

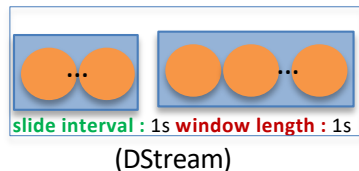
# Spark Streaming API Programming model (Illustration)



Spark Streaming API for Python: **pyspark.streaming**

**DStream** object represents the streaming data

**DStream** type provides **transformation** and **actions** like RDD



```
1  from pyspark import SparkContext
2  from pyspark.streaming import StreamingContext
3
4  # 1. Create StreamingContext
5  ssc = StreamingContext(1)
6
7  # 2. Connect to the data source
8  lines = ssc.socketTextStream("devenv", 9999)
9
10 # 3. Analysis with transformans and actions
11 words = lines.flatMap(f1)
12 pairs = words.map(f2)
13 word_counts = pairs.reduceByKey(f3)
14
15 word_counts.pprint(30)
16
17 # 4. Start the computation
18 ssc.start()
19 ssc.awaitTermination()
```

**batch interval = 1S**  
Decide the length of the RDD by time period.

# Spark Streaming API Programming model (Illustration)

😊 Spark Streaming API for Python: **pyspark.sql**

`spark-submit --master ... network_wordcount.py`

```
1 from pyspark import SparkContext
2 from pyspark.streaming import StreamingContext
3
4 # 1. Create StreamingContext
5 ssc = StreamingContext(1)
6
7 # 2. Connect to the data source
8 lines = ssc.socketTextStream("devenv", 9999)
9
10 # 3. Analysis with transformans and actions
11 words = lines.flatMap(f1)
12 pairs = words.map(f2)
13 word_counts = pairs.reduceByKey(f3)
14
15 word_counts.pprint(30)
16
17 # 4. Start the computation
18 ssc.start()
19 ssc.awaitTermination()
```

```
1 from pyspark import SparkContext
2 from pyspark.streaming import StreamingContext
3
4 # 1. Create StreamingContext
5 ssc = StreamingContext(1)
6
7 # 2. Connect to the data source
8 lines = ssc.socketTextStream("devenv", 9999)
9
10 # 3. Analysis with transformans and actions
11 words = lines.flatMap(f1)
12 pairs = words.map(f2)
13 word_counts = pairs.reduceByKey(f3)
14
15 word_counts.pprint(30)
16
17 # 4. Start the computation
18 ssc.start()
19 ssc.awaitTermination()
```

Driver

Standalone (/YARN/Mesos)

