

22 – Unpacking Operator

L'**unpacking operator** (`*` per liste/tuple e `**` per dizionari) permette di **espandere gli elementi di un iterabile** o le coppie chiave-valore di un dizionario in contesti come assegnazioni, chiamate di funzione o concatenazioni. È uno strumento potente per scrivere codice più pulito e flessibile.

Unpacking di liste o tuple

```
numeri = [1, 2, 3]
a, *b = numeri
print(a) # 1
print(b) # [2, 3]
```

- `*b` raccoglie il resto degli elementi.
- Funziona anche in posizione centrale o finale.

```
*a, b = numeri
print(a) # [1, 2]
print(b) # 3
```

Unpacking in funzioni

Permette di passare elementi di una lista come argomenti:

```
def somma(a, b, c):  
    return a + b + c  
  
numeri = [1, 2, 3]  
print(somma(*numeri)) # 6
```

Concatenare liste

```
lista1 = [1, 2]
lista2 = [3, 4]
unita = [*lista1, *lista2]
print(unita) # [1, 2, 3, 4]
```

Unpacking dei dizionari

Con `**` puoi unire più dizionari o passare coppie chiave-valore a una funzione:

```
d1 = {"a": 1, "b": 2}
d2 = {"c": 3, "d": 4}

d_unito = {**d1, **d2}
print(d_unito) # {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

```
def stampa_info(nome, età):
    print(f"{nome} ha {età} anni")

info = {"nome": "Anna", "età": 25}
stampa_info(**info) # Anna ha 25 anni
```

Vantaggi

- Permette di **espandere iterabili** o dizionari senza cicli manuali.
- Utile per concatenazioni, assegnazioni multiple e chiamate di funzioni.
- Rende il codice più **pulito e leggibile**.

Vuoi procedere con il **paragrafo 23 – Exercise?**