ESCUELA PROFESIONAL DE INGENIERÍA DE TELECOMUNICACIONES

LABORATORIO DE: COMPUTACIÓN 2

EXPERIENCIA N°: 05

TÍTULO DE LA EXPERIENCIA:

IMPORTAR FICHEROS EXCEL CON PANDAS

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Alumno(os): | | | Grupal | Indiv. | Total |  |  |  |
| 1.Prieto Tito Manuel Ismael | | |  | x |  |  |  |  |
| 2. | | |  |  |  |  |  |  |
| 3. | | |  |  |  |  |  |  |
| 4. | | |  |  |  |  |  |  |
| Grupo: | b | Docente: Ing William Mullisaca Atamari | | | |  |  |  |
| Semestre: | 3 |  |  |  |
| Fecha de entrega: | 15-06-22 | Hora: | 12.00 | | |  |  |  |

LABORATORIO N° 05

IMPORTAR FICHEROS EXCEL CON PANDAS

i.- ObjetivoS:

-. Que el estudiante sea capaz de importar ficheros Excel con pandas en Python.

II.- FUNDAMENTO TEÓRICO:

2.1. Programando con python como lenguaje de programación hay muchas librerías muy útiles para abrir excel: [Openpyxl](https://openpyxl.readthedocs.io/en/default/), [XlsxWriter](http://xlsxwriter.readthedocs.io/), y [Pandas](https://pandas.pydata.org/)

En esta guía utilizaremos Pandas debido a su flexibilidad y cantidad de herramientas. Mi recomendación es: utilizar Pandas si van a trabajar con muchos datos (porque tiene muchas herramientas) y aprovechar las otras librerías si quieren hacer desarrollo y solo quieren guardar datos en un archivo excel.

Para esta guia utilizaremos un set de datos : [Austin Weather](https://www.kaggle.com/grubenm/austin-weather)

Primero importaremos las librerías necesarias

import pandas as pd

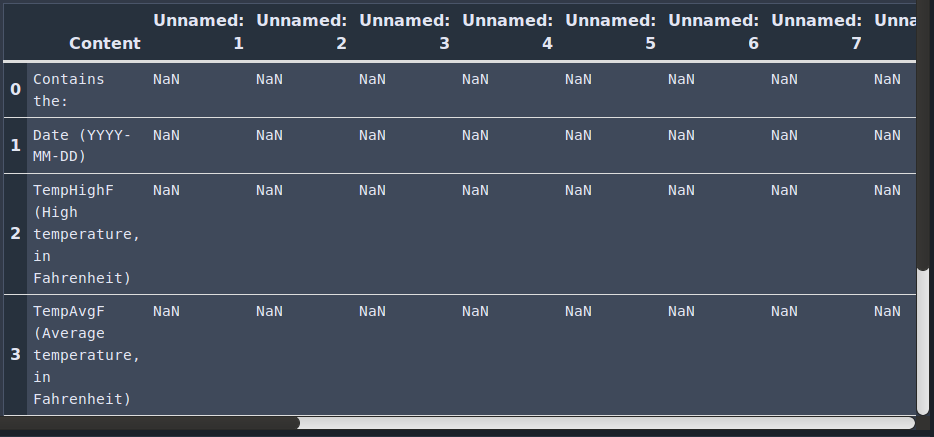
Los archivos están guardados en la misma carpeta que el dónde estoy ejecutando el codigo, por eso no es necesario colocar carpetas o subcarpetas en la dirección.

ipath = 'austion\_weather.xlsx'

df = pd.read\_excel(ipath)

df.head()

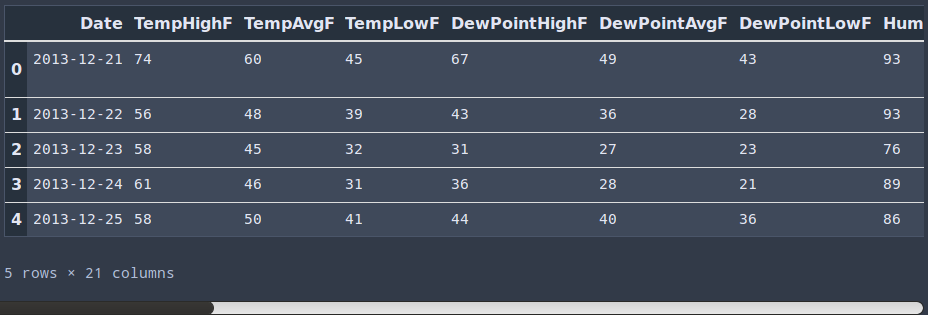
Si ejecutan esto les debería mostrar las primeras 5 filas del archivo excel. Está bastante feo porque solo la primera columna tiene datos.



