

Instituto de Capacitación y Asesoría en Informática de la Escuela de Informática



Programación en Python Básico

Ing Luis Diego Gamboa Chaverri, Mag

Agenda del día

- Librería Pandas
- Sitio oficial: <https://pandas.pydata.org/pandas-docs/stable/index.html>
- Recuerde ver el Pandas_Cheat_Sheet.pdf en los recursos

Qué es Pandas

- Pandas es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python.
- Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales
- Bajo la licencia BSD.
- El nombre deriva del término "datos de panel", término de econometría que designa datos que combinan una dimensión temporal con otra dimensión transversal.

*Tomado de wikipedia

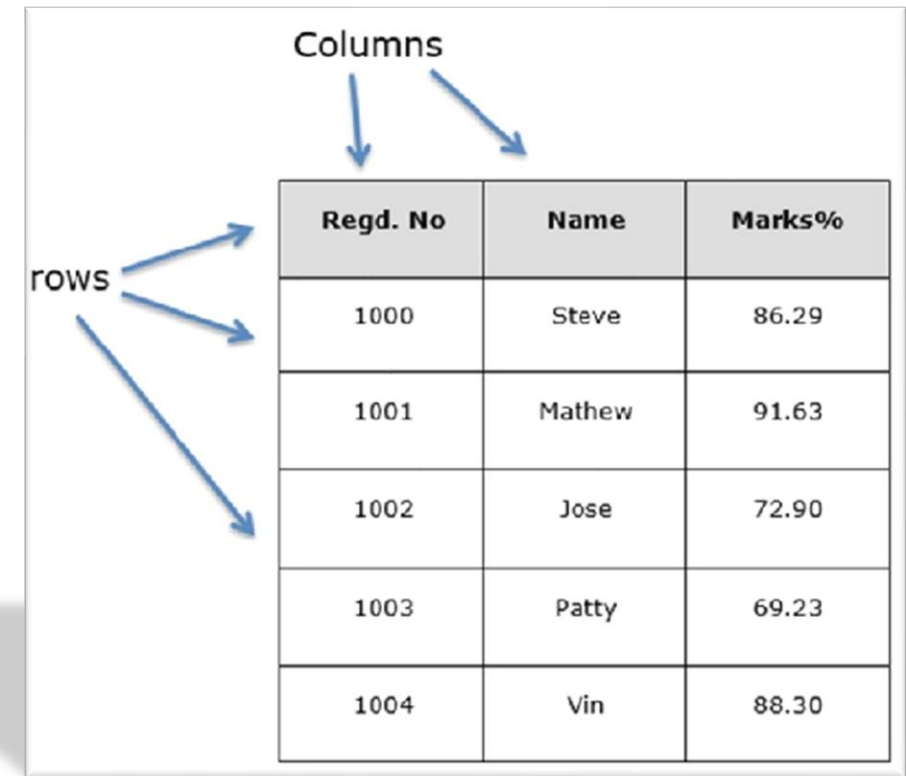
Cómo Importarlo?

- `from pandas import DataFrame , read_csv`
- `import pandas as pd`

Creación de Dataframe

Dataframe

- Un Dataframe es una estructura de datos bidimensional, es decir, los datos se alinean de forma tabular en filas y columnas
- Sus principales características son :
 - Las columnas pueden ser de diferentes tipos.
 - Tamaño: mutable
 - Ejes etiquetados (filas y columnas)
 - Puede realizar operaciones aritméticas en filas y columnas



The diagram shows a table representing a Dataframe. Above the table, the word "Columns" has two arrows pointing to the "Regd. No" and "Name" headers. To the left of the table, the word "rows" has three arrows pointing to the first three rows of the table.

