

esercizi-6

October 12, 2025

1 Esercizi di Python - Funzioni, Dizionari, Liste, Tuple, Insiemi e Sort

1.1 1. Esercizi su Liste

1.1.1 Esercizio 1.1 - Manipolazione base

Crea una lista con i numeri da 1 a 10. Scrivi del codice per: - Aggiungere il numero 11 alla fine - Inserire il numero 0 all'inizio - Rimuovere il numero 5 - Stampare la lunghezza della lista finale

1.1.2 Esercizio 1.2 - Operazioni con liste

Data la lista `numeri = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3]`: - Trova la somma di tutti gli elementi - Trova il valore massimo e minimo - Conta quante volte appare il numero 1 - Crea una nuova lista con solo i numeri pari

1.1.3 Esercizio 1.3 - Liste annidate

Crea una lista di liste che rappresenti una matrice 3x3:

```
1 2 3
4 5 6
7 8 9
```

Scrivi codice per stampare la diagonale principale (1, 5, 9).

1.2 2. Esercizi su Dizionari

1.2.1 Esercizio 2.1 - Conteggio caratteri

Scrivi una funzione che prenda una stringa e restituisca un dizionario con il conteggio di ogni carattere.

```
[ ]: def conta_caratteri(testo):
    # Il tuo codice qui
    pass

# Test
print(conta_caratteri("hello world"))
# Output atteso: {'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1}
↪ 1}
```

1.2.2 Esercizio 2.2 - Unione dizionari

Dati due dizionari, scrivere codice per unirli. Se una chiave è presente in entrambi, somma i valori.

```
[ ]: dict1 = {'a': 1, 'b': 2, 'c': 3}
dict2 = {'b': 3, 'c': 4, 'd': 5}
# Il tuo codice qui
# Risultato atteso: {'a': 1, 'b': 5, 'c': 7, 'd': 5}
```

1.3 3. Esercizi su Tuple

1.3.1 Esercizio 3.1 - Coordinate

Crea una lista di tuple che rappresentino coordinate (x, y): - Crea 5 punti casuali - Trova il punto più lontano dall'origine (0, 0) - Ordina i punti per distanza dall'origine

```
[ ]: import math
# Il tuo codice qui
```

1.3.2 Esercizio 3.2 - Swap di variabili

Usando le tuple, scrivi codice per scambiare i valori di due variabili senza usare una variabile temporanea.

```
[ ]: a = 10
b = 20
# Il tuo codice qui per scambiare a e b
print(f"a = {a}, b = {b}") # Dovrebbe stampare: a = 20, b = 10
```

1.3.3 Esercizio 3.3 - Ritorno multiplo

Scrivi una funzione che prenda una lista di numeri e restituisca una tupla con (minimo, massimo, media).

```
[ ]: def statistiche(numeri):
    # Il tuo codice qui
    pass

# Test
print(statistiche([1, 5, 3, 9, 2])) # (1, 9, 4.0)
```

1.4 4. Esercizi su Insiemi (Set)

1.4.1 Esercizio 4.1 - Operazioni base

Dati due insiemi di numeri: - set1 = {1, 2, 3, 4, 5} - set2 = {4, 5, 6, 7, 8}

Calcola: - Unione - Intersezione

- Differenza (elementi in set1 ma non in set2) - Differenza simmetrica

```
[ ]: set1 = {1, 2, 3, 4, 5}
      set2 = {4, 5, 6, 7, 8}
      # Il tuo codice qui
```

1.4.2 Esercizio 4.2 - Rimozione duplicati

Scrivi una funzione che prenda una lista e restituisca una nuova lista senza duplicati, mantenendo l'ordine.

```
[ ]: def rimuovi_duplicati(lista):
      # Il tuo codice qui
      pass

      # Test
      print(rimuovi_duplicati([1, 2, 2, 3, 1, 4, 5, 4])) # [1, 2, 3, 4, 5]
```

1.4.3 Esercizio 4.3 - Lettere uniche

Scrivi una funzione che prenda due parole e restituisca le lettere che appaiono in entrambe.

```
[ ]: def lettere_comuni(parola1, parola2):
      # Il tuo codice qui
      pass

      # Test
      print(lettere_comuni("python", "java")) # {'a'}
```

1.5 5. Esercizi su Sort (Ordinamento)

1.5.1 Esercizio 5.1 - Ordinamento personalizzato

Data una lista di nomi, ordinali: - In ordine alfabetico - Per lunghezza (dal più corto al più lungo)
- Per lunghezza inversa (dal più lungo al più corto)

```
[ ]: nomi = ["Alice", "Bob", "Charlie", "Diana", "Eve"]
      # Il tuo codice qui
```

1.5.2 Esercizio 5.2 - Ordinamento di dizionari

Data una lista di dizionari rappresentanti persone, ordinali per età crescente.

```
[ ]: persone = [
      {'nome': 'Alice', 'eta': 25},
      {'nome': 'Bob', 'eta': 30},
      {'nome': 'Charlie', 'eta': 20}
      ]
      # Il tuo codice qui
```

1.5.3 Esercizio 5.3 - Ordinamento complesso

Data una lista di tuple (nome, voto), ordina prima per voto decrescente, poi per nome alfabetico in caso di parità.

```
[ ]: studenti = [('Alice', 85), ('Bob', 92), ('Charlie', 85), ('Diana', 78)]
      # Il tuo codice qui
```

1.6 6. Esercizi su Funzioni

1.6.1 Esercizio 6.1 - Funzione con parametri opzionali

Scrivi una funzione che calcoli l'area di un rettangolo. Se viene passato solo un parametro, considera il rettangolo come un quadrato.

```
[ ]: def area_rettangolo(larghezza, altezza=None):
      # Il tuo codice qui
      pass

      # Test
      print(area_rettangolo(5))      # 25 (quadrato)
      print(area_rettangolo(4, 6))  # 24 (rettangolo)
```

1.6.2 Esercizio 6.2 - Elaborazione dati studenti

Scrivi un programma che gestisca i voti degli studenti:

Suggerimento: Spezza il programma in funzioni piccole.

```
[ ]: def elabora_voti(dati_studenti):
      """
      Prende una lista di tuple (nome, [lista_voti]) e restituisce:
      - Dizionario con nome -> media voti
      - Lista studenti ordinata per media decrescente
      - Set degli studenti che hanno almeno un voto >= 9
      - Statistiche globali (media classe, voto max, voto min)
      """
      # Il tuo codice qui
      pass

      # Test
      studenti = [
          ("Alice", [8, 7, 9, 6]),
          ("Bob", [9, 8, 10, 7]),
          ("Charlie", [6, 5, 7, 8]),
          ("Diana", [10, 9, 9, 8])
      ]

      risultati = elabora_voti(studenti)
      print(risultati)
```