

```
In [22]: #-numpy, pandas, matplotlib.py-#
import numpy as np
import pandas as pd

archivo2 = "iris.csv"
with open(archivo2, 'r') as f:
    contenido2 = f.read()

# Mostrar el contenido del archivo
print(contenido2)
```

```
"sepal.length","sepal.width","petal.length","petal.width","variety"
5.1,3.5,1.4,.2,"Setosa"
4.9,3,1.4,.2,"Setosa"
4.7,3.2,1.3,.2,"Setosa"
4.6,3.1,1.5,.2,"Setosa"
5,3.6,1.4,.2,"Setosa"
5.4,3.9,1.7,.4,"Setosa"
4.6,3.4,1.4,.3,"Setosa"
5,3.4,1.5,.2,"Setosa"
4.4,2.9,1.4,.2,"Setosa"
4.9,3.1,1.5,.1,"Setosa"
5.4,3.7,1.5,.2,"Setosa"
4.8,3.4,1.6,.2,"Setosa"
4.8,3,1.4,.1,"Setosa"
4.3,3,1.1,.1,"Setosa"
5.8,4,1.2,.2,"Setosa"
5.7,4.4,1.5,.4,"Setosa"
5.4,3.9,1.3,.4,"Setosa"
5.1,3.5,1.4,.3,"Setosa"
5.7,3.8,1.7,.3,"Setosa"
5.1,3.8,1.5,.3,"Setosa"
5.4,3.4,1.7,.2,"Setosa"
5.1,3.7,1.5,.4,"Setosa"
4.6,3.6,1,.2,"Setosa"
5.1,3.3,1.7,.5,"Setosa"
4.8,3.4,1.9,.2,"Setosa"
5,3,1.6,.2,"Setosa"
5,3.4,1.6,.4,"Setosa"
5.2,3.5,1.5,.2,"Setosa"
5.2,3.4,1.4,.2,"Setosa"
4.7,3.2,1.6,.2,"Setosa"
4.8,3.1,1.6,.2,"Setosa"
5.4,3.4,1.5,.4,"Setosa"
5.2,4.1,1.5,.1,"Setosa"
5.5,4.2,1.4,.2,"Setosa"
4.9,3.1,1.5,.2,"Setosa"
5,3.2,1.2,.2,"Setosa"
5.5,3.5,1.3,.2,"Setosa"
4.9,3.6,1.4,.1,"Setosa"
4.4,3,1.3,.2,"Setosa"
5.1,3.4,1.5,.2,"Setosa"
5,3.5,1.3,.3,"Setosa"
4.5,2.3,1.3,.3,"Setosa"
4.4,3.2,1.3,.2,"Setosa"
5,3.5,1.6,.6,"Setosa"
5.1,3.8,1.9,.4,"Setosa"
4.8,3,1.4,.3,"Setosa"
5.1,3.8,1.6,.2,"Setosa"
4.6,3.2,1.4,.2,"Setosa"
5.3,3.7,1.5,.2,"Setosa"
5,3.3,1.4,.2,"Setosa"
7,3.2,4.7,1.4,"Versicolor"
6.4,3.2,4.5,1.5,"Versicolor"
6.9,3.1,4.9,1.5,"Versicolor"
5.5,2.3,4,1.3,"Versicolor"
6.5,2.8,4.6,1.5,"Versicolor"
5.7,2.8,4.5,1.3,"Versicolor"
6.3,3.3,4.7,1.6,"Versicolor"
4.9,2.4,3.3,1,"Versicolor"
6.6,2.9,4.6,1.3,"Versicolor"
5.2,2.7,3.9,1.4,"Versicolor"
5,2,3.5,1,"Versicolor"
5.9,3,4.2,1.5,"Versicolor"
6,2.2,4,1,"Versicolor"
```

