



Python - Preparazione dei dati per le metriche

Qui dobbiamo prendere i dati grezzi raccolti dalle lezioni che sono state fatte e renderli possibili da leggere, insieme una sistemazione per farlo più volte in base all'anno.

✓ Sistemazione dell'algoritmo

```
import pandas as pd
import os
ANNO = 2023
OPERZIONE = 2
NOME = 'Lezioni.csv'
#DATI = 'Data\Da_aggiungere.csv'
DATI = 'Data\Data.csv'

df = pd.read_csv(DATI) #Da_aggiungere
df.head()
```

	Ora	Data	Slide	Professoressa
0	8	2205	B132	Virginia
1	11	2205	B135	Graziana
2	10	2305	C123	Francesca
3	11	2305	B143	Sara
4	10	2405	B141	Alice

✓ Riparando le colonne

- Lezione

```
def livello(valore):
    valore = list(valore)
    valore = f' {valore[1]}{valore[2]}'
    return valore

#livello(' C123')
df[' Livello'] = df[' Slide'].apply(livello)
```

✓ Le date

Qui provare oltre a correggere le date, si proverà ad aggiungere altre colonne per avere una comprensione più nel dettaglio

```
mapping_day = {"Monday": "Lunedì",
               "Tuesday": "Martedì",
               "Wednesday": "Mercoledì",
               "Thursday": "Giovedì",
               "Friday": "Venerdì",
               "Saturday": "Sabato",
               "Sunday": "Domenica"}

mapping_month = {"January": "Gennaio",
                "February": "Febbraio",
                "March": "Marzo",
                "April": "Aprile",
                "May": "Maggio",
                "June": "Giugno",
                "July": "Luglio",
                "August": "Agosto",
                "September": "Settembre",
                "October": "Ottobre",
                "November": "Novembre",
                "December": "Dicembre"}
```

- Date

```
df[' Data'] = df[' Data'].astype(str)
df[' Data'] = df[' Data'].str.zfill(4).str.replace(r'^(\d{2})(\d{2})$', r'\1-\2', regex=True)

def agg_anno(valore):
    valore = str(valore)
    valore = f'{valore}-{ANNO}'
    return valore

df[' Data'] = df[' Data'].apply(agg_anno)

df.head()
```

	Ora	Data	Slide	Professoressa	Livello
0	8	22-05-2023	B132	Virginia	B1
1	11	22-05-2023	B135	Graziana	B1
2	10	23-05-2023	C123	Francesca	C1
3	11	23-05-2023	B143	Sara	B1
4	10	24-05-2023	B141	Alice	B1

```
df[' Data'] = pd.to_datetime(df[' Data'].astype(str), format='%d-%m-%Y').dt.strftime('%d-%m-%Y')
df[' Data'] = pd.to_datetime(df[' Data'].astype(str), format='%d-%m-%Y')

df.head()
```

	Ora	Data	Slide	Professoressa	Livello
0	8	2023-05-22	B132	Virginia	B1
1	11	2023-05-22	B135	Graziana	B1
2	10	2023-05-23	C123	Francesca	C1
3	11	2023-05-23	B143	Sara	B1
4	10	2023-05-24	B141	Alice	B1

```
df[' Giorno della settimana'] = pd.to_datetime(df[' Data']).dt.day_name()
df[' Mese'] = pd.to_datetime(df[' Data']).dt.month_name()
df[' Anno'] = ANNO

df[" Giorno della settimana"] = df[" Giorno della settimana"].replace(mapping_day)
df[" Mese"] = df[" Mese"].replace(mapping_month)

df.head()
```

	Ora	Data	Slide	Professoressa	Livello	Giorno della settimana	Mese	Anno
0	8	2023-05-22	B132	Virginia	B1	Lunedì	Maggio	2023
1	11	2023-05-22	B135	Graziana	B1	Lunedì	Maggio	2023
2	10	2023-05-23	C123	Francesca	C1	Martedì	Maggio	2023

```
df = df.rename(columns={' Slide':'Lezione', ' Data':'Data',  
                        ' Professoressa':'Professoressa', ' Livello':'Livello',  
                        ' Giorno della settimana':'Giorno della settimana',  
                        ' Mese':'Mese', ' Anno':'Anno'})
```

```
df = df.reindex(columns=['Data', 'Ora', 'Professoressa',  
                        'Livello', 'Lezione', 'Giorno della settimana',  
                        'Mese', 'Anno']) # Riordinando le colonne
```

```
df.head()
```

	Data	Ora	Professoressa	Livello	Lezione	Giorno della settimana	Mese	Anno
0	2023-05-22	8	Virginia	B1	B132	Lunedì	Maggio	2023
1	2023-05-22	11	Graziana	B1	B135	Lunedì	Maggio	2023
2	2023-05-23	10	Francesca	C1	C123	Martedì	Maggio	2023

▼ Conclusione

Fin qui è già tutto di cui avevamo bisogno. Ma ciò che si vuole fare dopo può cambiare in base all'anno, o se si vuole metterli dentro ai dati principali.

▼ I dati da unire

- Dati di pertenza
- Dati nuovi

```
df.head()
```

	Data	Ora	Professoressa	Livello	Lezione	Giorno della settimana	Mese	Anno
0	2023-05-22	8	Virginia	B1	B132	Lunedì	Maggio	2023
1	2023-05-22	11	Graziana	B1	B135	Lunedì	Maggio	2023
2	2023-05-23	10	Francesca	C1	C123	Martedì	Maggio	2023

```
def conclusione(operazione:int, csv:None):  
    op = operazione  
    if op == 1:  
        dataframe = pd.read_csv(csv)  
        dataframe = dataframe.drop('Unnamed: 0', axis=1)  
        df['Data'] = df['Data'].astype(str)  
  
        df_finale = pd.concat([dataframe, df])  
        os.remove('Lezioni.csv')  
        df_finale.to_csv('Lezioni.csv')  
        return df_finale  
    elif op == 2:  
        df.to_csv('Lezioni.csv')
```

```
conclusione(OPERZIONE, NOME)
```

```

#df['Data'] = df['Data'].astype(str)

df.dtypes

Data          datetime64[ns]
Ora            int64
Professoressa object
Livello        object
Lezione         object
Giorno della settimana object
Mese            object
Anno            int64
dtype: object

df = pd.read_csv('Lezioni.csv')

# Correzione degli spazi

def strip(valore):
    tipo = str(type(valore))
    valore = str(valore)
    valore = valore.strip()

    if tipo == "<class 'int'>":
        valore = int(valore)
    elif tipo == "<class 'str'>":
        valore = str(valore)
    else:
        valore = float(valore)

    return valore

lista_colonne = df.columns.to_list()

for x in lista_colonne:
    df[x] = df[x].apply(strip)
df.to_csv('Lezioni.csv')

df['Data'] = pd.to_datetime(df['Data'].astype(str), format='%Y-%m-%d') df.to_csv('Lezioni.csv')

totale, _ = df.shape
totale = totale + 1
index = list(map(lambda x: x, range(1, totale)))

df = df.assign(Index=index)

df = df.reindex(columns=['Index', 'Data', 'Ora', 'Professoressa',
                        'Livello', 'Lezione', 'Giorno della settimana',
                        'Mese', 'Anno']) # Riordinando le colonne

df.to_csv('Lezioni.csv')

df.dtypes

Index          int64
Data           object
Ora            int64
Professoressa object
Livello         object
Lezione         object
Giorno della settimana object
Mese            object
Anno            int64
dtype: object

df.head()

```