```
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Exploration")
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
File <command-4287340436422431>:2
      1 # Create a SparkContext
----> 2 sc = SparkContext("local", "RDD Exploration")

File /databricks/spark/python/pyspark/context.py:202, in
SparkContext.__init__(self, master, appName, sparkHome, pyFiles,
environment, batchSize, serializer, conf, gateway, jsc, profiler_cls,
udf_profiler_cls, memory_profiler_cls)
    196 if gateway is not None and
gateway.gateway_parameters.auth_token is None:
    197     raise ValueError(
    198         "You are trying to pass an insecure Py4j gateway to
Spark. This"
    199         " is not allowed as it is a security risk."
    200     )
--> 202 SparkContext._ensure_initialized(self, gateway=gateway,
conf=conf)
    203 try:
    204     self._do_init(
    205         master,
    206         appName,
   (...)
    216         memory_profiler_cls,
    217     )

File /databricks/spark/python/pyspark/context.py:488, in
SparkContext._ensure_initialized(cls, instance, gateway, conf)
    485     callsite = SparkContext._active_spark_context._callsite
    487     # Raise error if there is already a running Spark context
--> 488     raise ValueError(
    489         "Cannot run multiple SparkContexts at once; "
    490         "existing SparkContext(app=%s, master=%s)"
    491         " created by %s at %s:%s "
    492         % (
    493             currentAppName,
    494             currentMaster,
    495             callsite.function,
    496             callsite.file,
    497             callsite.linenum,
    498         )
    499     )
```

```
    500 else:
    501     SparkContext._active_spark_context = instance

ValueError: Cannot run multiple SparkContexts at once; existing
SparkContext(app=Databricks Shell, master=local[8]) created by
__init__ at /databricks/python_shell/dbruntime/spark_connection.py:127
```

```
# Create an RDD from a list
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Perform some basic operations on the RDD
# 1. Count the number of elements
count = rdd.count()
print("Number of elements:", count)
```

```
Number of elements: 5
```

```
# 2. Sum all elements
total_sum = rdd.sum()
print("Sum of all elements:", total_sum)
```

```
Sum of all elements: 15
```

```
# 3. Calculate the mean
mean = total_sum / count
print("Mean of elements:", mean)
```

```
Mean of elements: 3.0
```

```
# 4. Find the maximum and minimum elements
max_element = rdd.max()
min_element = rdd.min()
print("Maximum element:", max_element)
print("Minimum element:", min_element)
```

```
Maximum element: 5
Minimum element: 1
```

```
# 5. Filter elements greater than 3
filtered_rdd = rdd.filter(lambda x: x > 3)
print("Elements greater than 3:", filtered_rdd.collect())
```

```
Elements greater than 3: [4, 5]
```

```
# 6. Map operation to square each element
squared_rdd = rdd.map(lambda x: x*x)
print("Squared elements:", squared_rdd.collect())
```

```
Squared elements: [1, 4, 9, 16, 25]
```

```python
# 7. Reduce operation to find the sum of elements
sum_using_reduce = rdd.reduce(lambda x, y: x + y)
print("Sum of elements using reduce:", sum_using_reduce)
```

```
Sum of elements using reduce: 15
```