We need to set up development environment for Python and Spark

1. Python
2. IDE such as Pycharm
3. Spark
4. Setting up of WinUtils
5. Integration of Pycharm and Spark

Prerequsites::

RAM : Atleast 4 GB RAM/8 GB highly desired

Operating system version : 64 bit

Set up Python on windows 10

* **How to Install?**
  + Launch Google chrome and search for Python
  + From [www.python.org](http://www.python.org/downloads) you can find the Downloads link.
  + Click on the link to launch the download page
* As we are going to work with Spark, we need to choose the compatible version for Spark
* For our environment, the Spark version we are using is 1.6.3
* **Spark version 1.6.3**and **Spark 2.x**are compatible with **Python 2.7**
* Make sure you choose Python 2.7.14 for download and click on the link
* **.msi** version will be downloaded(Microsoft Installer)
* Double click on the file and progress for further steps
  + You can choose ‘install for all users’, click on next
  + Path can be changed if you want to change. click on next
  + click on next and click on ‘yes’ at User Account Control window.
  + Setup will be automatically progressed and the setup is done.
  + Click on Finish button
  + Restart the System once it’s finished.
* After restarting the System search for Python and click on it
* It will launch the Python console.

**Python Environment Variables**

* **PYTHONPATH** – It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.
* **PYTHONSTARTUP** – contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH. .pythonrc.py is similar to .bashrc
* **PYTHONCASEOK** – is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.
* **PYTHONHOME** – alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

**Pycharm**

## Using IDE

For this course, we will be using pycharm.  
You can download IDE for the following [link 1](https://www.jetbrains.com/pycharm/).

* Select appropriate OS and download.
* Complete installation process.
* Launch pycharm

**Pass run time arguments**

As part of this topic, we will know how to pass the arguments from a program.

import sys print("Hello " + sys.argv[1] + " from " + sys.argv[0])

To pass the arguments click on run -> Edit Configuration. Here pass the argument.  
Once we updated the runtime parameters we can run the program.

**Installation of spark**

* o to the link [Spark Download 13](https://spark.apache.org/downloads.html)
* Choose the relevant version of Spark(this time it is 1.6.3)
* Choose the Hadoop version(this time it is 2.6)
* Click on the download link which will take to the downloads page
* Copy the link and paste it in the Cygwin
* wget [http://www-us.apache.org/dist/spark/spark-1.6.3/spark-1.6.3-bin-hadoop2.6.tgz 8](http://www-us.apache.org/dist/spark/spark-1.6.3/spark-1.6.3-bin-hadoop2.6.tgz)
* Copy the Spark folder from Cygwin to Local Directory
* Download winutils from [here 2](http://www.eaiesb.com/blogs/?tag=apache-spark-installation-on-windows) , save the link into your local machine by creating a hadoop/bin folder
* Set the path in the system variables
* Change the permissions if still the spark-shell is not launched
* Set the Spark path in system variables

As part of this topic, we will set up spark on our local machine

* Copy the Spark folder to Local Directory
* Set the Spark path in the system variables
* Change the permissions if still the pyspark is not launched

**JAVA SET UP**

* Before getting started check whether Java and JDK are installed or not
  + Launch command prompt – Go to the search bar on the windows laptop, type **cmd** and hit enter
  + Type java -version If it returns version, check whether 1.8 or not. It is better to have 1.8 version. If you have another version, consider uninstall and install 1.8 (Search for programs installed and uninstall Java)
  + Type javac -version If it returns version, check whether 1.8 or not. It is better to have 1.8 version. If you have another version, consider uninstall and install 1.8 (Search for programs installed and uninstall Java)
  + If you need other versions, make sure environment variables point to 1.8
  + If you do not have Java at all, make sure to follow the instructions and install 1.8 version of JRE and JDK.
* **Why do we need to install Java and JDK?** Scala, Spark, and many other technologies require Java and JDK to develop and build the applications. Scala is JVM based programming language.
* **How to install Java and JDK?**
  + Go to [official page of Oracle 2](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) where downloads are available
  + Accept the terms and download the 64-bit version
* **How to validate?**
  + Use java -version and javac -version commands in command prompt and see they return 1.8 or not

