

Experiment No :- 06

Title: Apache Spark

Aim: Installation and configuration of Apache Spark on Local Machine. Execute basic commands used in Spark.

Theory:

Apache Spark

What is Apache Spark?

Apache Spark is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

Key features

Batch/streaming data

Unify the processing of your data in batches and real-time streaming, using your preferred language: Python, SQL, Scala, Java or R.

SQL analytics

Execute fast, distributed ANSI SQL queries for dashboarding and ad-hoc reporting. Runs faster than most data warehouses.

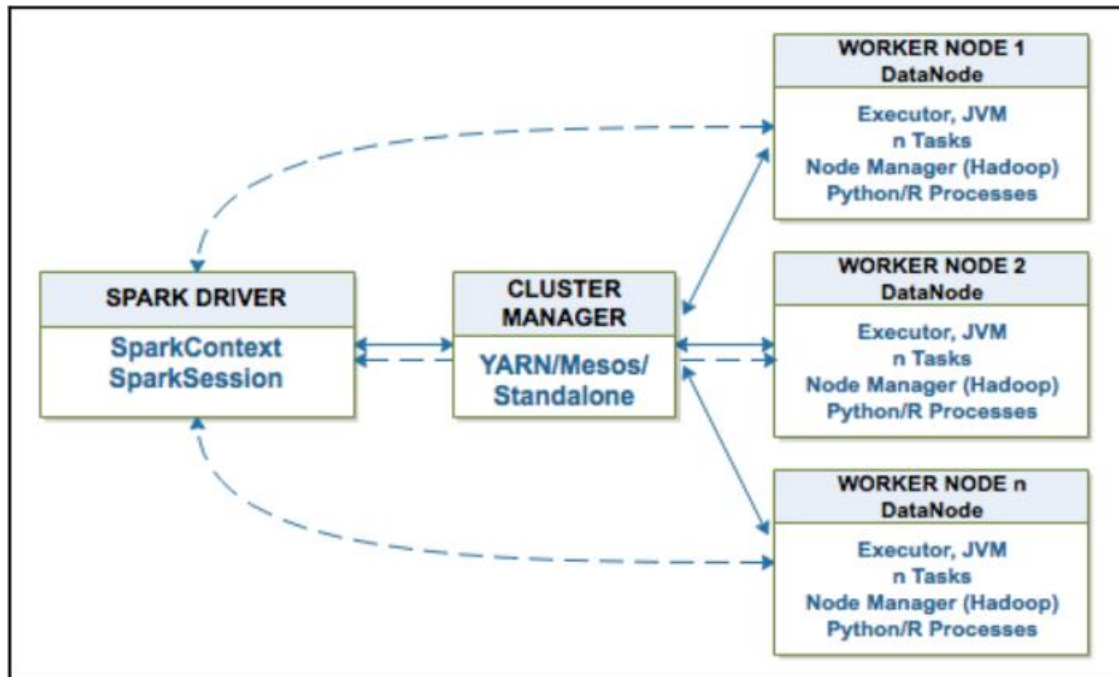
Data science at scale

Perform Exploratory Data Analysis (EDA) on petabyte-scale data without having to resort to downsampling

Machine learning

Train machine learning algorithms on a laptop and use the same code to scale to fault-tolerant clusters of thousands of machines.

