

Gestione dei file: TXT, CSV, TSV

Rachele Sprugnoli

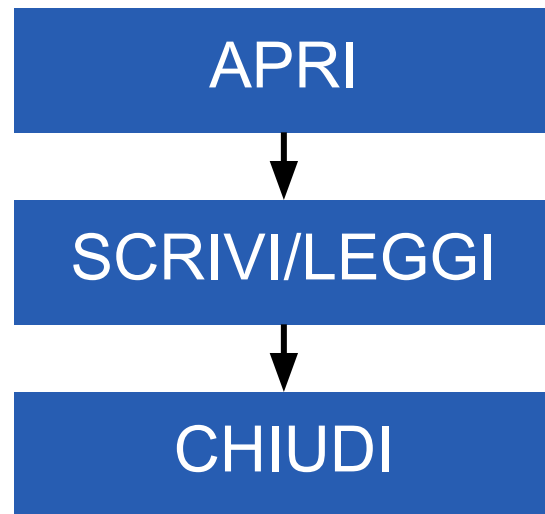
rachele.sprugnoli@unipr.it



**UNIVERSITÀ
DI PARMA**

Leggere e scrivere file

- Quando un programma è in esecuzione, i suoi dati sono archiviati nella RAM (*Random Access Memory*): veloce ma volatile
 - Quando il programma termina o il computer si spegne, i dati nella RAM scompaiono
- Per rendere i dati sempre disponibili è necessario scriverli su un supporto di memorizzazione non volatile
 - file
- Per aprire un file, specifichiamo il suo nome e indichiamo se lo vogliamo scrivere o leggere; poi lo dobbiamo chiudere





Fate delle copie dei file su cui volete lavorare e usate quelle: tenete sempre da parte una copia pulita di backup

Aprire e leggere un file

- Leggere:

```
with open("file.txt", "r") as f:  
    f.read()
```

- Scrivere:

```
with open("file.txt", "w") as f:  
    f.write(data)
```

- with: istruzione, **apre e chiude** il file
- open(): funzione
- argomento: r per read e w per write

→ ATTENZIONE all'indentazione

Aprire e leggere un file: alternativa

```
infile = open("commediaDante.txt", "r")  
infile.read()  
outfile = open("output.txt", "w")  
outfile.write("Hello!")  
infile.close()  
outfile.close()
```

- apre il file `commediaDante.txt`
- legge il file `commediaDante.txt`
- apre il file `output.txt` (se non esiste lo crea)
- scrive "Hello!" nel file `output.txt`
- chiude il file `commediaDante.txt`
- chiude il file `output.txt`

La funzione open()

```
open(file, mode='r', buffering=-1, encoding=None,  
errors=None, newline=None, closefd=True, opener=None)
```

<https://docs.python.org/3/library/functions.html#open>

- file: nome del file
- mode: modo d'uso
- encoding: utf-8
- newline: carattere che identifica la fine di una riga

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open for updating (reading and writing)

```
with open("file.txt", "r", encoding="utf-8") as f:  
    f.read()
```

Metodi

- Lettura:

`f.read()` → legge tutto il file

`f.read(size)` → legge la quantità di caratteri specificata

`f.readline()` → legge una sola riga

`f.readlines()` → legge tutte le righe

- Scrittura:

`f.write(string)` → scrive la stringa nel file

ATTENZIONE: se create un file con lo stesso nome di un altro file presente nella stessa cartella, quest'ultimo viene sovrascritto!



File CSV-TSV

- CSV = *comma separated values*
- TSV = *tab separated values*

ATTENZIONE: non sempre l'estensione del file corrisponde al delimitatore usato (ad es., ci sono file con colonne separate da tab che hanno come estensione .csv)



```
# sent_id = 1
# text = Gallia est omnis diuvisa in partes tres quarum unam incolunt Belgae
# aliam Aquitani tertiam qui ipsorum lingua Celtae nostra Galli appellantur
1 Gallia Gallia PROPN - - - - -
2 est sum AUX - - - - -
3 omnis omnis DET - - - - -
4 diuvisa diuido VERB - - - - -
5 in in ADP - - - - -
6 partes pars NOUN - - - - -
7 tres tres NUM - - - - -
8 quarum qui PRON - - - - -
9 unam unus DET - - - - -
10 incolunt incolo VERB - - - - -
11 Belgae Belgae PROPN - - - - -
12 aliam alius DET - - - - -
13 Aquitani Aquitani PROPN - - - - -
14 tertiam tertius ADJ - - - - -
15 qui qui PRON - - - - -
16 ipsorum ipse DET - - - - -
17 lingua lingua NOUN - - - - -
18 Celtae Celtae PROPN - - - - -
19 nostra noster DET - - - - -
20 Galli Galli PROPN - - - - -
21 appellantur appello VERB - - - - -
```


