Introduction

`pandas` can write and read data files from the disk.

A very common file format is CSV. A CSV file is a text file whose columns are separated by `,` . Sometimes, this is not a comma but another character: `;` , `/` , etc...

When reading a file, `pandas` try to guess the data types.

Read a CSV file

Let's define a CSV file `data.csv` and use the `read_csv` function:

```
Integer;Some value;Another column;Date
1;3.45;True;21/02/2026 12:56:54
5;2.75;False;21/02/2026 14:25:08
2;4.15;False;21/02/2026 16:56:41
```

In [1]:
```python
import pandas as pd
df = pd.read_csv('data.csv', index_col=0, sep=';')
df
```

Out[1]:

Integer	Some value	Another column	Date
1	3.45	True	21/02/2026 12:56:54
5	2.75	False	21/02/2026 14:25:08
2	4.15	False	21/02/2026 16:56:41

A quick dtype check:

In [2]:
```python
df.dtypes
```

Out[2]:
```
Some value        float64
Another column       bool
Date               object
dtype: object
```

The dates have not been interprated as dates but as strings.

We need an additional argument in `read_csv` to specify the date format:

In [3]:
```python
df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%Y %H:%M:%S')
df
```

Out[3]:

Integer	Some value	Another column	Date
1	3.45	True	2026-02-21 12:56:54
5	2.75	False	2026-02-21 14:25:08
2	4.15	False	2026-02-21 16:56:41

In [4]:
```python
df.dtypes
```

Out[4]:
```
Some value              float64
Another column             bool
Date             datetime64[ns]
dtype: object
```

Many other arguments exist for `read_csv` :

- `names` : define new names for the columns, instead of those of the file
- `index_col` : select the column to use as index (axis 0)
- `comment` : tell pandas no to care about rows starting with a specific character: these are not data
- `skiprows` , `skipfooter` , `nrows` : read only specific rows

Write a CSV file

The `to_csv` method of `DataFrame` and `Serie` is used hereafter.

In [5]:
```python
df['Last column'] = df['Some value'] / 8
df.to_csv('data_write.csv', sep=',',
          index=False) # do not write index
```

Result:

```
Some value,Another column,Date,Last column
3.45,True,2026-02-21 12:56:54,0.43125
2.75,False,2026-02-21 14:25:08,0.34375
4.15,False,2026-02-21 16:56:41,0.51875
```

Other file formats

`pandas` handle a lot of IO file formats: Excel, json, html, hdf, etc...

The dedicated methods are called `read_[...]` (data import) and `to_[...]` (data export).