The architecture of Spark

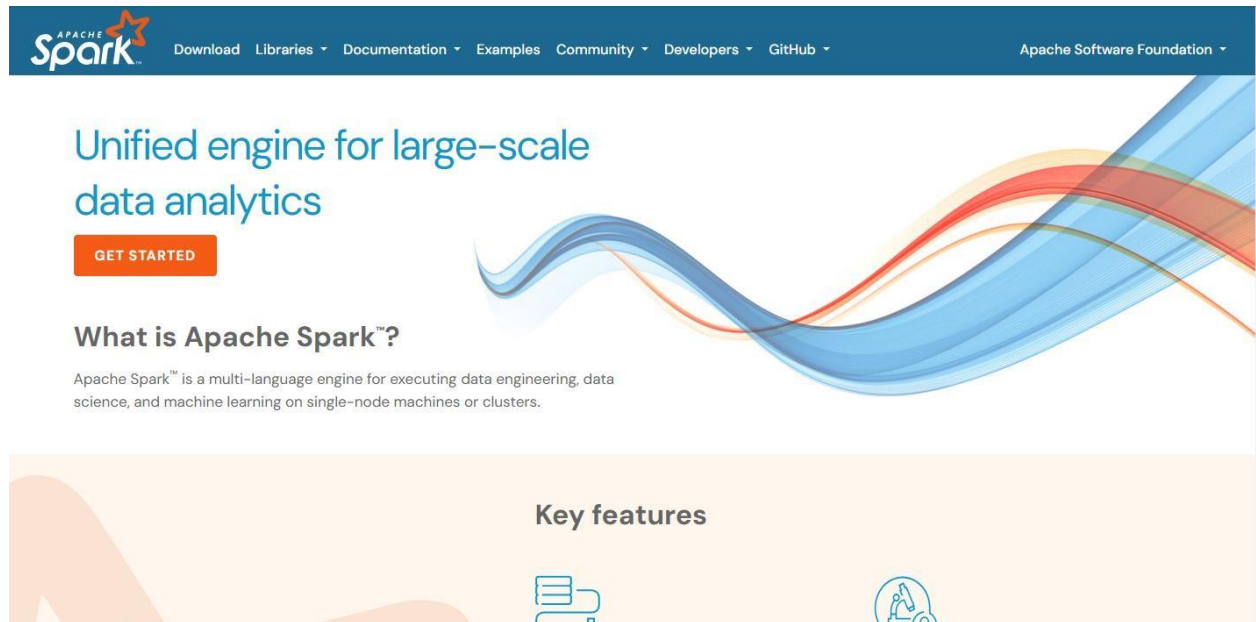


Spark's architecture consists of three primary components:

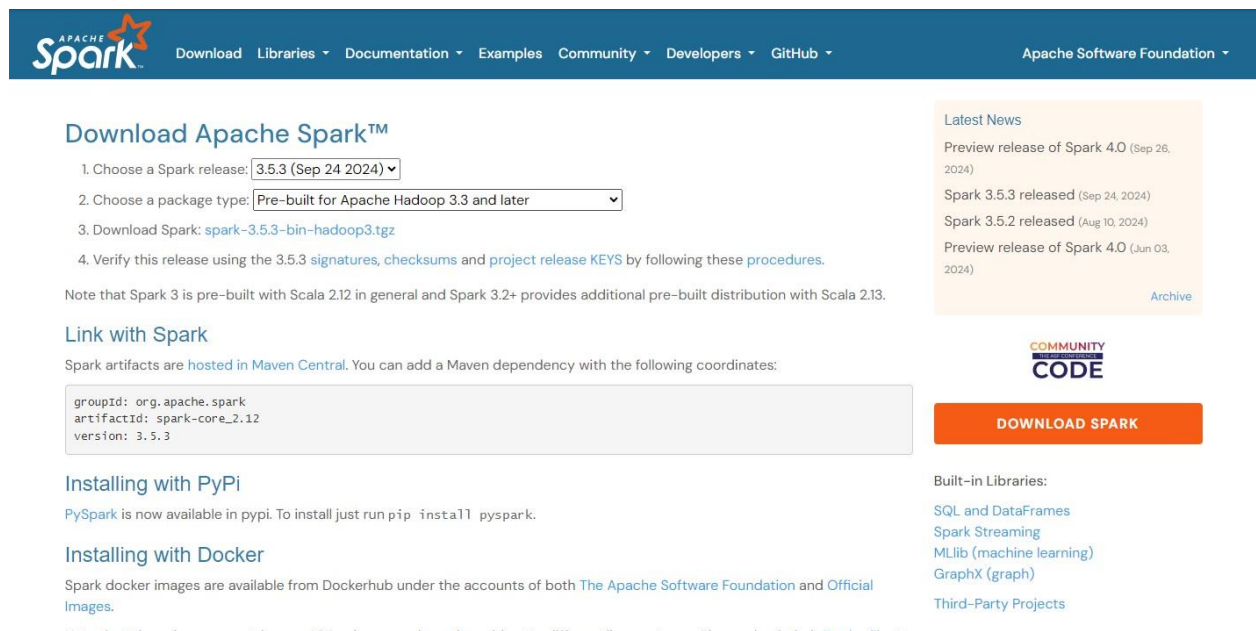
- 1. SparkSession/SparkContext (Driver):** The entry point for Spark applications. The driver creates RDDs, performs operations, and coordinates tasks by sending instructions to worker nodes.
- 2. Cluster Manager:** Manages resources and communication between worker nodes. It can be YARN, Mesos, or run in standalone mode. It handles node administration tasks like starting and stopping nodes.
- 3. Worker Nodes:** Hosts Spark's executor processes that perform tasks (actions and transformations). Each application has its own executor process to ensure isolation. The worker nodes include the Executor, JVM, and application-specific processes like Python or R.

