

```
In [2]: import pandas as pn
import numpy as np
import sympy as sp
```

TEMA 2: DATA LOADING IN PANDAS

```
In [3]: from IPython.display import Image
Image("tabla_acceso_datos.jpg")
```

Out[3]:

Function	Description
read_csv	Load delimited data from a file, URL, or file-like object. Use comma as default delimiter
read_table	Load delimited data from a file, URL, or file-like object. Use tab as default delimiter
read_fwf	Read data in fixed-width column format (that is, no delimiters)
read_clipboard	Version of read_table that reads data from the clipboard. Useful for converting tables from web pages

```
In [4]: dataframe=pn.read_csv('tabla_ejemplo.csv')
```

```
In [5]: dataframe.index=['a','b','c','d','e','f']
```

```
In [6]: dataframe
```

Out[6]:

	color	altura	peso
a	azul	1.45	3
b	amarillo	1.34	4
c	verde	1.67	2
d	morado	1.68	5
e	rosa	1.35	6
f	rojo	1.23	4

Cuando la tabla no tiene header tenemos 2 opciones:

a) Especificas que la tabla no tiene títulos de columna con **header=None** y pandas añade un header aleatorio

```
In [7]: pn.read_csv('tabla_ejemplo.csv', header=None)
```

Out[7]:

	0	1	2
0	color	altura	peso
1	azul	1.45	3
2	amarillo	1.34	4
3	verde	1.67	2
4	morado	1.68	5
5	rosa	1.35	6
6	rojo	1.23	4

b) Añades tú mismo los títulos de columna con **names=[]**:

```
In [8]: tabla1=pn.read_csv('tabla_ejemplo.csv', names=['color','altura','peso'])
tabla1
```

Out[8]:

	color	altura	peso
0	color	altura	peso
1	azul	1.45	3
2	amarillo	1.34	4
3	verde	1.67	2
4	morado	1.68	5
5	rosa	1.35	6
6	rojo	1.23	4

Especificar que una columna es el índice

Para ello está el comando **index_col='titulo_columna'**, en el que especificamos el título de la columna que funciona de índice y ésta se pondrá como el índice (el título no se incluye en el índice):

```
In [9]: tabla1=pn.read_csv('tabla_ejemplo.csv', index_col='peso')
tabla1
```

Out[9]:

	color	altura
peso		
3	azul	1.45
4	amarillo	1.34
2	verde	1.67
5	morado	1.68
6	rosa	1.35
4	rojo	1.23

Si tenemos varios índices, simplemente hacemos una lista con los títulos de los índices en **index_col=[indice1,indice2,...]**:

```
In [10]: tabla2=pn.read_csv('tabla_ejemplo_2.csv', index_col=['indice2','indice1'])
tabla2 #El one en indice1 se repite 4 veces, por eso al ponerlo como primer
#indice te lo pone así de bonito, pero si lo pones como segundo no.
```

Out[10]:

		color	altura	peso
	indice2	indice1		
one	a	azul	1.45	3
	b	amarillo	1.34	4
	c	verde	1.67	2
	d	morado	1.68	5
two	e	rosa	1.35	6
	f	rojo	1.23	4

Tratamiento de valores nulos en una tabla

Para tratar estos datos es necesario especificar qué elementos tienen un valor nulo, dado que en las tablas que cogemos de distintos sitios se puede dar el caso de que el valor nulo no se ponga como NaN, que es la forma que tiene pandas de marcar los valores nulos. Eso se hace gracias a **na_values={titulo_columna_donde_esta_el_valor_nulo_1:[valor_nulo1,valor_nulo2],titulo_columna_donde_esta_el_valor_nulo2:[valor_nulo3,valor_nulo4]}**. Es decir, para marcar los lugares que tienen valor NaN se hce con un diccionario:

```
In [15]: tabla3=pn.read_csv('tabla_ejemplo_2.csv', index_col=['indice2','indice1'], na_values={'color':['azul','morado'],'peso':[2,5]})
tabla3
```

Out[15]:

		color	altura	peso
	indice2	indice1		
one	a	NaN	1.45	3.0
	b	amarillo	1.34	4.0
	c	verde	1.67	NaN
	d	NaN	1.68	NaN
two	e	rosa	1.35	6.0
	f	rojo	1.23	4.0

Como podemos ver los valores indicados ahora tienen valor nulo

```
In [ ]:
```