6.1,2.9,4.7,1.4,"Versicolor"
5.6,2.9,3.6,1.3,"Versicolor"
6.7,3.1,4.4,1.4,"Versicolor"
5.6,3,4.5,1.5,"Versicolor"
5.8,2.7,4.1,1,"Versicolor"
6.2,2.2,4.5,1.5,"Versicolor"
5.6,2.5,3.9,1.1,"Versicolor"
5.9,3.2,4.8,1.8,"Versicolor"
6.1,2.8,4,1.3,"Versicolor"
6.3,2.5,4.9,1.5,"Versicolor"
6.1,2.8,4.7,1.2,"Versicolor"
6.4,2.9,4.3,1.3,"Versicolor"
6.6,3,4.4,1.4,"Versicolor"
6.8,2.8,4.8,1.4,"Versicolor"
6.7,3,5,1.7,"Versicolor"
6,2.9,4.5,1.5,"Versicolor"
5.7,2.6,3.5,1,"Versicolor"
5.5,2.4,3.8,1.1,"Versicolor"
5.5,2.4,3.7,1,"Versicolor"
5.8,2.7,3.9,1.2,"Versicolor"
6,2.7,5.1,1.6,"Versicolor"
5.4,3,4.5,1.5,"Versicolor"
6,3.4,4.5,1.6,"Versicolor"
6.7,3.1,4.7,1.5,"Versicolor"
6.3,2.3,4.4,1.3,"Versicolor"
5.6,3,4.1,1.3,"Versicolor"
5.5,2.5,4,1.3,"Versicolor"
5.5,2.6,4.4,1.2,"Versicolor"
6.1,3,4.6,1.4,"Versicolor"
5.8,2.6,4,1.2,"Versicolor"
5,2.3,3.3,1,"Versicolor"
5.6,2.7,4.2,1.3,"Versicolor"
5.7,3,4.2,1.2,"Versicolor"
5.7,2.9,4.2,1.3,"Versicolor"
6.2,2.9,4.3,1.3,"Versicolor"
5.1,2.5,3,1.1,"Versicolor"
5.7,2.8,4.1,1.3,"Versicolor"
6.3,3.3,6,2.5,"Virginica"
5.8,2.7,5.1,1.9,"Virginica"
7.1,3,5.9,2.1,"Virginica"
6.3,2.9,5.6,1.8,"Virginica"
6.5,3,5.8,2.2,"Virginica"
7.6,3,6.6,2.1,"Virginica"
4.9,2.5,4.5,1.7,"Virginica"
7.3,2.9,6.3,1.8,"Virginica"
6.7,2.5,5.8,1.8,"Virginica"
7.2,3.6,6.1,2.5,"Virginica"
6.5,3.2,5.1,2,"Virginica"
6.4,2.7,5.3,1.9,"Virginica"
6.8,3,5.5,2.1,"Virginica"
5.7,2.5,5,2,"Virginica"
5.8,2.8,5.1,2.4,"Virginica"
6.4,3.2,5.3,2.3,"Virginica"
6.5,3,5.5,1.8,"Virginica"
7.7,3.8,6.7,2.2,"Virginica"
7.7,2.6,6.9,2.3,"Virginica"
6,2.2,5,1.5,"Virginica"
6.9,3.2,5.7,2.3,"Virginica"
5.6,2.8,4.9,2,"Virginica"
7.7,2.8,6.7,2,"Virginica"
6.3,2.7,4.9,1.8,"Virginica"
6.7,3.3,5.7,2.1,"Virginica"
7.2,3.2,6,1.8,"Virginica"
6.2,2.8,4.8,1.8,"Virginica"

```
6.1,3,4.9,1.8,"Virginica"  
6.4,2.8,5.6,2.1,"Virginica"  
7.2,3,5.8,1.6,"Virginica"  
7.4,2.8,6.1,1.9,"Virginica"  
7.9,3.8,6.4,2,"Virginica"  
6.4,2.8,5.6,2.2,"Virginica"  
6.3,2.8,5.1,1.5,"Virginica"  
6.1,2.6,5.6,1.4,"Virginica"  
7.7,3,6.1,2.3,"Virginica"  
6.3,3.4,5.6,2.4,"Virginica"  
6.4,3.1,5.5,1.8,"Virginica"  
6,3,4.8,1.8,"Virginica"  
6.9,3.1,5.4,2.1,"Virginica"  
6.7,3.1,5.6,2.4,"Virginica"  
6.9,3.1,5.1,2.3,"Virginica"  
5.8,2.7,5.1,1.9,"Virginica"  
6.8,3.2,5.9,2.3,"Virginica"  
6.7,3.3,5.7,2.5,"Virginica"  
6.7,3,5.2,2.3,"Virginica"  
6.3,2.5,5,1.9,"Virginica"  
6.5,3,5.2,2,"Virginica"  
6.2,3.4,5.4,2.3,"Virginica"  
5.9,3,5.1,1.8,"Virginica"
```

```
In [23]: from csv import reader  
  
archivo = "iris.csv"  
  
with open(archivo, "r") as f:  
    listaDatos = []  
    file = reader(f)  
  
    for fila in file:  
        listaDatos.append(fila)  
  
#print(listaDatos)  
  
listaD = listaDatos[1:len(listaDatos)]  
  
for elem in listaD:  
    print(elem)
```