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Dataframe

- Constructor
 - `pandas.DataFrame(data, index, columns, dtype, copy)`
 - Ver <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

Parámetro	Descripción
Data	Los datos toman varias formas como ndarray, series, mapas, listas, diccionarios, constantes y también otro DataFrame.
Index	Para las etiquetas de fila, el índice que se utilizará para el dataframe resultante es <code>np.arange</code> . Si no se pasa nada el valor predeterminado es <code>np.arange (n)</code> , que es el tamaño de los datos .
Columns	Para las etiquetas de columna, la sintaxis predeterminada es - <code>np.arange (n)</code> . Esto sólo es cierto si no se pasa ningún índice.
Dtype	Tipo de cada columna
Copy	Este comando se usa para copiar datos. El valor predeterminado es <code>False</code>

Dataframes

- Como estructuras de entrada para crear los dataframe's se pueden usar
 - Lists
 - Diccionarios
 - Series
 - Numpy ndarrays
 - Otros DataFrames

Ejemplo: Dataframe vacío

```
from pandas import DataFrame , read_csv  
import pandas as pd
```

```
df = pd.DataFrame()  
print(df)
```

```
Empty DataFrame  
Columns: []  
Index: []
```

Dataframe: a partir de una lista

```
#dataframe a partir de una lista  
data = [1,2,3,4,5]  
df = pd.DataFrame(data)  
print(df)
```

	0
0	1
1	2
2	3
3	4
4	5

Dataframe: a partir de una lista

```
data = [['Juan',10],['Paco',12],['Luis',13]]  
df = pd.DataFrame(data,columns=['Nombre','Años'])  
print(df)
```

	Nombre	Años
0	Juan	10
1	Paco	12
2	Luis	13

*Observe: si el índice no es incluido por defecto este es `range(n)`, donde `n` es el tamaño del arreglo

Dataframe: a partir de una lista

```
data = [['Juan',10],['Paco',12],['Luis',13]]  
df = pd.DataFrame(data,columns=['Nombre','Años'],dtype=float)  
print(df)
```

	Nombre	Años
0	Juan	10.0
1	Paco	12.0
2	Luis	13.0

* Observe el cambio de tipo aplicado

Dataframe: a partir de diccionarios de arrays y listas

```
data = {'Nombre': ['Tom', 'Jack', 'Steve', 'Ricky'], 'Años': [28, 34, 29, 42]}  
df = pd.DataFrame(data, index=['rank1', 'rank2', 'rank3', 'rank4'])  
print(df)
```

	Nombre	Años
rank1	Tom	28
rank2	Jack	34
rank3	Steve	29
rank4	Ricky	42

Dataframe: usando lista de diccionarios

```
data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]
df = pd.DataFrame(data)
print(df)
```

	a	b	c
0	1	2	NaN
1	5	10	20.0

Observe el NaN que se agregó. Que es?

Dataframe: usando lista de diccionarios

```
data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]  
df = pd.DataFrame(data, index=['primero', 'segundo'])  
print(df)
```

	a	b	c
primero	1	2	NaN
segundo	5	10	20.0

Dataframe: usando diccionarios e índices de columna

```
data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]
# se usando índices de columnas que existen
df1 = pd.DataFrame(data, index=['primero', 'segundo'], columns=['a', 'b'])
# se usa un índice de columna que no existe
df2 = pd.DataFrame(data, index=['primero', 'segundo'], columns=['a', 'b1'])
print(df1)
print(df2)
```

	a	b
primero	1	2
segundo	5	10

	a	b1
primero	1	NaN
segundo	5	NaN

Dataframe: usando diccionario y series

```
: d = {'one' : pd.Series ([1. , 2., 3.] ,index=[ 'a', 'b', 'c'])  
      , 'two' : pd.Series ([1. , 2., 3., 4.] , index=[ 'a', 'b', 'c', 'd'])}
```

```
: df = pd.DataFrame(d)
```

```
: df
```

	one	two
a	1.0	1.0
b	2.0	2.0
c	3.0	3.0
d	NaN	4.0

Manipulación de Dataframe

Cargar desde una fuente externa

- Por medio de Pandas podemos extraer los datos de fuentes tales como: archivos de texto, csv , Excel entre otros

```
import matplotlib.pyplot as plt
import pandas as pd
color_table = pd.io.parsers.read_table("C:\\Util\\UNA\\Python\\Semana6\\resources\\Colors.txt")
print(color_table)
```

```
import pandas as pd
titanic = pd.io.parsers.read_csv("C:\\Util\\UNA\\Python\\Semana6\\resources\\Titanic.csv")
print(titanic)
```

```
import pandas as pd
xls = pd.ExcelFile("C:\\Util\\UNA\\Python\\Semana6\\resources\\Values.xls")
trig_values = xls.parse('Sheet1', index_col=None, na_values=['NA'], skiprows=[0])
print(trig_values)
```