Installation Of Apache Spark:

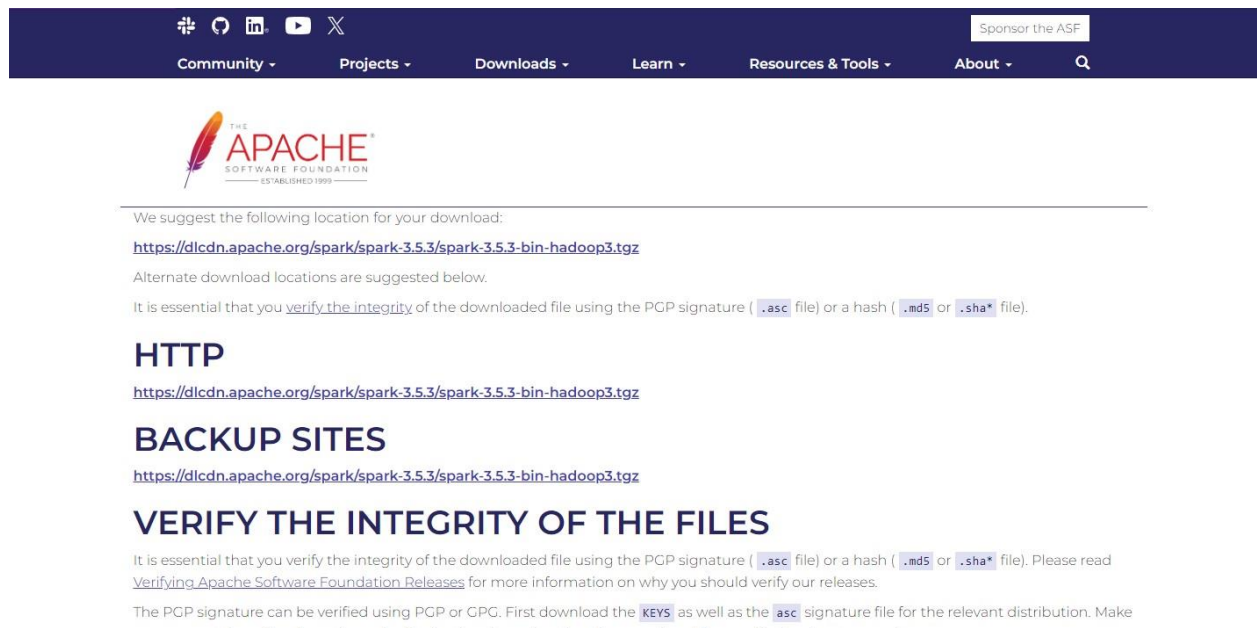
Step 1 :- Visit the official website to download [Apache spark](https://spark.apache.org/)



2. Click the Download a release now! link to access the mirrors page.



3. Choose the default mirror link.



The screenshot shows the Apache Spark download page. At the top is a dark blue navigation bar with links for Community, Projects, Downloads, Learn, Resources & Tools, About, and a search icon. Below the navigation bar is the Apache Software Foundation logo. The main content area has a light blue background and contains the following text:

We suggest the following location for your download:
<https://d1cdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz>

Alternate download locations are suggested below.

It is essential that you [verify the integrity](#) of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

HTTP

<https://d1cdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz>

BACKUP SITES

<https://d1cdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz>

VERIFY THE INTEGRITY OF THE FILES

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file). Please read [Verifying Apache Software Foundation Releases](#) for more information on why you should verify our releases.

The PGP signature can be verified using PGP or GPG. First download the `KEYS` as well as the `.asc` signature file for the relevant distribution. Make