Cell In[23], line 9
for fila in file:
^
IndentationError: unexpected indent

```
In [19]: import pandas as pd  
  
datos = pd.read_csv("iris.csv")  
datos.head()
```

Out[19]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

```
In [20]: datos = pd.read_csv("iris.csv", header = None)
datos.head()
```

```
Out[20]:
```

	0	1	2	3	4
0	sepal.length	sepal.width	petal.length	petal.width	variety
1	5.1	3.5	1.4	.2	Setosa
2	4.9	3	1.4	.2	Setosa
3	4.7	3.2	1.3	.2	Setosa
4	4.6	3.1	1.5	.2	Setosa

```
In [21]: datos.columns = ["cha1", "cha2", "cha3", "cha4", "Target"]
print(datos)
```

	cha1	cha2	cha3	cha4	Target
0	sepal.length	sepal.width	petal.length	petal.width	variety
1	5.1	3.5	1.4	.2	Setosa
2	4.9	3	1.4	.2	Setosa
3	4.7	3.2	1.3	.2	Setosa
4	4.6	3.1	1.5	.2	Setosa
..
146	6.7	3	5.2	2.3	Virginica
147	6.3	2.5	5	1.9	Virginica
148	6.5	3	5.2	2	Virginica
149	6.2	3.4	5.4	2.3	Virginica
150	5.9	3	5.1	1.8	Virginica

[151 rows x 5 columns]

```
In [25]: #Impresion de una columna#
col1 = datos["cha1"]
print(col1)

col5 = datos["Target"]
print(col5)
```

```
0    sepal.length
1           5.1
2           4.9
3           4.7
4           4.6
...
146          6.7
147          6.3
148          6.5
149          6.2
150          5.9
Name: cha1, Length: 151, dtype: object
0    variety
1    Setosa
2    Setosa
3    Setosa
4    Setosa
...
146  Virginica
147  Virginica
148  Virginica
149  Virginica
150  Virginica
Name: Target, Length: 151, dtype: object
```

```
In [27]: #Impresion de varias columnas#
col12 = datos[["cha1", "Target"]]
print(col12)
```

	cha1	Target
0	sepal.length	variety
1	5.1	Setosa
2	4.9	Setosa
3	4.7	Setosa
4	4.6	Setosa
..
146	6.7	Virginica
147	6.3	Virginica
148	6.5	Virginica
149	6.2	Virginica
150	5.9	Virginica

[151 rows x 2 columns]

```
In [28]: #Impresion de varias columnas diferente orden#
col12 = datos[["Target", "cha1"]]
print(col12)
```

	Target	cha1
0	variety	sepal.length
1	Setosa	5.1
2	Setosa	4.9
3	Setosa	4.7
4	Setosa	4.6
..
146	Virginica	6.7
147	Virginica	6.3
148	Virginica	6.5
149	Virginica	6.2
150	Virginica	5.9

[151 rows x 2 columns]

```
In [30]: #Impresion de filas#
fila1 = datos[3:6]
print(fila1)
```

	cha1	cha2	cha3	cha4	Target
3	4.7	3.2	1.3	.2	Setosa
4	4.6	3.1	1.5	.2	Setosa
5	5	3.6	1.4	.2	Setosa

```
In [33]: fila1 = datos[1:10]
print(fila1)
```

	cha1	cha2	cha3	cha4	Target
1	5.1	3.5	1.4	.2	Setosa
2	4.9	3	1.4	.2	Setosa
3	4.7	3.2	1.3	.2	Setosa
4	4.6	3.1	1.5	.2	Setosa
5	5	3.6	1.4	.2	Setosa
6	5.4	3.9	1.7	.4	Setosa
7	4.6	3.4	1.4	.3	Setosa
8	5	3.4	1.5	.2	Setosa
9	4.4	2.9	1.4	.2	Setosa

```
In [41]: filx= datos[1:3]
print(filx)
```

```
colx = filx["cha3"]
print(colx)

#Tambien#
datosx = datos[1:3]["cha3"]
print(datosx)
```

```
   cha1 cha2 cha3 cha4 Target
1  5.1  3.5  1.4   .2  Setosa
2  4.9   3  1.4   .2  Setosa
1    1.4
2    1.4
Name: cha3, dtype: object
1    1.4
2    1.4
Name: cha3, dtype: object
```

