



Import dei dati



Importiamo il primo file .csv

Dopo aver importato la libreria *pandas* (*import pandas as pd*), eseguiamo il suo metodo *read_csv*

```
clienti = pd.read_csv(filepath_or_buffer = path + r"\Clienti.csv",
    sep = ";", #separtore del file, la virgola è il default
    header = 0, #per indicare quale riga contiene l'intestazione,
    #0 vuol dire che l'intestazione è nella prima riga
    #se non c'è intestazione utilizzare None
    #names = ["NumeroCliente", "IsActive", "Nome", "Cognome", "DataNascita"]
    # se non avessi l'intestazione assegnerei i nome con l'attributo names
    usecols = ["NumeroCliente", "Nome", "DataNascita", "Regione"], #colonne che vogliamo importare
    #dtype = {"NumeroCliente": np.str}, #tipi delle colonne
    #parse_dates=["DataNascita"], #colonne da importare come date
    #dayfirst=True, #MOLTO IMPORTANTE, per indicare che nelle date il giorno è indicato prima del mese
    #decimal=".", #separatore dei decimali, il default è il punto
    #index_col = "NumeroCliente", #impostare una colonna come indice
    #nrows = 10 #numero di righe da importare,
    #skiprows = 0, #per saltare alcune righe all'inizio del file
)
```

Abbiamo creato il nostro primo DataFrame

Il DataFrame è una struttura tabellare dove le righe:

- hanno un vincolo di tipo
- hanno un nome (l'indice)
- hanno un ordinamento implicito

```
[25]: clienti
```

```
[25]:
```

	NumeroCliente	Nome	DataNascita	Regione
0	1	Nicoletta	01/01/2010	NaN
1	2	Giovanni	01/03/1976	Lazio
2	3	Marco	01/04/1980	Lazio
3	4	Giovanna	01/05/1977	Lazio
4	5	Alice	01/06/1969	Sicilia
5	6	Fabrizio	01/07/1996	Sicilia
6	7	Irene	01/08/1990	Sicilia
7	8	Maria	01/09/1999	Sicilia



Informazioni sul Dataframe

```
[48]: type(clienti)
```

```
[48]: pandas.core.frame.DataFrame
```

```
[49]: clienti.dtypes
```

```
[49]: NumeroCliente    int64  
Nome              object  
DataNascita       object  
Regione           object  
dtype: object
```

```
[50]: clienti.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 39 entries, 0 to 38  
Data columns (total 4 columns):  
#   Column          Non-Null Count  Dtype    
---  ---            
0   NumeroCliente  39 non-null    int64    
1   Nome           39 non-null    object    
2   DataNascita    39 non-null    object    
3   Regione        38 non-null    object    
dtypes: int64(1), object(3)  
memory usage: 1.3+ KB
```

```
[52]: clienti.columns
```

```
[52]: Index(['NumeroCliente', 'Nome', 'DataNascita', 'Regione'], dtype='object')
```

```
[53]: clienti.shape
```

```
[53]: (39, 4)
```

```
[54]: clienti.index
```

```
[54]: RangeIndex(start=0, stop=39, step=1)
```

Analogamente posso importare dati da excel e json

```
: from_excel = pd.read_excel(io = path + r'/FattureDettaglio.xlsx',  
                             sheet_name = 'Tabelle1',  
                             usecols = 'A:D,F',  
                             header = 0  
                             )
```

```
clienti_j = pd.read_json(path_or_buf = path + "/Clienti.json")  
clienti_j
```

```
: import json  
  
with open(path + "/Clienti2.json", "r") as f:  
    data_json = json.load(f)  
clienti2_j = pd.json_normalize(data_json)
```

Conversioni più "ostiche"

```
fatture['DataFattura2'] = pd.to_datetime(fatture['DataFattura'],  
                                         format="%d/%m/%Y",  
                                         )
```

```
fatture['NumeroFornitore'] = fatture['NumeroFornitore'].astype('Int64')
```

```
fatture['Iva'] = fatture['Iva'].fillna(0).astype(np.int64)
```

```
fatture2['NumeroFornitore'] = fatture2['NumeroFornitore'].astype('str').replace('<NA>', pd.NA)
```

Funzione per importare un file di testo strutturato

```
def acquisisci_file_fasta(path, file ):
    #acquisiamo il contenuto del file all'interno della variabile testo
    with open(os.path.join(path,file),"r") as f:
        testo = f.read()
    #creiamo una lista a partire dalla variabile di tipo stringa ed eliminiamo il primo elemento
    lista = testo.split(">")
    lista.pop(0)
    #creiamo una lista vuota da popolare con gli elementi della lista precedente
    lista_dati = []
    for el in lista:
        accessId = el.split("\n")[0].split(" ")[0]
        seq = el.split("\n")[1]
        lista_dati.append([accessId, seq])
    #utilizziamo la lista appena creata per definire un DataFrame di Pandas
    df = pd.DataFrame(columns = ['accessid','sequenza'], data = lista_dati)
    return df
```

Chiamiamo la funzione

```
dataframe = acquisisci_file_fasta(path = r"C:\Users\ianto\Desktop\python\file", file = "file_input_fasta.txt")
dataframe
```

Import da API: libreria requests

```
[1]: import requests
```

```
[2]: url = "https://api.github.com/repos/iantomasinicola/PortfolioDataAnalyst"
```

```
[3]: res = requests.get(url)
```

```
[4]: json_res = res.json()
```

```
[5]: json_res
```

```
[5]: {'id': 451192837,  
      'node_id': 'R_kgDOGuSoBQ',  
      'name': 'PortfolioDataAnalyst',  
      'full_name': 'iantomasinicola/PortfolioDataAnalyst',  
      'private': False,  
      'owner': {'login': 'iantomasinicola',  
                 'id': 59792312,
```


Import da SQL Server

```
[22]: import getpass
import pandas as pd
from sqlalchemy import create_engine
```

```
[5]: server_name = 'LAPTOP-UDP6N0UL\\SQLEXPRESS'
database_name = 'CorsoSQL'
utente = getpass.getpass(prompt='Inserisci nome utente: ')
password = getpass.getpass(prompt='Inserisci la password: ')

conn_str = f'mssql+pyodbc://{utente}:{password}@{server_name}/{database_name}?driver=ODBC+Driver+17+for+SQL+Server'
```

```
Inserisci nome utente: .....
Inserisci la password: .....
```

```
[6]: engine = create_engine(conn_str)
```

```
[7]: df = pd.read_sql( sql = "SELECT * \
                        FROM Clienti \
                        WHERE RegioneResidenza = 'Lombardia'",
                        con = engine)
```

Import da MongoDB

```
[1]: from pymongo import MongoClient

[2]: client = MongoClient("mongodb://localhost:27017")

[3]: db = client.corso_mongodb

[4]: result = db.fatture.find(
        {"importo":{"$gt" : 25}},
        {"id_fattura":1,
         "data_fattura":1,
         "importo":1,
         "_id":0}
    );

[5]: result_df = pd.DataFrame(result)
```