Vamos a saltarnos unas cuantas filas para encontrar los datos.

df = pd.read\_excel(ipath, skiprows=26)

df.head()



¿Eso está mejor verdad? Ahora pandas me muestra los títulos de la columna como ser: Date, TempAvgF, TempLowF, etc

Ahora veamos algo que no sé ve a simple vista. ¿Cómo son los datos que tenemos? Para eso ejecutamos:

df.info()

Esto nos debería mostrar los detalles por cada columna:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1319 entries, 0 to 1318

Data columns (total 21 columns):

Date 1319 non-null object

TempHighF 1319 non-null int64

TempAvgF 1319 non-null int64

TempLowF 1319 non-null int64

DewPointHighF 1319 non-null object

DewPointAvgF 1319 non-null object

DewPointLowF 1319 non-null object

HumidityHighPercent 1319 non-null object

HumidityAvgPercent 1319 non-null object

HumidityLowPercent 1319 non-null object

SeaLevelPressureHighInches 1319 non-null object

SeaLevelPressureAvgInches 1319 non-null object

SeaLevelPressureLowInches 1319 non-null object

VisibilityHighMiles 1319 non-null object

VisibilityAvgMiles 1319 non-null object

VisibilityLowMiles 1319 non-null object

WindHighMPH 1319 non-null object

WindAvgMPH 1319 non-null object

WindGustMPH 1319 non-null object

PrecipitationSumInches 1319 non-null object

Events 1319 non-null object

dtypes: int64(3), object(18)

memory usage: 216.5+ KB

¿Qué leemos? Resumiendo: cantidad de datos por columna, qué tipo de datos es (object, int64) Object es una mezcla de datos (texto, números, etc. Pandas llama así a las columnas con datos mixtos) y el int64, se refiere a números enteros.

Hay más información, pero por ahora esto es todo lo que nos interesa.

Una tabla como está nos puede dar problemas para trabajar. ¿Por qué? Porque por ejemplo no puedo sacar datos de estadísticos con columnas que tienen datos de texto y números. No puedo sacar promedios si Pandas no reconoce a los datos como números.

¿Qué ocurre aquí? Por qué Pandas me dice que la columna de DewPointAvgF tiene texto y números. Si revisan esa columna verán que cuando no hay datos no dejan la celda vacía, colocan un guión (-), un ejemplo es el 14-06-2014 o el 10-08-2015. Esto hace que Pandas crea que hay texto y número... y tiene razón!!!

Vamos a avisarle a Pandas que los guiones significan datos inexistentes. Eso se llama NaN Values, y Pandas si reconoce esos datos, los considera datos inexistentes.

df = pd.read\_excel(ipath, skiprows=26, na\_values='-')

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1319 entries, 0 to 1318

Data columns (total 21 columns):

Date 1319 non-null object

TempHighF 1319 non-null int64

TempAvgF 1319 non-null int64

TempLowF 1319 non-null int64

DewPointHighF 1312 non-null float64

DewPointAvgF 1312 non-null float64

DewPointLowF 1312 non-null float64

HumidityHighPercent 1317 non-null float64

HumidityAvgPercent 1317 non-null float64

HumidityLowPercent 1317 non-null float64

SeaLevelPressureHighInches 1316 non-null float64

SeaLevelPressureAvgInches 1316 non-null float64

SeaLevelPressureLowInches 1316 non-null float64

VisibilityHighMiles 1307 non-null float64

VisibilityAvgMiles 1307 non-null float64

VisibilityLowMiles 1307 non-null float64

WindHighMPH 1317 non-null float64

WindAvgMPH 1317 non-null float64

WindGustMPH 1315 non-null float64

PrecipitationSumInches 1319 non-null object

Events 1319 non-null object

dtypes: float64(15), int64(3), object(3)

memory usage: 216.5+ KB

Cambio, ¿verdad? Ahora Pandas me da nuevos tipos de datos. Los float64, que significan que son datos con comas, o sea, decimales.