## Setup WinUtils to get HDFS APIs working

* Not required for Mac or Linux
* **Why install winutils?**
  + In the process of building data processing applications using Spark, we need to read data from files
  + Spark uses HDFS API to read files from several file systems like HDFS, s3, local etc
  + For HDFS APIs to work on Windows, we need to have WinUtils
* **How to install winutils?**
  + Click [here 6](https://codeload.github.com/gvreddy1210/64bit/zip/master) to download 64 bit winutils.exe
  + Create a directory structure like this C:/hadoop/bin
  + Setup new environment variable HADOOP\_HOME
    - Search for **Environment Variables** on Windows search bar
    - Click on **Add Environment Variables**
    - There will be 2 categories of environment variables
      * User Variables on top
      * **System Variables** on **bottom**
      * Make sure to click on Add for **System Variables**
      * Name: HADOOP\_HOME
      * Value: **C:\hadoop** (don’t include bin)
    - Also, choose **Path** and **click on Edit**
      * Click on Add
      * Add new entry **%HADOOP\_HOME%\bin**

**Integration of spark with Pycharm**

**Open Pycharm Ide and select project.**

**Then go to settings ( from file option), then in project name select python interpreter select python version**

**Also you need to integrate spark by choosing setting and select project structure and choose python folder and py4j file from spark folder.**

rom pyspark import SparkConf, SparkContext  
sc = SparkContext(master="local",appName="Spark Demo")  
print(sc.textFile("F:\spark\deckofcards.txt.txt").first())

**we need to import few objects like SparkConf and SparkContext**

**then create variable for object and define its paramaters**

### Install Cygwin

* Go to the link provided [here 5](https://www.cygwin.com/install.html)
* Download and Install Cygwin
* Install packages like ssh, inetutils
* Launch Cygwin.

### Connect to Labs Using Cygwin

* To connect to the labs in Cygwin we must have ssh package pre-installed.
* use the command **ssh username@hostname**
* Copy the password from labs n use **shift+insert** to paste the password in Cygwin.

### Password Less Login in Cygwin

* Use **ssh-keygen** which will generate necessary keys for passwordless login
* Once the key is generated copy the id using the command **ssh-copy-id username@hostname** and hit enter
* Now we have successfully set up the passwordless login using Cygwin.

### Copy a File From Local Machine to Remote Machine Using Cygwin

* To copy a file to the remote machine we use below command

scp path/fileName.Format username@hostname valid path in the remote machine

**HDFS**

As part of this topic, we will see the overview of HDFS

* HDFS – Hadoop Distributed file system
* Command line hadoop fs or hdfs dfs
* Properties files
  + /etc/hadoop/conf/core-site.xml
  + /etc/hadoop/conf/hdfs-site.xml
* Important Properties
  + fs.defaultFS
  + dfs.blocksize
  + dfs.replication
* HDFS commands
  + Copying Files
    - From local file system(hadoop fs -copyFromLocal or -put)
    - To local file system (hadoop fs -copyToLocal or -get)
    - From one HDFS location to other (hadoop fs -cp)
  + Listing files (hadoop fs -ls)
  + Previewing data from files (hadoop fs -tail or -cat)
  + Checking sizes of all files (hadoop fs -du).

**Properties Files**

* From gateway node, we can go to the location /etc/hadoop/conf
* Here we can see properties files which control the environment of HDFS, YARN etc
* We have a bunch of files in this location but our primary focus is on only two files core-site.xml and hdfs-site.xml
* In core-site.xml we can see the information about our cluster
* In hdfs-site.xml we can see the important properties file such as block size and replication

**Copy Files From Local Machine to HDFS**

* To copy files from local machine to HDFS use below command

hadoop fs -copyFromLocal /data/crime /user/username/.

* Go to [https://github.com/dgadiraju/data 93](https://github.com/dgadiraju/data)
* Clone or Download on to Virtual Machines created using Cloudera Quickstart or Hortonworks Sandbox
* You can setup locally for practicing for Spark, but it is highly recommended to use HDFS which comes out of the box with Cloudera Quickstart or Hortonworks or our labs
* On lab, they are already available
* retail\_db
  + Master tables
    - customers
    - products
    - categories
    - departments
  + Transaction tables
    - orders
    - order\_items
* To get the number of records from the file we can use below command **wc -l /data/retail\_db//**