



# Esercizi lezione 6

## Esercizio

Dal database **AdventureWorks** estraiamo la tabella dimproduct

- Sulla colonna DealerPrice, utilizzando il metodo `.round()`, arrotondiamo i valori alle due cifre decimali, e poi al valore intero più vicino
- Utilizzando il metodo `.clip()`, facciamo in modo che i valori siano compresi tra un minimo di 0 e un massimo di 1000

## Esercizio

Creiamo un DataFrame sintetico, che contiene i guadagni mensili di diverse annate, con il seguente codice:

```
years = 5  
guadagni = pd.DataFrame({ "Mese": list("GFAMGLASOND")*years),  
                         "Anno": np.repeat(list(range(years)), 12),  
                         "Valore": np.random.randint(800, 5000, 12*years) })
```

- Calcola la somma cumulativa dei guadagni utilizzando il metodo `.cumsum()`
- Come sopra, ma raggruppato per ogni anno usando prima un `.groupby()`

## Esercizio 1/2

Dal database **AdventureWorks** estraiamo la tabella `dimcustomer`

- Trasformiamo i nomi dei clienti in modo che abbiano solo lettere minuscole, e i cognomi in modo che abbiano solo lettere maiuscole
- Sulla colonna `EmailAddress`, utilizzando il metodo `.str.split()`, estraiamo nome utente e dominio
- Sulla colonna `Phone`, estraiamo ogni parte del numero (ad es. da "1 (11) 500 555-0162" a ["1", "(11)", "500", "555-0162"])
- Utilizzando il metodo `.str.contains()`, estraiamo tutti gli indirizzi e-mail che contengono il numero "21"

## Esercizio 2/2

- Estraiamo tutti gli indirizzi e-mail che contengono il numero "20" oppure il numero "10"
- Calcolare la lunghezza di ogni indirizzo e-mail ed estrarre i cinque più lunghi e i cinque più corti
- Modificare il dominio degli indirizzi e-mail da "adventure-works.com" a "aw-db.com" mediante il metodo `.str.replace()`
- Dalla colonna AddressLine1 estraiamo tutti gli indirizzi che contengono la sottostringa "Street"

## Esercizio 1/2

Dai **beginner\_datasets** carichiamo in un DataFrame il file `facebook.csv`, che contiene dei post con data di pubblicazione, tipo (foto, video, ...) e numero di reactions raccolte:

- Con la funzione `pd.to_datetime()` convertiamo la colonna `status_published` in formato Timestamp
- Utilizzando gli attributi `.dt.year`, `.dt.month`, `.dt.day`,  
`.dt.dayofweek`, `.dt.dayofyear`, ottieniamo informazioni specifiche sulle date delle transazioni, come l'anno, il mese, il giorno della settimana, il giorno dell'anno, eccetera

## Esercizio 2/2

- Estraiamo solo i post relativi al 2012
- Estraiamo solo i post relativi a maggio 2018
- Confrontiamo il numero di post pubblicati nei weekend rispetto al numero di post pubblicati nel resto della settimana
- Troviamo il primo e ultimo post pubblicati in ogni anno
- Quanti tipi di post ci sono? E quanti per ogni tipo?

## Esercizio

Dai `beginner_dataset` carichiamo in un DataFrame il file `pokemon.csv`:

- Tramite i metodi `.isnull()` e `.sum()` controlliamo se ci sono valori nulli nel dataset e contiamo quanti valori nulli ci sono in ogni colonna
- Ci sono valori nulli?
- Se sì, avrebbe senso cercare di riempirli?
- Eliminiamo le righe che contengono valori nulli

## Esercizio

Dai `beginner_dataset` carichiamo in un DataFrame il file `automobile.csv`:

- Ci sono valori nulli? Dove? Quanti?
- Quali righe hanno un valore nullo nella colonna `num-of-doors`?
- Esaminando i dati nel dataset, cerchiamo una logica per sostituire i valori nulli nella colonna `num-of-doors`

## Esercizio

Abbiamo il seguente DataFrame che raccoglie le misurazioni di un sensore che misura la temperatura atmosferica giornaliera:

```
import numpy as np, pandas as pd  
temp = pd.DataFrame({"Giorno": [0, 1, 2, 3, 4, 5, 6  
                                , 7, 8, 9, 10, 11, 12],  
                      "Temperatura": [18, 19, 18, np.nan, 21, 20, 20,  
                                      np.nan, 21, 23, np.nan, 23, 24] })
```

- Il sensore a volte non funziona, dunque alcuni dati sono mancanti: quale sarebbe la migliore strategia per gestirli?

## Esercizio 1/2

Nel pacchetto `os` della standard library c'è la funzione `os.listdir()` che permette di avere la lista dei nomi di file all'interno di una directory; senza input di default li cerca nella directory di lavoro corrente, altrimenti si può passare un path per esaminare una directory specifica, ad esempio

```
os.listdir("mio_progetto/beginner_datasets/")
```

- Nella directory dei `beginner_datasets`, quali sono i dataset che contengono dati nulli?

## Esercizio 2/2

1. Dovremo usare un ciclo `for` per esaminare tutti i nomi dei file
2. Dovremo selezionare solo i nomi di file con estensione `.csv` (quindi usare un costrutto `if`)
3. Nel corpo dovremo leggere di volta in volta il file in esame, e caricarlo in un `DataFrame` con la funzione `.read_csv()`
4. Sul `DataFrame` dovremo utilizzare il metodo `.isna()` per trovare la maschera booleana dei dati nulli
5. Dovremo contare i dati nulli, utilizzando `.sum()`; potremmo doverlo utilizzare più di una volta
6. Dovremo stampare, o memorizzare in una `list`, solo i nomi dei file che contengono dati nulli



**GRAZIE**  
EPCODE