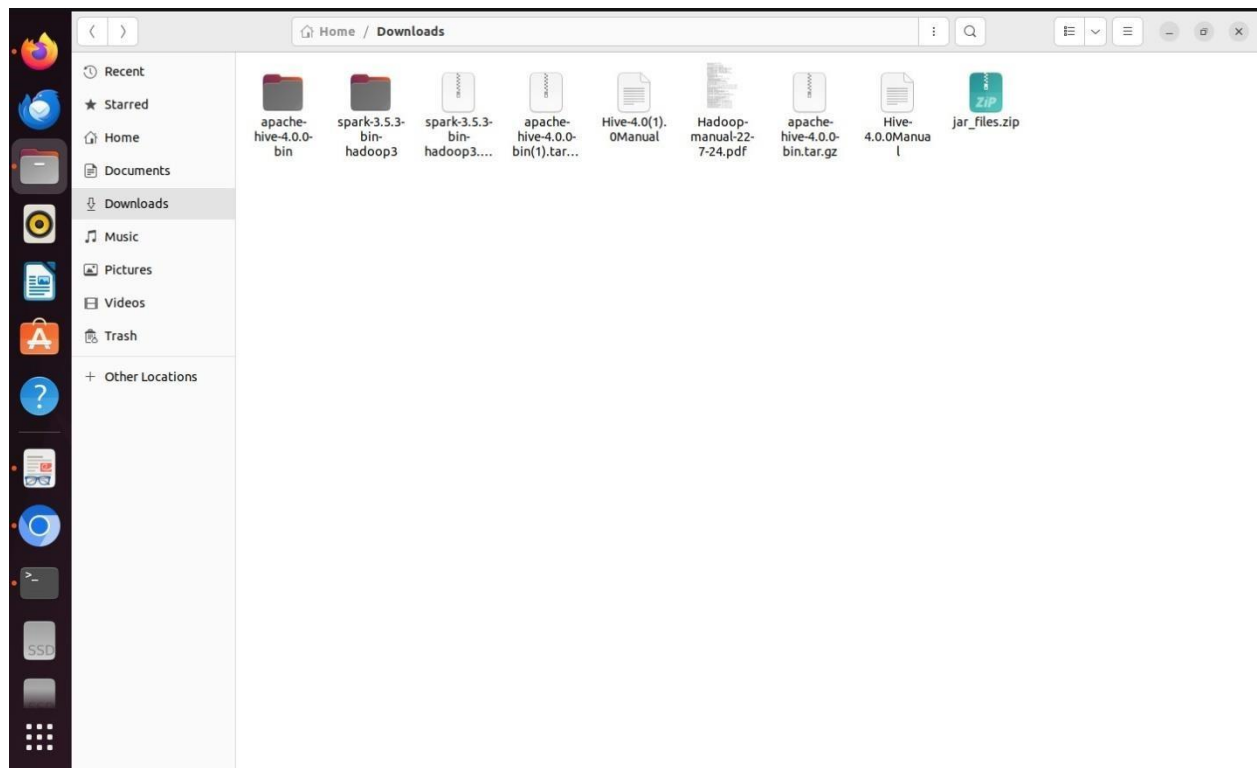
4. Here to check the installation of apache spark

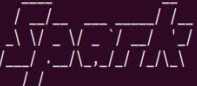
```
Open ▾ [🔍] .bashrc /home/hduser Save [≡] [⏪] [⏩] [X]

90 # some more ls aliases
91 alias ll='ls -lF'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands. Use like so:
96 # sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${S? = 0 }" && echo terminal || echo error)' "${history|tail -n1|sed -e '\`s/\s*[0-9]\+\s*//;s/[[:&]]\`s*alert$//'\`}'"
98
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 #HADOOP VARIABLES START
120 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
121 export HADOOP_HOME=/usr/local/hadoop
122 export PATH=$PATH:$HADOOP_HOME/bin
123 export PATH=$PATH:$HADOOP_HOME/sbin
124 export HADOOP_MAPRED_HOME=$HADOOP_HOME
125 export HADOOP_COMMON_HOME=$HADOOP_HOME
126 export HADOOP_YARN_HOME=$HADOOP_HOME
127 export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
128 export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
129 export SPARK_HOME=/home/hduser/Downloads/spark-3.5.3-bin-hadoop3
130 export PATH=$PATH:$SPARK_HOME/bin
131 export PATH=$PATH:$SPARK_HOME/sbin
132 #HADOOP VARIABLES END
```

```
hduser@bv-ThinkCentre-neo-50t-Gen-3: ~ [🔍] [≡] [⏪] [⏩] [X]
GNU nano 6.2 /home/hduser/.bashrc
if ! shopt -oq posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```



```
hduser@bv-ThinkCentre-neo-50t-Gen-3: ~  
hduser@bv-ThinkCentre-neo-50t-Gen-3:~$ nano ~/.bashrc  
hduser@bv-ThinkCentre-neo-50t-Gen-3:~$ source ~/.bashrc  
hduser@bv-ThinkCentre-neo-50t-Gen-3:~$ spark-shell  
24/09/28 12:46:49 WARN Utils: Your hostname, bv-ThinkCentre-neo-50t-Gen-3 resolves to a loopback address: 127.0.1.1; using 10.10.11.213 instead (on interface eno1)  
24/09/28 12:46:49 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
24/09/28 12:46:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Spark context Web UI available at http://10.10.11.213:4040  
Spark context available as 'sc' (master = local[*], app id = local-1727507812556).  
Spark session available as 'spark'.  
Welcome to  
 version 3.5.3  
Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.24)  
Type in expressions to have them evaluated.  
Type :help for more information.
```

5. To run the basic commands