0	138.550574	0.661959	-0.749540	-0.883153
1	305.535745	-0.813753	0.581211	-1.400100
2	280.518695	-0.983195	0.182556	-5.385709
3	216.363795	-0.592910	-0.805269	0.736289
4	36.389247	0.593268	0.805005	0.736974
...
66	324.199562	-0.584964	0.811059	-0.721234
67	187.948172	-0.138277	-0.990394	0.139619
68	270.678249	-0.999930	0.011837	-84.472139
69	270.779159	-0.999908	0.013598	-73.530885
70	200.213513	-0.345520	-0.938412	0.368196

[71 rows x 4 columns]

Obtener los tipos de datos

```
: # verifica el tipo de las columnas  
color_table.dtypes
```

```
: Color      object  
Value        int64  
dtype: object
```

```
# verifica el tipo de una columna en especial  
color_table.Value.dtype
```

```
dtype('int64')
```

Cabeza y Cola

	Color	Value
0	Red	1
1	Orange	2
2	Yellow	3
3	Green	4
4	Blue	5
5	Purple	6
6	Black	7
7	White	8

```
: color_table.head(2)
```

	Color	Value
0	Red	1
1	Orange	2

```
: color_table.tail(2)
```

	Color	Value
6	Black	7
7	White	8

Columnas , valores e índices

```
color_table.columns
```

```
Index(['Color', 'Value'], dtype='object')
```

```
color_table.index
```

```
RangeIndex(start=0, stop=8, step=1)
```

```
color_table.values
```

```
array([[ 'Red', 1],  
       [ 'Orange', 2],  
       [ 'Yellow', 3],  
       [ 'Green', 4],  
       [ 'Blue', 5],  
       [ 'Purple', 6],  
       [ 'Black', 7],  
       [ 'White', 8]], dtype=object)
```

Descripción de datos

```
color_table['Value'].max()
```

8

```
color_table['Value'].min()
```

1

```
color_table.describe()
```

	Value
count	8.00000
mean	4.50000
std	2.44949
min	1.00000
25%	2.75000
50%	4.50000
75%	6.25000
max	8.00000

Agregar una Columna

```
color_table["Color_Esp"] = ['Rojo', 'Naranja', 'Amarillo', 'Verde', 'Azul', 'Purpura', 'Negro', 'Blanco']
```

color_table

	Color	Value	Color_Esp
0	Red	1	Rojo
1	Orange	2	Naranja
2	Yellow	3	Amarillo
3	Green	4	Verde
4	Blue	5	Azul
5	Purple	6	Purpura
6	Black	7	Negro
7	White	8	Blanco

Borrar Columna

```
del color_table["Color_Esp"]  
color_table
```

	Color	Value
0	Red	1
1	Orange	2
2	Yellow	3
3	Green	4
4	Blue	5
5	Purple	6
6	Black	7
7	White	8

```
# otra forma es mediante el uso de drop , el cual permite eliminar varias columnas  
# por medio del nombre:  
df.drop(['Col1', 'Col3'], axis='columns', inplace=True)  
# por medio de número de columna (los índices de columna comienzan en cero)  
df.drop(df.columns[[0, 2]], axis='columns')
```

Agregar filas

```
# agregar una fila  
color_table = pd.DataFrame([[ 'Brown',9]], columns=["Color", "Value"]).append(color_table, ignore_index=True)  
print(color_table)
```

	Color	Value
0	Brown	9
1	Red	1
2	Orange	2
3	Yellow	3
4	Green	4
5	Blue	5
6	Purple	6
7	Black	7
8	White	8

Acceso a datos del Dataframe

- El acceso se realiza por medio del uso del índice o nombre de la columnas

```
color_table["Color"]
```

1	Red
2	Orange
3	Yellow
4	Green
5	Blue
6	Purple
7	Black
8	White
0	Brown

Name: Color, dtype: object

```
color_table[["Color_Esp", "Value"]]
```

	Color_Esp	Value
1	Rojo	1
2	Naranja	2
3	Amarillo	3
4	Verde	4
5	Azul	5
6	Purpura	6
7	Negro	7
8	Blanco	8
0	Cafe	9

Instituto de Capacitación y Asesoría en Informática de la Escuela de Informática

icai@una.cr

www.icaia.ac.cr

