

## 9 – Map Function

La funzione `map()` in Python è uno strumento potente per applicare una **funzione a tutti gli elementi di un iterabile** (come liste, tuple o set) restituendo un nuovo iterabile con i risultati. È molto utile per trasformazioni rapide senza dover usare un ciclo `for` esplicito.

# Sintassi

```
map(funzione, iterabile)
```

- **funzione**: una funzione da applicare a ciascun elemento. Può essere definita con `def` o con `lambda`.
- **iterabile**: la sequenza su cui applicare la funzione.
- Restituisce un oggetto `map`, che può essere convertito in lista, tupla o set.

## Esempio base

```
numeri = [1, 2, 3, 4, 5]

# raddoppiare ogni numero
risultato = map(lambda x: x * 2, numeri)
print(list(risultato))  # [2, 4, 6, 8, 10]
```

## Usare map con una funzione definita

```
def quadrato(x):  
    return x ** 2  
  
numeri = [1, 2, 3, 4]  
quadrati = map(quadrato, numeri)  
print(list(quadrati))  # [1, 4, 9, 16]
```

## Map con più iterabili

`map()` può prendere più iterabili della stessa lunghezza; la funzione deve accettare tanti argomenti quanti gli iterabili:

```
a = [1, 2, 3]
b = [4, 5, 6]

somma = map(lambda x, y: x + y, a, b)
print(list(somma))  # [5, 7, 9]
```

## Vantaggi di map()

- Codice più conciso e leggibile rispetto a un ciclo `for` .
- Funziona con qualsiasi iterabile.
- Facilmente combinabile con lambda per trasformazioni rapide.

## Nota

`map()` restituisce un oggetto iterabile; per ottenere una lista o una tupla, bisogna convertirlo esplicitamente con `list()` o `tuple()` .

Vuoi procedere con il **paragrafo 10 – Filter Function?**