```
ies Terminal Sep 28 13:11
hduser@bv-ThinkCentre-neo-50t-Gen-3: ~

Type :help for more information.

scala> val data = seq(1,2,3,4,5)
<console>:22: error: not found: value seq
    val data = seq(1,2,3,4,5)
                ^

scala> val data = Seq(1,2,3,4,5)
<console>:22: error: not found: value seq
    val data = seq(1,2,3,4,5)
                ^

scala> val data = Seq(1,2,3,4,5)
data: Seq[Int] = List(1, 2, 3, 4, 5)

scala> val rdd = sc.parallelize(data)
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> rdd.collect()
res0: Array[Int] = Array(1, 2, 3, 4, 5)

scala> res0:Array[Int]=Array(1,2,3,4,5)
<console>:1: error: ';' expected but '=' found.
    res0:Array[Int]=Array(1,2,3,4,5)
                ^

scala> val evenNumber = rdd.filter(_%2==0)
evenNumber: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:23

scala> evenNumber.collect()
res1: Array[Int] = Array(2, 4)

scala> val rdd = sc.parallelize(seq(1,2,3,4,5))
<console>:23: error: not found: value seq
Error occurred in an application involving default arguments.
    val rdd = sc.parallelize(seq(1,2,3,4,5))
                              ^

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[2] at parallelize at <console>:23

scala> val sum = rdd.reduce(_+_ )
sum: Int = 15

scala>
```

```
ies Terminal Sep 28 13:13
hduser@bv-ThinkCentre-neo-50t-Gen-3: ~

scala> val data = seq(1,2,3,4,5)
<console>:22: error: not found: value seq
    val data = seq(1,2,3,4,5)
                ^

scala> val data = Seq(1,2,3,4,5)
data: Seq[Int] = List(1, 2, 3, 4, 5)

scala> val rdd = sc.parallelize(data)
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> rdd.collect()
res0: Array[Int] = Array(1, 2, 3, 4, 5)

scala> res0:Array[Int]=Array(1,2,3,4,5)
<console>:1: error: ';' expected but '=' found.
    res0:Array[Int]=Array(1,2,3,4,5)
                ^

scala> val evenNumber = rdd.filter(_%2==0)
evenNumber: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:23

scala> evenNumber.collect()
res1: Array[Int] = Array(2, 4)

scala> val rdd = sc.parallelize(seq(1,2,3,4,5))
<console>:23: error: not found: value seq
Error occurred in an application involving default arguments.
    val rdd = sc.parallelize(seq(1,2,3,4,5))
                              ^

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[2] at parallelize at <console>:23

scala> val sum = rdd.reduce(_+_ )
sum: Int = 15

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[3] at parallelize at <console>:23

scala> val divide = rdd.reduce(_/_ )
divide: Int = 0

scala>
```

```
ies Terminal Sep 28 13:16 hduser@bv-ThinkCentre-neo-50t-Gen-3: ~
scala> rdd.collect()
res0: Array[Int] = Array(1, 2, 3, 4, 5)

scala> res0:Array[Int]=Array(1,2,3,4,5)
<console>:1: error: ';' expected but '=' found.
    res0:Array[Int]=Array(1,2,3,4,5)
           ^

scala> val evenNumber = rdd.filter(_%2==0)
evenNumber: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:23

scala> evenNumber.collect()
res1: Array[Int] = Array(2, 4)

scala> val rdd = sc.parallelize(seq(1,2,3,4,5))
<console>:23: error: not found: value seq
Error occurred in an application involving default arguments.
    val rdd = sc.parallelize(seq(1,2,3,4,5))
                           ^

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[2] at parallelize at <console>:23

scala> val sum = rdd.reduce(_+_ )
sum: Int = 15

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[3] at parallelize at <console>:23

scala> val divide =rdd.reduce(_/_ )
divide: Int = 0

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[4] at parallelize at <console>:23

scala> val subtract =rdd.reduce(_-_ )
subtract: Int = -13

scala> val rdd = sc.parallelize(Seq(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[5] at parallelize at <console>:23

scala> val mul =rdd.reduce(_*_ )
mul: Int = 120

scala> 
```