Las últimas dos columnas continuan siendo object, debido a que PrecipitationSumInches tiene un valor de trazas (T). Cada vez que precipito/llovió menos de 0.01 inches (pulgadas) colocaron T en la celda. No es un dato inexistente, y debe ser tratado de una forma diferente a la anterior.

La columna Events tiene descripciones de los eventos, podemos dejarlo en texto.

Bien... En proximos tutoriales veremos cómo convertir la columna Date en tipo DateTime, esto nos permitira trabajar estos datos como una serie de tiempo.

Cualquier duda pueden dejarla en los comentarios, o sino buscarme en las redes sociales.

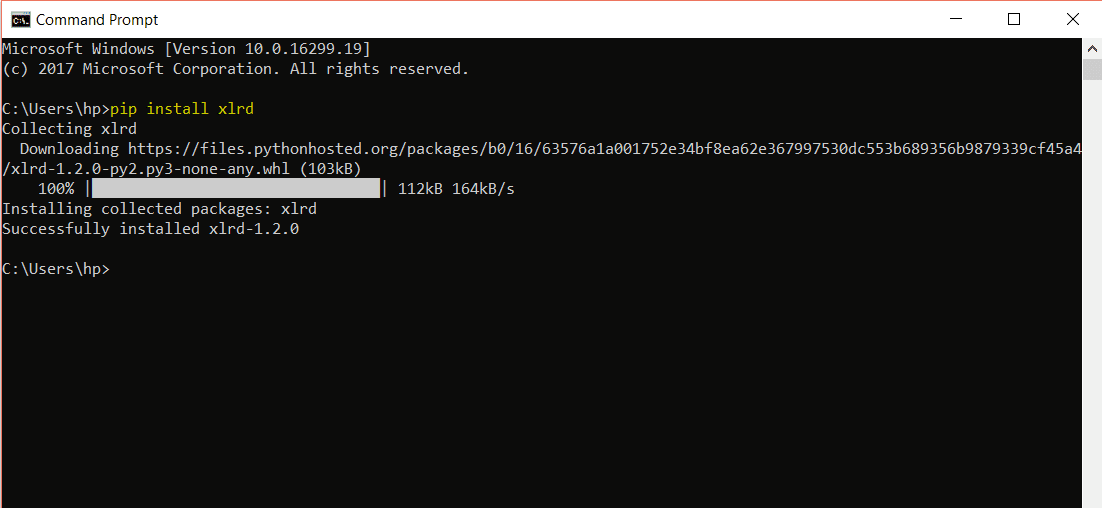
Nota: Las imagenes son del Jupyter Notebook. Más adelante hablaré de esa herramienta, pero si ejecutan los codigos en algun IDE (Spyder, Pycharm, etc.) o de la consola interactiva de Python no deberían tener problemas en obtener los mismos resultados.

## **2.2. LEER UN ARCHIVO DE EXCEL**

Puedes leer desde un archivo de Excel usando el método read\_excel () de pandas. Para esto, necesitas importar un módulo más llamado xlrd.

