 'first'), 2),
 (('first', 'good'), 1),
 (('super', 'won'), 1),
 (('boy', 'won'), 1),
 (('try', 'won'), 1)]
```

```
# Pi calculation problem
import random
partitions = 2
n = 100000 * partitions

def func(_):
    x = random.random() * 2 - 1
    y = random.random() * 2 - 1 #Range between -1 to 1
    return 1 if x ** 2 + y ** 2 <= 1 else 0

sample = spark.sparkContext.parallelize(range(1, n + 1), partitions)
count = sample.map(func).reduce(lambda x, y: x + y)
print("Pi is roughly %f" % (4.0 * count / n))
```

```
Pi is roughly 3.143360
```

```
# PI PROBLEM IN MAP REDUCE
from mrjob.job import MRJob
import random

class PiCalculationMRJob(MRJob):

    def mapper(self, _, __):
        partitions = 2
        n = 100000 * partitions

        # Generate random points and determine if they fall inside the unit circle
        for _ in range(n):
            x = random.random() * 2 - 1
            y = random.random() * 2 - 1
            yield None, (1 if x ** 2 + y ** 2 <= 1 else 0)

    def combiner(self, _, counts):
        # Local aggregation of counts of points inside the unit circle
        total_points = 0
        points_inside_circle = 0
        for count in counts:
            total_points += 1
            points_inside_circle += count
        yield None, (total_points, points_inside_circle)

    def reducer(self, _, counts):
        # Aggregate counts of points inside the unit circle received from combiners
        total_points = 0
        points_inside_circle = 0
        for count in counts:
            total_points += count[0]
            points_inside_circle += count[1]
        yield None, 4.0 * points_inside_circle / total_points

if __name__ == '__main__':
    PiCalculationMRJob.run()
```