

12 – Zip Function

La funzione `zip()` in Python permette di **combinare più iterabili** elemento per elemento, creando una sequenza di tuple. È molto utile quando si vogliono **iterare più liste contemporaneamente** o unire dati correlati.

Sintassi

```
zip(iterabile1, iterabile2, ...)
```

- Gli iterabili possono essere liste, tuple, stringhe o qualsiasi oggetto iterabile.
- La funzione restituisce un oggetto zip, che può essere convertito in lista, tupla o set.
- L'iterazione termina quando **il più corto degli iterabili finisce**.

Esempio base

Unire due liste di uguale lunghezza:

```
nomi = ["Anna", "Luca", "Marco"]
eta = [25, 30, 22]

combinati = zip(nomi, eta)
print(list(combinati)) # [('Anna', 25), ('Luca', 30), ('Marco', 22)]
```

Iterazione diretta

Puoi usare `zip()` direttamente in un ciclo `for`:

```
for nome, anni in zip(nomi, eta):  
    print(f"{nome} ha {anni} anni")
```

Output:

```
Anna ha 25 anni  
Luca ha 30 anni  
Marco ha 22 anni
```

Più iterabili

Puoi combinare anche più di due iterabili:

```
citta = ["Roma", "Milano", "Napoli"]
dati = list(zip(nomi, eta, citta))
print(dati) # [('Anna', 25, 'Roma'), ('Luca', 30, 'Milano'), ('Marco', 22, 'Napoli')]
```

Zip con iterabili di lunghezza diversa

`zip()` si ferma al più corto:

```
a = [1, 2, 3]
b = ['a', 'b']
```

```
print(list(zip(a, b))) # [(1, 'a'), (2, 'b')]
```

Se vuoi mantenere anche gli elementi mancanti, puoi usare

`itertools.zip_longest()`.

Scompattare con zip

`zip()` può essere usato anche per “invertire” una lista di tuple:

```
dati = [('Anna', 25), ('Luca', 30), ('Marco', 22)]
nomi, eta = zip(*dati)
print(nomi) # ('Anna', 'Luca', 'Marco')
print(eta) # (25, 30, 22)
```

Vantaggi

- Permette di iterare più sequenze contemporaneamente.
- Utile per creare coppie o tuple da liste correlate.
- Combinabile con unpacking per separare nuovamente i dati.

Vuoi procedere con il **paragrafo 13 – Stacks**?