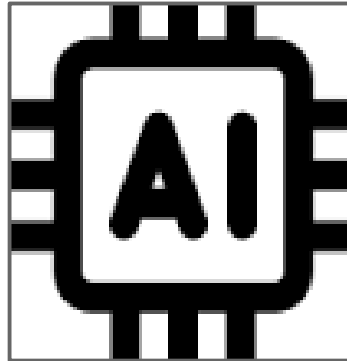


Introducción a la Inteligencia Artificial



Uso de la Librería Pandas

Acerca de Pandas

“Pandas es un paquete de Python que proporciona estructuras de datos similares a los dataframes de R. Pandas depende de Numpy, la librería que añade un potente tipo matricial a Python.”

Pandas es una herramienta de manipulación de datos de alto nivel desarrollada por Wes McKinney. Es construido con el paquete Numpy y su estructura de datos clave es llamada el DataFrame. El DataFrame te permite almacenar y manipular datos tabulados en filas de observaciones y columnas de variables.

La documentación oficial de Pandas puede hallarse en su sitio:
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Conformación básica de un DataFrame Pandas

```
dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98] }
import pandas as pd
brics = pd.DataFrame(dict)
print(brics)
```


Fuentes: <https://www.learnpython.org/es/Pandas%20Basics>

Contacto: ia@frh.utn.edu.ar

Conformación básica de un DataFrame Pandas

```
dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98]}
import pandas as pd
brics = pd.DataFrame(dict)
print(brics)
```

DATOS



Variables



Fuentes: <https://www.learnpython.org/es/Pandas%20Basics>

Contacto: ia@frh.utn.edu.ar

Conformación básica de un DataFrame Pandas

```
dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98] }
import pandas as pd
brics = pd.DataFrame(dict)
print(brics)
```

```
[1]: runfile('/home/brics.py', wdir='/home')
      country  capital  area  population
0      Brazil  Brasilia  8.516      200.40
1      Russia   Moscow  17.100      143.50
2       India  New Dehli  3.286     1252.00
3       China   Beijing  9.597     1357.00
4  South Africa  Pretoria  1.221       52.98
```

Fuentes: <https://www.learnpython.org/es/Pandas%20Basics>

Contacto: ia@frh.utn.edu.ar

Conformación básica de un DataFrame Pandas

```
dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98] }
import pandas as pd
brics = pd.DataFrame(dict)
print(brics)
```

```
[1]: runfile('/home/brics.py', wdir='/home')
```

	country	capital	area	population	← Variables
0	Brazil	Brasilia	8.516	200.40	
1	Russia	Moscow	17.100	143.50	
2	India	New Dehli	3.286	1252.00	
3	China	Beijing	9.597	1357.00	
4	South Africa	Pretoria	1.221	52.98	

Índice

DATOS

Fuentes: <https://www.learnpython.org/es/Pandas%20Basics>

Contacto: ia@frh.utn.edu.ar

¿Qué es un Archivo CSV?

“Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal como en Chile, Perú, Argentina, España, Brasil, entre otros) y las filas por saltos de línea”.

Son archivos que suelen usarse para almacenar grandes volúmenes de datos, en tablas sin formato: sin líneas, ni formato de texto. No incluye fórmulas. Es muy económico para almacenar datos.

Pandas tiene incluidos métodos muy simples para transformar dataframes en archivos csv.

Fuentes: https://es.wikipedia.org/wiki/Valores_separados_por_comas

Almacenar un Dataframe en un archivo CSV

Si se dispone de un dataframe pandas, exportarlo a CSV para almacenamiento es muy sencillo.

```
brics.to_csv('brics.csv')
```

Método

Nombre
Archivo

Nombre del
dataframe

Obtener un Dataframe desde un archivo CSV

Análogamente, si se posee un archivo CSV, importarlo como dataframe pandas es muy simple.

```
df=pd.read_csv('brics.csv')
```

