**Practica N° 02**

**Indicaciones:**

* Para las preguntas del tipo taller, ingrese a su carpeta de trabajo del Jupyter (asignada a usted), y cree un archivo llamado practica\_02\_[nombre\_apellido], en esta sección desarrolle las preguntas.
* El servidor estará disponible el domingo (3:00 pm a 7:00 pm) para que puedan desarrollar sus prácticas (cualquier consulta me la hacen saber por el whatsup).
* La Fecha de entrega de la práctica es como máximo el lunes 03/09 hasta las (11:59 pm).

**Nombre Estudiante:**

1. Cuáles son las características más relevantes de Python, detalle.

INTERPRETADO : No necesita un compilador para pasar su sintaxis a codigo maquina, con lo cual sacrifica velocidad pero gana en no depender del SO (MULTIPLATAFORMA) y poder facilitar un interprete interactivo.

SIMPLICIDAD : Posee una sintaxis, filosofia y tecnicas que apuntan a una sintaxis limpia, practica y facil de comprender. Detallando por ejemplo tenemos:

Para sintaxis TIPADO DINAMICO, con lo cual puedes manejar variables sin especificar su tipo.

Para filosofia El ZEN DE PYTHON, son como los mandamientos que debe leer todo pythonero tan solo codeando “import this”

Para tecnica PEP-8, Conjunto de reglas que nos llevaran a presentar un codigo mas legible y ordenado.

CODIGO LIBRE: Es codigo es 100% libre y tiene detras una gran comunidad que constantemente esta enfocada en desarrollar mejoras, solucionar errores e indagar e nuevas areas.

1. Indique la opción correcta en la creación de una tupla
   1. tuplaA<-{2,3,4,5}
   2. tuplaA=(2,3,4,5}
   3. tuplaA=(2,3,4,5)
   4. tuplaA=[2,3,4,5]
   5. a y d

RESPUESTA : c.

1. Se tiene la siguiente lista:

Lista\_datos=[12,3,2,34]

A=(Lista\_datos[-2]+4)\*\*2

B= (Lista\_datos[2]+4)\*\*2

Cuál es el resultado de la operación:

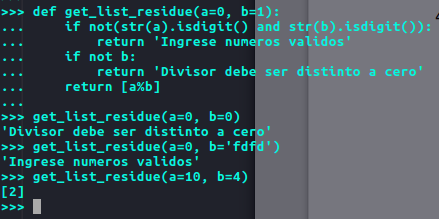
print("A=",A,"B=",B,"Revisar",A==B)

* 1. ('A=', 36, 'B=', 16, 'Revisar', False)
  2. ('A=', 16, 'B=', 16, 'Revisar', False)
  3. ('A=', 64, 'B=', 64, 'Revisar', True)
  4. ('A=', 36, 'B=', 36, 'Revisar', True)
  5. Ninguna de las anteriores

RESPUESTA: d.

1. Cree una función la cual tenga los siguientes parámetros (ver viñetas en la parte inferior), el nombre de la función se lo dejo a su criterio:

* Input: entero a, entero b
* Output: devuelve una lista que contiene el residuo de a y b.



1. Elementos básicos que conforman el Spark
   1. Driver Manager, DAG y Cluster Manager.
   2. Executors, Driver Manager y las particiones.
   3. Executors, Driver Manager y YARN.
   4. Cluster Manager, Executor y el Driver Process.
   5. Ninguna de las anteriores.

RESPUESTA d.

1. Que elemento de SPARK se encarga de la sesión SPARK
   1. Driver Manager.
   2. Mesos
   3. Hive
   4. Cluster Manager
   5. Ninguna de las anteriores.

RESPUESTA: e.

1. Diferencias entre una transformación y una acción, detalle:

RESPUESTA: Veamos, spark almacena los datos que lee de otras fuentes en objetos RDD o Dataset, ahora cuando realizamos un operación que implique crear un nuevo objeto basado en el original, estaremos hablando de una TRANSFORMACION, donde por la canitda de proceso Spark lo realiza de manera lazy realizando para ello un plan de ejecucuion en espera a ser invocado.

Mientras que cuando simplemente extraemos ciertos resultados del RDD sin alterarlo, diremos que hemos realizado una ACCION.

1. Que librerías me permite encontrar la ruta de donde se encuentra el spark
   1. findspark
   2. pyspark
   3. numpy
   4. pandas
   5. c y d

RESPUESTA b.

1. Que librerías me permite usar el spark SQL
   1. findspark
   2. pyspark
   3. numpy
   4. pandas
   5. c y d

RESPUESTA : d.

1. Que es el optimizer catalyst, detalle:

Secuencia de fases por las cuales internamente SPARK SQL optimiza nuestras operaciones para lograr un mejor rendiemiento, logrando como objetivo un plan de ejecución en modo lazy.

1. Que objetos o abstracciones utiliza el SPARK para la creación de objetos, detalle:

Las abstracciones usadas son RDD, DATAFRAME Y DATASET, donde cada una presenta un ventaja sobre la otra dependiendo la situación, en lo personal lo mas atractivo al uso de pyhton es Dataframe, el cual se puede entender como una matriz de filas y columnas que además almacena la información de cada columna, la igual que el RDD se aloja en diferentes maquinas convirtiéndose internamente como una red de datos distribuidos según su tamaño.

APARTIR EN ADELANTE LOS EJERCICIOS SE REALIZARON

En la carpeta: /practica\_2

1. Ingestar archivo creditobanco.csv al HDFS utilizando el comando PUT **(TALLER)**

Ruta Servidor Local: /home/jupyter/[carpeta\_estudiante]/creditobanco.csv

Ruta HDFS Destino: /user/[carpeta\_estudiante]/data\_in/credito\_banco

El archivo se encuentra adjunto al correo.

1. Ingestar conexión a BD “select \* from personas\_reniev” al HDFS (la ruta de la conexión se encuentra en las clases de ingesta 6) utilizando Spark **(TALLER)**

Ruta HDFS Destino: /user/[carpeta\_estudiante]/dat\_in/personas\_reniec

1. Cree dos spark dataframes abriendo los archivos ingestados en la pregunta 12 y 13 **(TALLER)**
2. Se desea los dataframes de la pregunta 14, realice las siguientes: **(TALLER).**

* Cuente la cantidad de registros de cada dataframe
* Revise los nombres de las columnas
* Revise los tipos de columnas de cada dataframe
* Vea que tipo de objeto son los dos dataframes creados

1. Crear un nuevo dataframe Df3, el cual una (left join) los dos dataframe Df1 y Df2 (la llave para unir los dos dataframes es el NUMERODOCUMENTO), el nuevo dataframe contendrá todas las columnas de los dos dataframes (cuidado con las columnas duplicadas), puede utilizar cualquier tipo de sintaxis Dataframe o SQL **(TALLER):**
2. Con el dataframe creado en el pregunta 17 ingestar el resultado en la siguiente ruta **(TALLER):**

Ruta HDFS Destino: /user/[carpeta\_estudiante]/data\_out/resultado

El archivo debe de ser almacenado como parquet.