

20 – Dictionary Comprehensions

Le **dictionary comprehensions** permettono di creare dizionari in modo compatto e leggibile, analogamente alle list comprehension. Sono utili quando vuoi **costruire un dizionario trasformando o filtrando dati** in un'unica espressione.

Sintassi base

```
{chiave: valore for elemento in iterabile}
```

- chiave : espressione che genera la chiave del dizionario.
- valore : espressione che genera il valore corrispondente.
- iterabile : lista, tupla, range o altro iterabile da cui derivare chiavi e valori.

Esempio semplice

Creare un dizionario con numeri e loro quadrati:

```
numeri = [1, 2, 3, 4]
quadrati = {x: x**2 for x in numeri}
print(quadrati) # {1: 1, 2: 4, 3: 9, 4: 16}
```

Con condizione

Puoi filtrare gli elementi usando `if`:

```
numeri = range(10)
pari_quadrati = {x: x**2 for x in numeri if x % 2 == 0}
print(parì_quadrati) # {0: 0, 2: 4, 4: 16, 6: 36, 8: 64}
```

Da due liste

Se hai due liste correlate, puoi combinarle in un dizionario:

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

dizionario = {n: e for n, e in zip(nomi, eta)}
print(dizionario) # {'Anna': 25, 'Luca': 30, 'Marco': 22}
```

Trasformazioni complesse

Puoi applicare funzioni o espressioni ai valori:

```
parole = ["ciao", "mondo", "python"]
lunghezze = {p: len(p) for p in parole}
print(lunghezze) # {'ciao': 4, 'mondo': 5, 'python': 6}
```

Vantaggi

- Codice più compatto e leggibile.
- Permette di filtrare e trasformare dati in un'unica espressione.
- Ideale per generare dizionari da liste, tuple o altre strutture iterabili.

Vuoi procedere con il **paragrafo 21 – Generator Expressions?**