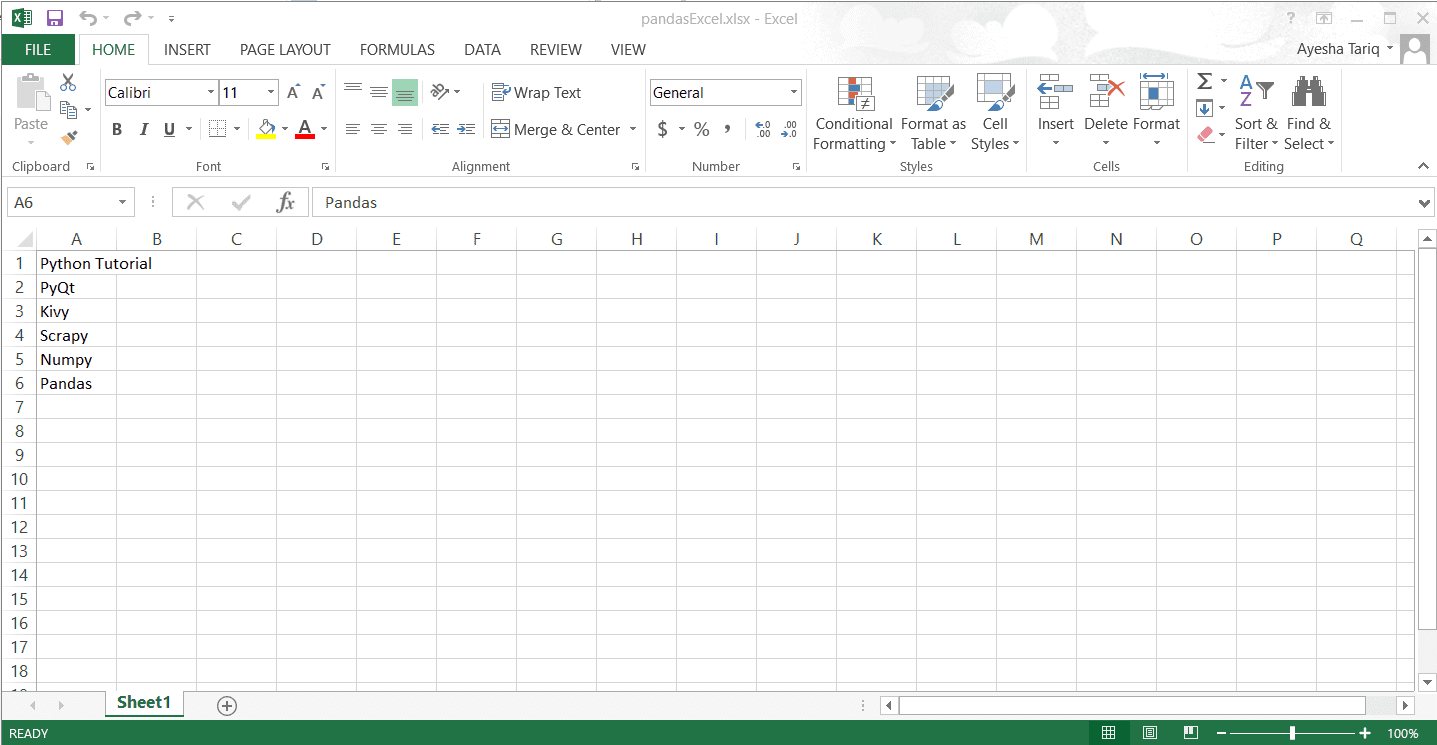
Instala xlrd usando pip:

pip install xlrd



El siguiente ejemplo muestra cómo leer de una hoja de Excel:

1. Creamos una hoja de Excel con los siguientes contenidos:



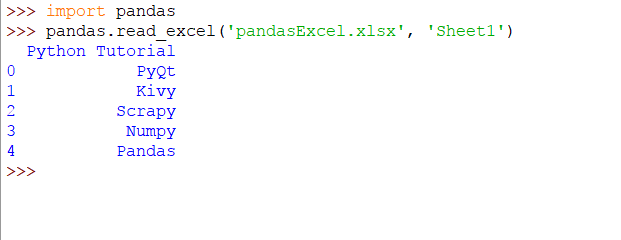
1. Importa el módulo de pandas.

import pandas

1. Pasaremos el nombre del archivo de Excel y el número de hoja del que necesitamos leer los datos al método read\_excel ().

pandas.read\_excel('pandasExcel.xlsx', 'Sheet1')

El fragmento anterior generará el siguiente resultado:



Si verificas el tipo de salida usando la palabra clave de type, te dará el siguiente resultado:

<class 'pandas.core.frame.DataFrame'>

Este resultado es llamado **DataFrame**. Esa es la unidad básica de pandas con la que se va a tratar mas adelante.

El DataFrame es una estructura de 2 dimensiones etiquetada donde podemos almacenar datos de diferentes tipos. DataFrame es similar a una tabla SQL o una hoja de cálculo de Excel.

## **2.3. IMPORTAR ARCHIVO CSV**

Para leer un archivo CSV, puedes usar el método read\_csv () de pandas.

