

```
In [1]: import pandas as pd
```

esercizio

creare e visualizzare la Series di nome *giorni* che contenga i nomi dei giorni della settimana

```
In [ ]:
```

esercizio

creare la serie *a* con i primi 5 numeri dispari e la serie *b* con i primi 5 numeri pari.
effettuare e visualizzare il risultato delle operazioni di somma, sottrazione, moltiplicazione e divisione fra le Series *a* e *b*

```
In [ ]:
```

esercizio

dato il dizionario Python `d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}` creare e visualizzare la Series Pandas di nome *pd1*

```
In [ ]:
```

esercizio

supponendo che le prenotazioni a un ristorante nei giorni della settimana siano le seguenti:

- lunedì 0 (ristorante chiuso)
- martedì 50
- mercoledì 32
- giovedì 25
- venerdì 56
- sabato 78
- domenica 66 realizzare una Series Pandas di nome *ristorante*

```
In [ ]:
```

esercizio

le prenotazioni di giovedì sono state in realtà 44, modificare la Series *ristorante*

```
In [ ]:
```

esercizio

raddoppiare le prenotazioni delle Series *ristorante* poi visualizzare la Series modificata

```
In [ ]:
```

esercizio

visualizzare la Series *ristorante* ordinata in base al numero di prenotazioni

In []:

esercizio

visualizzare il numero medio di prenotazioni giornaliere nel ristorante e la deviazione standard

In []:

esercizio

creare e visualizzare la Series giorni_negativi con i giorni e le prenotazioni in cui si hanno avuto un numero di prenotazioni inferiore alla media

In []:

dati di esempio

```
In [25]: diamonds = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/diamonds.csv')
print(diamonds)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

[53940 rows x 10 columns]

```
In [27]: calcio = pd.read_csv("calcio.csv")
print(calcio)
```

	Div	Date	Time	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	\
0	I1	24/08/2019	17:00	Parma	Juventus	0	1	A	0	
1	I1	24/08/2019	19:45	Fiorentina	Napoli	3	4	A	1	
2	I1	25/08/2019	17:00	Udinese	Milan	1	0	H	0	
3	I1	25/08/2019	19:45	Cagliari	Brescia	0	1	A	0	
4	I1	25/08/2019	19:45	Roma	Genoa	3	3	D	2	
..	
375	I1	02/08/2020	17:00	Spal	Fiorentina	1	3	A	1	
376	I1	02/08/2020	19:45	Bologna	Torino	1	1	D	1	
377	I1	02/08/2020	19:45	Genoa	Verona	3	0	H	3	
378	I1	02/08/2020	19:45	Lecce	Parma	3	4	A	2	
379	I1	02/08/2020	19:45	Sassuolo	Udinese	0	1	A	0	

	HTAG	...	AvgC<2.5	AHCh	B365CAHH	B365CAHA	PCAHH	PCAHA	MaxCAHH	\
0	1	...	2.15	1.50	1.77	2.02	1.81	2.11	1.93	
1	2	...	2.15	0.75	1.75	2.05	1.81	2.11	1.94	
2	0	...	1.78	0.50	1.86	2.07	1.88	2.04	1.91	
3	0	...	1.83	-0.75	2.10	1.83	2.11	1.82	2.12	
4	2	...	2.23	-0.75	1.97	1.96	1.98	1.94	1.99	
..	
375	1	...	2.45	1.25	2.05	1.88	2.03	1.89	2.07	
376	0	...	3.14	-0.75	2.07	1.86	2.03	1.88	2.11	
377	0	...	2.55	-0.75	1.87	2.06	1.99	1.93	2.00	
378	2	...	3.34	-1.00	1.87	2.06	1.87	2.05	1.92	
379	0	...	2.99	-0.50	1.93	2.00	1.92	2.00	1.93	

	MaxCAHA	AvgCAHH	AvgCAHA
0	2.12	1.83	2.04
1	2.14	1.80	2.09
2	2.09	1.85	2.04
3	1.85	2.08	1.82
4	2.20	1.92	1.97
..
375	1.98	1.98	1.89
376	1.88	2.04	1.84
377	2.10	1.87	2.01
378	2.11	1.85	2.02
379	2.11	1.86	2.02

[380 rows x 105 columns]

In []: