

Manipulación de datos

Sesión 04

Ing. Gómez Marín, Jaime¹

¹Fundamentos de Python para Ciencia de Datos
Departamento de TdG

August 2019



- Introducción
- Lectura de archivos
- Escritura de archivos
- Librería Pandas
- Conclusiones
- Bibliografía



En esta sesión se aprenderá a leer y escribir archivos usando las funciones básicas de Python y usando librerías especializadas como Pandas.



Python usa funciones nativas para la lectura de un archivo, los pasos a seguir son:

- Abrir archivo
- Obtener el contenido del archivo
- Cerrar apertura de archivo



```
1 '''
2 Programa : Ciencia de Datos con Pthon
3 Modulo 01 : Fundamentos de Python para Ciencia de Da
4 Sesion 04 : Manipulacion de datos - Leer archivos
5 Fecha : 10/08/2019
6 Version : 1
7 Author :
8 '''
9
10 # Lectura de archivo
11 archivo = open("ejemplo.txt","r")
12
13 # Tipo de atributos
14 print(type(archivo))
15 print(archivo.name)
16 print(archivo.mode)
```



```
17 |
18 | # Lectura de datos de archivo
19 | arch_datos = archivo.read()
20 | print(arch_datos)
21 | print(type(arch_datos))
22 |
23 | # Cerrar archivo
24 | archivo.close()
25 | print(type(archivo))
26 | print(archivo)
```

En python otra forma de leer archivos no necesita que tengamos abierto el archivos, se usa la expresión "with"



```
1 # Otra forma de leer archivos
2 print("-----")
3 with open("ejemplo.txt","r") as archivo:
4     contenido = archivo.read()
5     print(contenido)
6
7
8 print(archivo.closed)
9 print(contenido)
```


Leer archivo - contenido por fragmentos

```
1 # Lectura por fragmentos
2 print("-----")
3 with open("ejemplo.txt","r") as archivo:
4     print(archivo.read(4))
5
6
7 print("-----")
8 with open("ejemplo.txt","r") as archivo:
9     print(archivo.read(4))
10    print(archivo.read(4))
11    print(archivo.read(4))
12    print(archivo.read(4))
```



Leer archivo - por lineas

```
1 # Lectura por linea
2 print("-----")
3 with open("ejemplo.txt","r") as archivo:
4     print("Primera linea : " , archivo.readline())
5
6
7 # Lectura de todas las lineas una por una
8 print("-----")
9 with open("ejemplo.txt","r") as archivo:
10     for linea in archivo:
11         print("Linea :", linea)
```



Leer archivo - por líneas en un listado

```
1 # Lectura de todas las líneas en una lista
2 print("-----")
3 with open("ejemplo.txt","r") as archivo:
4     contenido = archivo.readlines()
5     print("Contenido : " , contenido)
6
7 for linea in contenido:
8     print(linea)
```



En python para escribir archivos usamos la función `open()` con la expresión `"with"`. Un archivo puede ser abierto bajo 2 modalidades, para escribir un archivo nuevo, y el segundo es para agregar datos a un archivo ya existente. En ambos casos si un archivo no existe se crea.



Escribir en un archivo - si no existe lo crea

```
1  '''
2  Programa : Ciencia de Datos con Pthon
3  Modulo 01 : Fundamentos de Python para Ciencia de Da
4  Sesion 04 : Manipulacion de datos - Escribir archivo
5  Fecha : 10/08/2019
6  Version : 1
7  Author :
8  '''
9
10 # Escribir en un archivo
11 print("-----")
12 with open("ejemplo2.txt","w") as archivo:
13     archivo.write("Es una nueva linea")
```



Escribir en un archivo - varias lineas

```
1 # Escribir mas lineas en un archivo
2 print("-----")
3 with open("ejemplo2.txt","w") as archivo:
4     archivo.write("Es una nueva linea")
5     archivo.write("Es otra nueva linea")
6
7 # Escribir mas lineas en un archivo
8 print("-----")
9 with open("ejemplo2.txt","w") as archivo:
10     archivo.write("Es una nueva linea\n")
11     archivo.write("Es otra nueva linea\n")
```



Escribir en un archivo - agregar lineas

```
1  
2 # Agregar una linea a un archivo  
3 print("-----")  
4 with open("ejemplo2.txt","a") as archivo:  
5     archivo.write("Es una nueva linea agregada\n")
```



Copiar un archivo

```
1  
2 # Copiar un archivo  
3 print("-----")  
4 with open("ejemplo2.txt","r") as arch_origen:  
5     with open("ejemplo3.txt", "w") as arch_dest:  
6         for linea in arch_origen:  
7             arch_dest.write(linea)
```



Es una librería de Python escrita para la manipulación y análisis de datos, entre sus características tenemos:

- Manejo de dataframe para la manipulación de datos
- Lectura y escritura de estructuras de datos en memoria
- Maneja diferentes tipos de archivos
- Label-based slicing, fancy indexing, and subsetting of large data sets.



	Posicion	País	Oro	Plata	Bronce	Total	Ranking	Total
1	1	Estados Unidos	120	88	85	293	1	
2	2	Brasil	55	45	71	171	2	
3	3	México	37	36	63	136	4	
4	4	Canadá	35	64	53	152	3	
5	5	Cuba	33	27	38	98	6	
6	6	Argentina	32	35	34	101	5	
7	7	Colombia	28	23	33	84	7	
8	8	Chile	13	19	18	50	8	
9	9	Perú	11	7	21	39	11	
10	10	República Dominicana	10	13	17	40	10	
11	11	Ecuador	10	7	14	31	12	
12	12	Venezuela	9	15	19	43	9	
13	13	Jamaica	6	6	7	19	14	
14	14	Puerto Rico	5	5	14	24	13	

Lectura de archivo CSV con Pandas

```
1  '''
2  Programa : Ciencia de Datos con Pthon
3  Modulo 01 : Fundamentos de Python para Ciencia de Da
4  Sesion 04 : Manipulacion de datos - Leer archivos
5  Fecha : 10/08/2019
6  Version : 1
7  Author :
8  '''
9
10 import pandas as pd
11
12 # Leer archivo csv
13 print("-----")
14 archivo_csv = "medallero_Panamericanos_Lima2019.csv"
15 df = pd.read_csv(archivo_csv)
16 print(df.head())
17 print(df.keys())
```



Lectura de archivo : Series y DataFrame

```
1
2 # Leer un campo del archivo csv : Series
3 print("-----")
4 col_pais = df['Pais']
5 print(type(col_pais))
6 print(col_pais)
7
8 # Leer un campo del archivo csv : DataFrame
9 print("-----")
10 medallas = df[['Pais', 'Oro', 'Plata', 'Bronce']]
11 print(type(medallas))
12 print(medallas)
```



Lectura de filas de un archivo : iloc()

```
1 # Leer una fila
2 print("-----")
3 print(df.head())
4
5 x = 0
6 dato = df.iloc[x]
7 print(type(dato))
8 print(dato)
9
10 x = 1
11 dato = df.iloc[x]
12 print(type(dato))
13 print(dato)
```



Lectura de columnas de un archivo : iloc()

```
1 # Leer una columna
2 print("-----")
3 print(df.head())
4
5 x = 0
6 dato = df.iloc[:,x]
7 print(type(dato))
8 print(dato)
9
10 x = 1
11 dato = df.iloc[:,x]
12 print(type(dato))
13 print(dato)
```

Lectura de multiples columnas y filas : iloc()

```
1 # Leer multiples filas y columnas
2 print("-----")
3 print(df.head())
4
5 dato = df.iloc[0:2]
6 print(type(dato))
7 print(dato)
8
9 dato = df.iloc[:,0:3]
10 print(type(dato))
11 print(dato)
12
13 dato = df.iloc[0:2,0:3]
14 print(type(dato))
15 print(dato)
```



Filtros por columnas : loc() - 1

```
1 #  
2 print("-----")  
3 print(df.head())  
4  
5 df.set_index("País", inplace=True)  
6 print(df.head())  
7  
8 dato = df.loc["Brasil"]  
9 print(type(dato))  
10 print(dato)
```


Filtros por columnas : loc() - 2

```
1 #
2 print("-----")
3 print(df.head())
4
5 #df.set_index("Pais", inplace=True)
6 #print(df.head())
7
8 dato = df.loc[["Brasil","Perú"]]
9 print(type(dato))
10 print(dato)
```

Filtros por columnas : loc() - 3

```
1 print("-----")
2 print(df.head())
3
4 #df.set_index("Pais", inplace=True)
5 #print(df.head())
6
7 dato = df.loc[["Brasil","Perú"],
8               ["Oro","Plata","Bronce"]]
9 print(type(dato))
10 print(dato)
```

Filtros por columnas : loc() - 4

```
1 # Retorna un Dataframe
2 print("-----")
3 print(df.head())
4
5 #df.set_index("Pais", inplace=True)
6 #print(df.head())
7
8 dato = df.loc[df["Plata"] == 1 ,
9               ["Oro","Plata","Bronce"]]
10 print(type(dato))
11 print(dato)
```



Filtros por columnas : loc() - 5

```
1 # Retorna un Dataframe
2 print("-----")
3 print(df.head())
4
5 #df.set_index("Pais", inplace=True)
6 #print(df.head())
7
8 dato = df.loc[df["Oro"] > 10 ,
9               ["Oro","Plata","Bronce"]]
10 print(type(dato))
11 print(dato)
```



Filtros por columnas : loc() - 6

```
1
2 # Retorna un Dataframe
3 print("-----")
4 print(df.head())
5
6 #df.set_index("Pais", inplace=True)
7 #print(df.head())
8
9 dato = df.loc[(df["Oro"]>10) & (df["Oro"]<30)
10              ,["Oro","Plata","Bronce"]]
11 print(type(dato))
12 print(dato)
```

En esta sesión se ha visto la lectura y escritura de archivos dentro del entorno de Python y el empleo de la librería Pandas





Naomi Ceder. The Quick Python Book - Manning Publications, 2018.