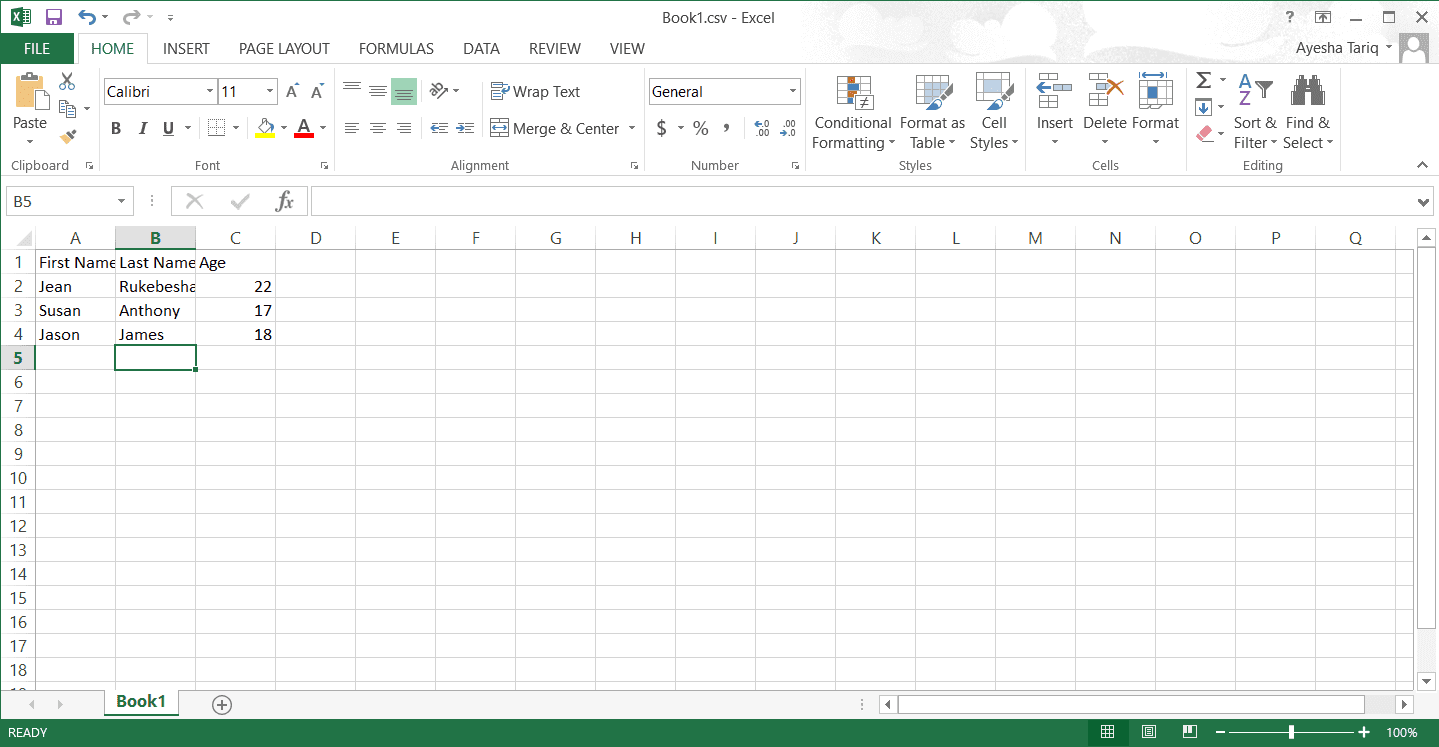
Importa el módulo de pandas:

import pandas

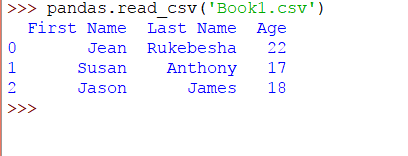
Ahora llama al método read\_csv () de la siguiente manera:

pandas.read\_csv('Book1.csv')

Book1.csv tiene el siguiente contenido:



El código generará el siguiente DataFrame:



III.- RECOMENDACIONES EN SEGURIDAD

3.1. En condiciones de una emergencia Identifique:

* Vías de acceso y evacuación
* Equipos de respuesta a emergencias
* Señalización de seguridad

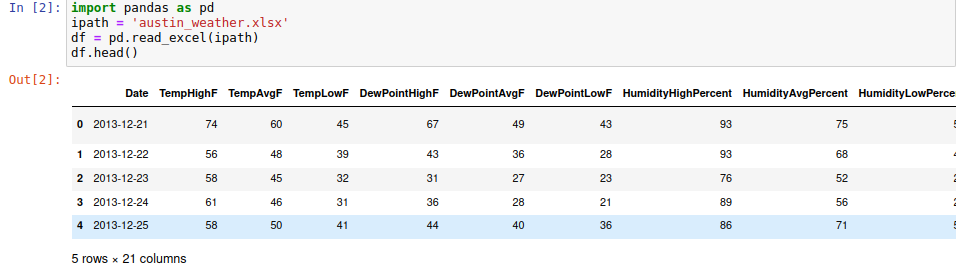
3.2. Complete el ATS (Anexo 1) y cumpla las condiciones obligatorias para el uso del ambiente

