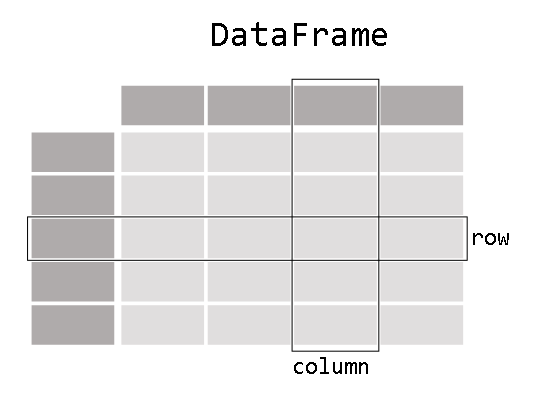
**What kind of data does pandas handle?**

In panda column is called series.

Combination of series called dataframe.

When working with tabular data, such as data stored in spreadsheets or databases, pandas is the right tool for you. pandas will help you to explore, clean, and process your data. In pandas, a data table is called a **[DataFrame](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html" \l "pandas.DataFrame" \o "pandas.DataFrame)**.



First we start with pandas then we should import first

Import pandas as pd

To load the pandas package and start working with it, import the package.

Suppose I want to store passenger data. When we store it in dataframe it display in pandas like

Name Age Sex

0 Braund, Mr. Owen Harris 22 male

1 Allen, Mr. William Henry 35 male

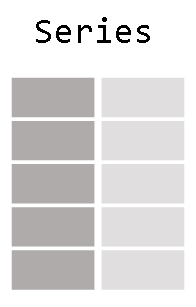
2 Bonnell, Miss. Elizabeth 58 female

Practically applied in vscode

To manually store data in a table, create a DataFrame. When using a Python dictionary of lists, the dictionary keys will be used as column headers and the values in each list as columns of the DataFrame.

A **[DataFrame](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html" \l "pandas.DataFrame" \o "pandas.DataFrame)** is a 2-dimensional data structure that can store data of different types (including characters, integers, floating point values, categorical data and more) in columns. It is similar to a spreadsheet, a SQL table or the data.frame in R.

Each column in a DataFrame is a Series



I’m just interested in working with the data in the column Age

df["Age"]

When selecting a single column of a pandas **[DataFrame](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html" \l "pandas.DataFrame" \o "pandas.DataFrame)**, the result is a pandas [**Series**](https://pandas.pydata.org/docs/reference/api/pandas.Series.html#pandas.Series). To select the column, use the column label in between square brackets [].

You can create a Series from scratch as well:

ages = pd.Series([22, 35, 58], name="Age")

A pandas Series has no column labels, as it is just a single column of a DataFrame. A Series does have row labels.

## Do something with a DataFrame or Series

I want to know the maximum Age of the passengers

We can do this on the DataFrame by selecting the Age column and applying max():

df["Age"].max()

Or to the Series:

ages.max()

As illustrated by the max() method, you can do things with a DataFrame or Series. pandas provides a lot of functionalities, each of them a method you can apply to a DataFrame or Series. As methods are functions, do not forget to use parentheses ().

I’m interested in some basic statistics of the numerical data of my data table

df.describe()

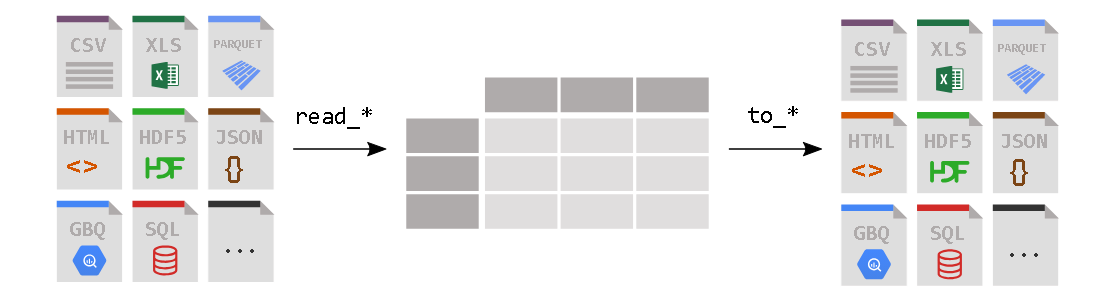
The [**describe()**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html#pandas.DataFrame.describe) method provides a quick overview of the numerical data in a DataFrame. It gives all the description of all columns in the table.

#### REMEMBER

* Import the package, aka import pandas as pd
* A table of data is stored as a pandas DataFrame
* Each column in a DataFrame is a Series
* You can do things by applying a method to a DataFrame or Series

**How do I read and write tabular data?**

pandas supports the integration with many file formats or data sources out of the box (csv, excel, sql, json, parquet,…). Importing data from each of these data sources is provided by function with the prefix read\_\*. Similarly, the to\_\* methods are used to store data.



I want to analyze the Titanic passenger data, available as a CSV file.

titanic = pd.read\_csv("data/titanic.csv")

pandas provides the **[read\_csv()](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html" \l "pandas.read_csv" \o "pandas.read_csv)** function to read data stored as a csv file into a pandas DataFrame. pandas supports many different file formats or data sources out of the box (csv, excel, sql, json, parquet, …), each of them with the prefix read\_\*.

I want to see the first 8 rows of a pandas DataFrame.

titanic.head(8)

A check on how pandas interpreted each of the column data types can be done by requesting the pandas dtypes attribute:

titanic.dtypes

For each of the columns, the used data type is enlisted. The data types in this DataFrame are integers (int64), floats (float64) and strings (object).

My colleague requested the Titanic data as a spreadsheet.

titanic.to\_excel("titanic.xlsx", sheet\_name="passengers", index=False)

Whereas read\_\* functions are used to read data to pandas, the to\_\* methods are used to store data. The **[to\_excel()](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html" \l "pandas.DataFrame.to_excel" \o "pandas.DataFrame.to_excel)** method stores the data as an excel file. In the example here, the sheet\_name is named passengers instead of the default Sheet1. By setting index=False the row index labels are not saved in the spreadsheet.

The equivalent read function **read\_excel()** will reload the data to a DataFrame:

titanic = pd.read\_excel("titanic.xlsx", sheet\_name="passengers")

I’m interested in a technical summary of a DataFrame

titanic.info()

#### REMEMBER

* Getting data in to pandas from many different file formats or data sources is supported by read\_\* functions.
* Exporting data out of pandas is provided by different to\_\*methods.
* The head/tail/info methods and the dtypes attribute are convenient for a first check.