File CSV-TSV: lettura

- Serve un apposito modulo della libreria standard: csv,
<https://docs.python.org/3/library/csv.html>

All'inizio bisogna importare la libreria

```
import csv
```

```
with open('file.csv', 'r', encoding='UTF8') as f:  
    reader = csv.reader(f)  
    for row in reader:  
        print(row)
```

Metodo di lettura: come parametri si passano almeno il file (f) e il delimitatore (se non è presente, di default è la virgola)

File CSV-TSV: lettura

- Serve un apposito modulo della libreria standard: csv,
<https://docs.python.org/3/library/csv.html>

```
import csv
```

```
with open('file.csv', 'w', encoding='UTF8') as f:  
    reader = csv.reader(f, delimiter=';')  
    for row in reader:  
        print(row)
```



In caso di tabulazioni: '\t'

File CSV-TSV: scrittura

- Serve un apposito modulo della libreria standard: csv,
<https://docs.python.org/3/library/csv.html>

All'inizio bisogna importare la libreria

```
import csv
```

```
with open('some.csv', 'w', encoding='UTF8') as f:
```

```
    writer = csv.writer(f)
```

```
    writer.writerow(row)
```

Metodo di creazione del file: come parametri si passano almeno il file (f) e il delimitatore (se non è presente, di default è la virgola)

Metodo di scrittura di una riga

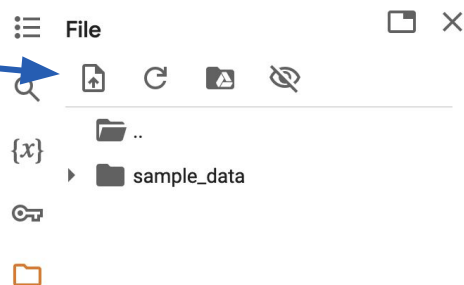
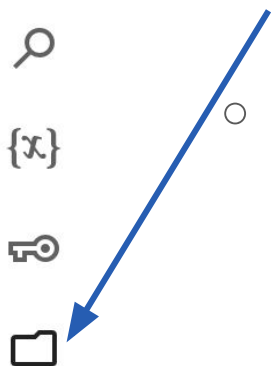
Un po' di pratica



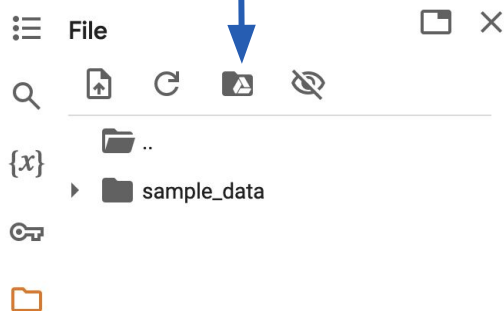
- Lezione4.ipynb

<https://colab.research.google.com/drive/10kj-PUveBLXQh7NpyecjzBfTLxvOua9?usp=sharing>

- Cliccare sull'icona della cartella nella barra verticale sinistra
- Trascinare il file `commediaDante.txt` o caricarlo usando l'icona di upload



I file caricati o creati nel notebook vengono rimossi automaticamente quando viene chiuso. Per salvarli in maniera permanente, cliccare su “monta Drive”. In ogni caso, sempre meglio salvare tutto anche in locale!



Esercizio 1



- Cerca nel file “canzonierePetrarca.txt” tutte le parole che finiscono in “-mente”
 - Salva l’output della ricerca in un file

Esercizio 2

- Dividi i testi nel file “canzonierePetrarca.txt” per verso
 - Salva l’output della ricerca in un file

Esercizio 3

- Rinomina il file *canzonierePetrarca.txt* in *Petrarca_canzoniere.txt*

Soluzioni esercizi



- https://colab.research.google.com/drive/1zGR7mjS_zfoEflQrwi-XM8WTQ2puUn2L?usp=sharing