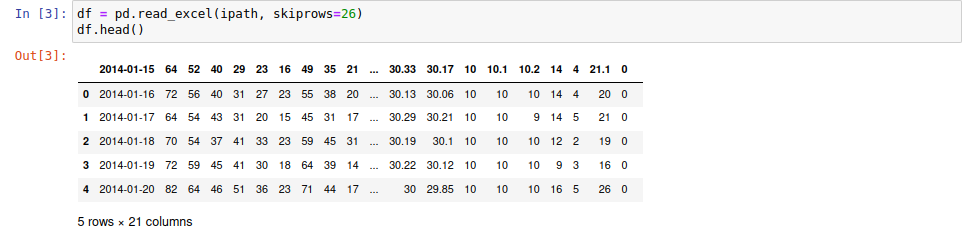
IV.- EQUIPOS Y MATERIALES A UTILIZAR:

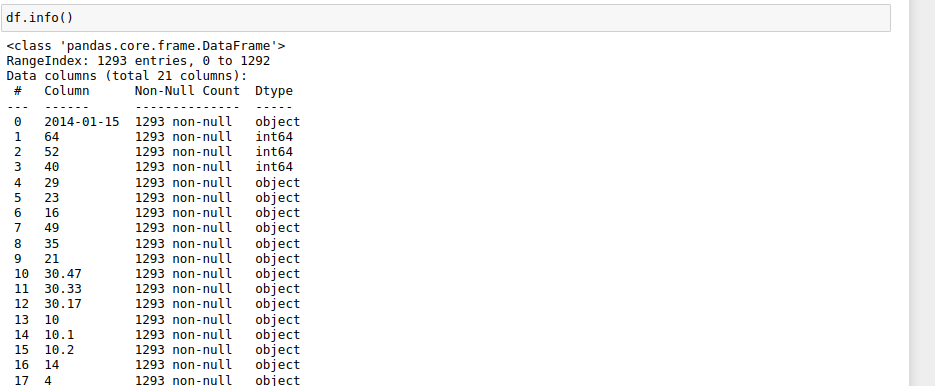
|  |  |
| --- | --- |
| Cantidad | Descripción |
| 1 | PC con software Python |

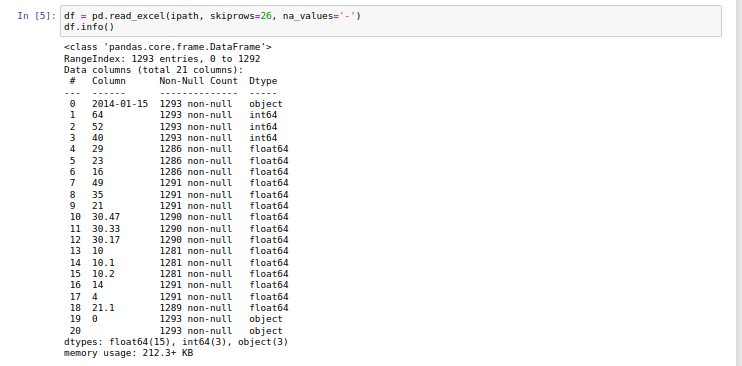
V.- PROCEDIMIENTO:

5.1. Ejecutar, el ítem 2.1 con el archivo Excel que te da el docente.



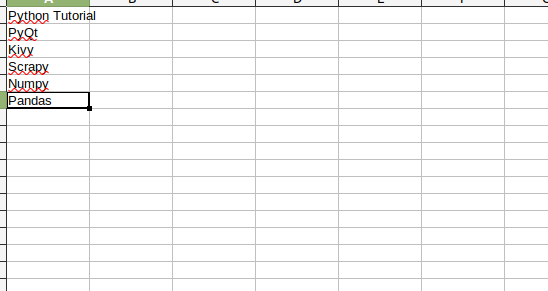


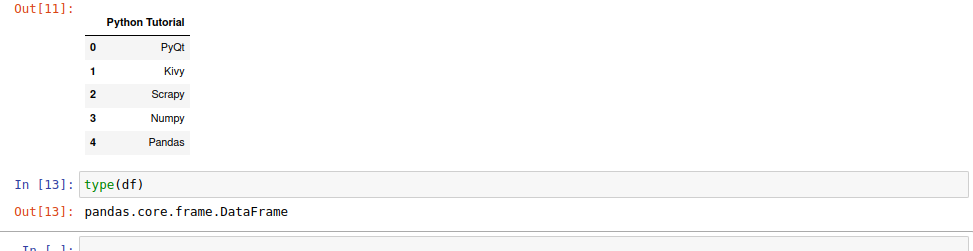




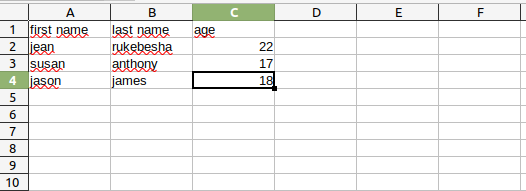
VI.- EJERCICIO:

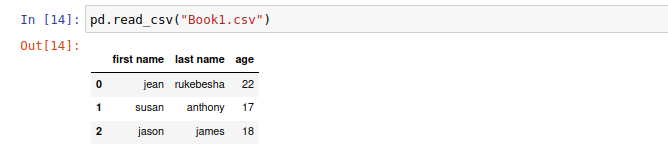
6.1. ejecutar el ítem 2.2, para ello debe crear un archivo 'pandasExcel.xlsx’





6.2. ejecutar el ítem 2.3, para ello debe crear un archivo 'Book1.csv’





VII.- BIBLIOGRAFÍA:

* Eugenia Bahit Curso: Python para Principiantes [www.eugeniabahit.com](http://www.eugeniabahit.com/)
* <https://www.youtube.com/watch?v=_8onVOY2j4E>

Rubrica:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Puntualidad | Asistencia | Desarrollo de experiencias | | | | | | | | | | Observaciones Conclusiones | Bibliografía | Sugerencias | Total | Porcentaje % |
| x |  | 5.1 | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 | 6.1 | 6.2 | 6.3 |  |  |  |  |  |
| 2.0 | 2.0 | 4.0 | - | - | - | - | - | - | 5.0 | 5.0 | - | - | 1.0 | 1.0 | 20 | 15 |

Ing William Mullisaca

Docente DAIE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **Anexo 1**  **ATS: Análisis de trabajo seguro** | | | | | | | | | Fecha | 30/04/2020 |  |
| Versión | 1.1 |
| Código | ATS20V1 |
| Curso: |  | | | Tarea: |  | | | Docente: | | |  | | | |
| Ambiente: | |  | | | Grupo: |  | | Mesa: | | |  | | Fecha: |  |
| Integrantes (Apellidos y nombres) | | | | | | Firma | Integrantes (Apellidos y nombres) | | | | | | | Firma |
| 1. | | | | | |  | 2. | | | | | | |  |
| 3. | | | | | |  | 4. | | | | | | |  |
| Elementos de protección (Marque con aspa) | | | | | | Características de elementos de protección, equipos y herramientas | | | | | | | | |
|  |  |  |  |  |  | 1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | | | | |
| USO OBLIGATORIO DEL CASCO DE SEGURIDAD | USO OBLIGATORIO DE BOTAS AISLANTES | USO OBLIGATORIO DE GUANTES AISLANTES | USO OBLIGATORIO DE PROTECCIÓN OCULAR | USO OBLIGATORIO DE MASCARILLA | USO OBLIGATORIO DE PROTECTOR FACIAL |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| USO OBLIGATORIO DE PROTECCIÓN AUDITIVA | APAGAR DESCONECTAR CUANDO NO SE USE | USO OBLIGATORIO DE PROTECTOR AJUSTABLE | OBLIGATORIO CONECTAR A TIERRA | USO OBLIGATORIO DE TRAJE DE SEGURIDAD | OTRO |
|  |  |  |  |  |  |
| Acciones a realizar  (marque con aspa las advertencias por cada acción) | | | | |  |  |  |  |  | |  | Otros riesgos en las acciones  a realizar, considerar  medidas de control | | |
| 1. | | | | |  |  |  |  | |  |  | 1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| 2. | | | | |  |  |  |  | |  |  |
| 3. | | | | |  |  |  |  | |  |  |
| 4. | | | | |  |  |  |  | |  |  |
| 5. | | | | |  |  |  |  | |  |  |
| 6. | | | | |  |  |  |  | |  |  |