In [42]: *#Impresion de varias filas de columnas diferentes#*

```
datosz = datos[3:7][["cha1", "Target"]]
print(datosz)
```

```
   cha1 Target
3  4.7  Setosa
4  4.6  Setosa
5    5  Setosa
6  5.4  Setosa
```

In [46]: *#Tipo de dato: DATAFRAME#*

```
print(type(datos))
print(type(col1))
print(type(fila1))
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
<class 'pandas.core.frame.DataFrame'>
```

In [53]: **import** pandas **as** pd

```
#Tablas
##Encabezados: Los equipos
equipos = [
    ["Cristal", 10], ["Alianza", 25], ["Total Clean", 20], ["Melgar", 5]
]
print(equipos)

datos1 = pd.DataFrame(equipos)

#Agregar etiquetas a las columnas#
datos1.columns = ["equipo", "copas"]
print(datos1)
```

```
[['Cristal', 10], ['Alianza', 25], ['Total Clean', 20], ['Melgar', 5]]
   equipo  copas
0   Cristal    10
1   Alianza    25
2 Total Clean    20
3    Melgar     5
```

In [55]:

```
notas = [
    ["Electronicos", 15], ["Antenas", 10], ["Dispositivos", 4]
]
datos2 = pd.DataFrame(notas, columns = ["Materia", "Notas"])

print(datos2)
```

```
   Materia  Notas
0 Electronicos    15
1    Antenas     10
2 Dispositivos     4
```

```
In [61]: import pandas as pd

#Diccionario#
ventas = {"frutas":["Manzana", "Pera", "Sandia", "Naranja"], "costo":[2, 4.5, 1, 1.5], "stock": [200, 100, 40, 300]}
print(ventas)
print(type(ventas))

#Pasar el DataFrame#
datos3 = pd.DataFrame(ventas)
print(datos3)

print(type(datos3))
```

{'frutas': ['Manzana', 'Pera', 'Sandia', 'Naranja'], 'costo': [2, 4.5, 1, 1.5], 'stock': [200, 100, 40, 300]}

<class 'dict'>

	frutas	costo	stock
0	Manzana	2.0	200
1	Pera	4.5	100
2	Sandia	1.0	40
3	Naranja	1.5	300

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

```
In [62]: #Condicionales, Filtros#
datos4 = datos3.loc[datos3["costo"]>=2]
print(datos4)
```

	frutas	costo	stock
0	Manzana	2.0	200
1	Pera	4.5	100

```
In [65]: datos5 = datos3.loc[datos3["frutas"] == "Manzana" or datos3["frutas"] == "Naranja"]
print(datos5)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[65], line 1
----> 1 datos5 = datos3.loc[datos3["frutas"] == "Manzana" or datos3["frutas"] == "Naranja"]
      2 print(datos5)

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:1527, in NDFrame.__nonzero__(self)
    1525 @final
    1526 def __nonzero__(self) -> NoReturn:
-> 1527     raise ValueError(
    1528         f"The truth value of a {type(self).__name__} is ambiguous. "
    1529         "Use a.empty, a.bool(), a.item(), a.any() or a.all()."
    1530     )

ValueError: The truth value of a Series is ambiguous. Use a.empty, a.bool(), a.item(), a.any() or a.all().
```

```
In [67]: datos6 = datos3[datos3["stock"]>=100]
print(datos6)
```

