

19 – Dictionaries

Un **dictionary** (o dizionario) in Python è una collezione **non ordinata di coppie chiave-valore**, dove ogni chiave è **unica**. I dizionari sono estremamente utili per memorizzare e accedere rapidamente a dati strutturati tramite **chiavi descrittive** anziché indici numerici.

Creazione di un dictionary

Si possono creare con parentesi graffe `{}`:

```
studente = {  
    "nome": "Anna",  
    "età": 20,  
    "corso": "Informatica"  
}  
print(studente)
```

Oppure usando la funzione `dict()`:

```
studente = dict(nome="Luca", età=22, corso="Matematica")
```

Accesso ai valori

I valori si accedono tramite la chiave:

```
print(studente["nome"]) # Anna
```

- Usare `get()` evita errori se la chiave non esiste:

```
print(studente.get("città", "Non definita")) # Non definita
```

Aggiungere o modificare valori

```
studente["età"] = 21          # modifica  
studente["città"] = "Roma"    # aggiunge nuova chiave
```

Rimuovere elementi

```
studente.pop("corso")    # rimuove la chiave "corso"  
studente.popitem()       # rimuove l'ultima coppia inserita  
del studente["età"]      # rimuove la chiave specificata  
studente.clear()         # cancella tutto il dictionary
```

Iterare sui dizionari

```
studente = {"nome": "Anna", "età": 20, "corso": "Informatica"}

# chiavi
for chiave in studente:
    print(chiave)

# valori
for valore in studente.values():
    print(valore)

# coppie chiave-valore
for chiave, valore in studente.items():
    print(f"{chiave}: {valore}")
```

Verifica appartenenza

```
print("nome" in studente) # True  
print("città" not in studente) # True
```

Vantaggi

- Accesso rapido ai dati tramite chiavi.
- Ideali per strutturare informazioni eterogenee.
- Possono contenere qualsiasi tipo di oggetto come valore, inclusi liste, tuple o altri dizionari.

Vuoi procedere con il **paragrafo 20 – Dictionary Comprehensions?**