Método

Nombre
Archivo

Nombre del
dataframe

Obtener un Dataframe desde un archivo CSV

```
df=pd.read_csv('brics.csv')
```

```
df
```

```
Out[6]:
```

	Unnamed: 0	country	capital	area	population
0	0	Brazil	Brasilia	8.516	200.40
1	1	Russia	Moscow	17.100	143.50
2	2	India	New Dehli	3.286	1252.00
3	3	China	Beijing	9.597	1357.00
4	4	South Africa	Pretoria	1.221	52.98

Obtener un Dataframe desde un archivo CSV

```
df=pd.read_csv('brics.csv')
```

```
df
```

```
Out[6]:
```

	Unnamed: 0	country	capital	area	population
0	0	Brazil	Brasilia	8.516	200.40
1	1	Russia	Moscow	17.100	143.50
2	2	India	New Dehli	3.286	1252.00
3	3	China	Beijing	9.597	1357.00
4	4	South Africa	Pretoria	1.221	52.98

Obtener un Dataframe desde un archivo CSV

```
df=pd.read_csv('brics.csv',index_col=0)
```

```
df
```

```
Out[7]:
```

	country	capital	area	population
0	Brazil	Brasilia	8.516	200.40
1	Russia	Moscow	17.100	143.50
2	India	New Dehli	3.286	1252.00
3	China	Beijing	9.597	1357.00
4	South Africa	Pretoria	1.221	52.98

Indicando en la importación “index_col=0”, se asume que la columna 0 corresponde al índice, y no agrega otra columna a tal fin

Presentar un Resumen de los Datos

Presentación del Dataset Titanic

```
df=pd.read_csv('datasets/titanic.csv', index_col=0)
```

```
df
```

```
Out[8]:
```

	Survived	Pclass	...	Cabin	Embarked
PassengerId			...		
1	0	3	...	NaN	S
2	1	1	...	C85	C
3	1	3	...	NaN	S
4	1	1	...	C123	S
5	0	3	...	NaN	S
...
887	0	2	...	NaN	S
888	1	1	...	B42	S
889	0	3	...	NaN	S
890	1	1	...	C148	C
891	0	3	...	NaN	Q

Presentación de la forma del Dataset

Para obtener la cantidad de Filas y Columnas del Dataset

```
df.shape
```

```
Out[22]: (891, 11)
```

Presentar un Resumen de los Datos

Presentación de las primeras líneas del Dataset

```
df.head()
```

```
Out[10]:
```

	Survived	Pclass	...	Cabin	Embarked
PassengerId			...		
1	0	3	...	NaN	S
2	1	1	...	C85	C
3	1	3	...	NaN	S
4	1	1	...	C123	S
5	0	3	...	NaN	S

```
[5 rows x 11 columns]
```

Presentar un Resumen de los Datos

Descripción General del Dataset

```
df.describe()
```

```
Out[23]:
```

	Survived	Pclass	...	Parch	Fare
count	891.000000	891.000000	...	891.000000	891.000000
mean	0.383838	2.308642	...	0.381594	4026.270117
std	0.486592	0.836071	...	0.806057	22153.876204
min	0.000000	1.000000	...	0.000000	0.000000
25%	0.000000	2.000000	...	0.000000	8.050000
50%	0.000000	3.000000	...	0.000000	19.500000
75%	1.000000	3.000000	...	0.000000	56.929200
max	1.000000	3.000000	...	6.000000	262375.000000

```
[8 rows x 6 columns]
```


Obtener Info de una celda

```
df.iloc[0,1]  
Out[24]: 3
```

Se obtiene el valor del elemento de la fila 0, columna 1.

Obtener Info de un subconjunto del Dataframe

```
df.iloc[1,:] # Se obtienen todos los elementos de la fila 1.
```

```
Out[23]:
```

```
PassengerId      2
Survived         1
Pclass           1
Name      Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex              female
Age             38
SibSp           1
Parch           0
Ticket          PC 17599
Fare            71.2833
Cabin           C85
Embarked         C
```

```
Name: 1, dtype: object
```

FIN