```
In [1]:   import findspark
          from pyspark.streaming import StreamingContext
          from pyspark.sql import SparkSession
          import math

          findspark.init()
```

```
In [29]:  try: ssc.stop(True, True)
          except: pass
          try: spark.stop()
          except: pass
```

```
[]
[]
[]
[]
[]
[]
[]
[]
[]
[]
[]
```

```
In [31]:  spark=SparkSession.builder.appName("SparkStreaming-02").master('local[1]').getOrCreate()
          sc=spark.sparkContext
          # .config("spark.driver.allowMultipleContexts","true")
          spark
```

Out[31]: **SparkSession - in-memory**

SparkContext

Spark UI

Version	v2.4.8
Master	local[1]
AppName	SparkStreaming-02

```
In [23]:  ssc=StreamingContext(sc, 1)
          # ssc.checkpoint('checkpoint-01') compulsory in stateful transformations only
          ssc # 1=1 second
```

Out[23]: `<pyspark.streaming.context.StreamingContext at 0x2412fc49608>`

```
In [24]:  stream=ssc.queueStream([
              sc.parallelize([(1,'a'),(2,'b'),(1,'c'),(2,'d'),(1,'e'),(3,'f')]),
              sc.parallelize([(2,'h'),(2,'i'),(1,'j'),(2,'k'),(1,'l'),(3,'m')])
          ],oneAtATime=True)
```

```
In [25]:  stream2=ssc.queueStream([
              ssc.sparkContext.parallelize([(1,'a'),(2,'b'),(1,'c'),(2,'d'),(1,'e'),(3,'f')]),
              ssc.sparkContext.parallelize([(2,'h'),(2,'i'),(1,'j'),(2,'k'),(1,'l'),(3,'m')])
          ],oneAtATime=False)
```

```
In [26]:   maxstream=stream.reduceByKey(lambda x,y: x+y)

           # maxstream.pprint(2)
           # print(maxstream.pprint(2))

           def printrdd(rdd):
               print(rdd.take(2))
           maxstream.foreachRDD(printrdd)
```

```
In [27]:   def mapValues(rdd):
               rdd=rdd.map(lambda x: (x[1],x[0]))
               return rdd
           maxstream=maxstream.transform(mapValues)

           # maxstream.pprint(2)
           def printrdd(rdd):
               print(rdd.collect())
           maxstream.foreachRDD(printrdd)
```

```
In [28]:   ssc.start()
```

```
[(1, 'ace'), (2, 'bd')]
[('ace', 1), ('bd', 2), ('f', 3)]
[(2, 'hik'), (1, 'jl')]
[('hik', 2), ('jl', 1), ('m', 3)]
[]
[]
[]
```

```
In [199…   lines=ssc.socketTextStream("localhost", 8000)
           lines.pprint()
```

```
In [200…   ssc.start()
```

```
In [76]:
```

```
In [ ]:
```