	frutas	costo	stock
0	Manzana	2.0	200
1	Pera	4.5	100
3	Naranja	1.5	300

```
In [68]: writer = csv.writer(datos6, delimiter = ',')
print("Hola Mundo")
```



```
-----
NameError                                Traceback (most recent call last)
Cell In[68], line 1
----> 1 writer = csv.writer(datos6, delimiter = ',')
      2 print("Hola Mundo")

NameError: name 'csv' is not defined
```

```
In [72]: import pandas as pd

ventasRobots = {"nombres": ["Honda", "Autobots", "Decepticons"], "costos": [1000, 2000, 3000],
                "stock": [400, 600, 1000]}
datos7 = pd.DataFrame(ventasRobots)
print(datos7)

print(datos7.shape)
print(datos7.columns)
print(datos7.shape[0])

datos7 = datos7.assign(descuento = datos7["costos"]*datos7["stock"]/10000*(3.14))
print(datos7)
```

	nombres	costos	pais	stock
0	Honda	1000	Korea	400
1	Autobots	2000	Peru	600
2	Decepticons	3000	Bolivia	1000

(3, 4)

Index(['nombres', 'costos', 'pais', 'stock'], dtype='object')

3

	nombres	costos	pais	stock	descuento
0	Honda	1000	Korea	400	125.6
1	Autobots	2000	Peru	600	376.8
2	Decepticons	3000	Bolivia	1000	942.0

```
In [73]: #--Combinar tablas MERCH y JOIN--#

salarios = {'Id_empleado': [1, 3, 4], 'Salario': [2000, 2500, 1800]}
df_salarios = pd.DataFrame(salarios)
print(df_salarios)
```

	Id_empleado	Salario
0	1	2000
1	3	2500
2	4	1800

```
In [75]: import pandas as pd
empleados = {'Id_empleado': [1, 2, 3], 'Nombre': ['Juan', 'María', 'Pedro'], 'Edad': [25, 30, 28]}
df_empleados = pd.DataFrame(empleados)
print("-----Tabla de empleados-----")
print(df_empleados)

df_combinado = pd.merge(df_empleados, df_salarios, on = 'Id_empleado', how = 'inner')
print(df_combinado)
```

```
-----Tabla de empleados-----
   Id_empleado  Nombre  Edad
0             1    Juan   25
1             2  María   30
2             3   Pedro   28

   Id_empleado  Nombre  Edad  Salario
0             1    Juan   25    2000
1             3   Pedro   28    2500
```

```
In [76]: # Combina las tablas en base a la columna común 'Id_empleado'
df_combinada = pd.merge(df_empleados, df_salarios, on='Id_empleado', how='inner')
```

```
print("-----Inner-----")
print(df_combinada)
```

```
-----Tabla Combinada-----
   Id_employado Nombre  Edad  Salario
0              1   Juan    25    2000
1              3  Pedro    28    2500
```

```
In [77]: # Combina las tablas en base a la columna 'Id_employado'
df_combinada = pd.merge(df_empleados, df_salarios, on='Id_employado', how='left')

print("-----Left-----")
print(df_combinada)
```

```
-----Tabla combinada-----
   Id_employado Nombre  Edad  Salario
0              1   Juan    25    2000.0
1              2  María    30         NaN
2              3  Pedro    28    2500.0
```

```
In [78]: # Combina las tablas en base a la columna común 'Id_employado'
df_combinada = pd.merge(df_empleados, df_salarios, on='Id_employado', how='right')

print("-----Right-----")
print(df_combinada)
```

```
-----Right-----
   Id_employado Nombre  Edad  Salario
0              1   Juan    25.0    2000
1              3  Pedro    28.0    2500
2              4   NaN     NaN    1800
```

```
In [79]: import pandas as pd
# Crear DataFrame 1
df1 = pd.DataFrame({'id': [1, 2, 3, 4],
                    'nombre': ['Juan', 'Pedro', 'María', 'Ana']})
print("-----Tabla 1-----")
print(df1)

# Crear DataFrame 2
df2 = pd.DataFrame({'id': [2, 3, 5],
                    'edad': [25, 30, 20]})
print("-----Tabla 2-----")
print(df2)

# Realizar Full Join
df3 = pd.merge(df1, df2, on='id', how='outer')
print("-----Tabla junta-----")
print(df3)
```

```
-----Tabla 1-----
  id nombre
0  1  Juan
1  2  Pedro
2  3  María
3  4   Ana

-----Tabla 2-----
  id  edad
0  2    25
1  3    30
2  5    20

-----Tabla junta-----
  id nombre  edad
0  1  Juan   NaN
1  2  Pedro  25.0
2  3  María  30.0
3  4   Ana   NaN
4  5   NaN   20.0
```

In []: