

Dizionari e set

Cos'è un dizionario?

Pensa a un dizionario vero: cerchi una **parola** (la chiave) e trovi la sua **definizione** (il valore). Ecco, i dizionari di Python funzionano esattamente così! Sono collezioni di coppie **chiave** → **valore**. Tipo la rubrica del telefono: cerchi il nome, trovi il numero.

```
# Dizionario: chiave → valore
studente = {
    "nome": "Mario",
    "cognome": "Rossi",
    "eta": 16,
    "media": 7.5
}

print(studente)
print(studente["nome"])
print(studente["media"])
```

Creare un dizionario

Ci sono vari modi per creare un dizionario. Scegli quello che ti piace di più:

```
# Dizionario vuoto (pronto per essere riempito)
vuoto = {}

# Con valori iniziali
colori = {
    "rosso": "#FF0000",
    "verde": "#00FF00",
    "blu": "#0000FF"
}

# Con la funzione dict() (un po' meno usata)
persona = dict(nome="Luigi", eta=20)

print(colori)
print(persona)
```

Accesso e modifica

Accedere ai valori è semplicissimo: usa la chiave come indice. Tipo cercare una parola nell'enciclopedia, ma istantaneo!

```

rubrica = {
    "Mario": "333-1234567",
    "Luigi": "333-7654321",
    "Peach": "333-1111111"
}

# Accesso
print(rubrica["Mario"])

# Modifica (Mario ha cambiato numero!)
rubrica["Mario"] = "333-9999999"
print("Mario aggiornato:", rubrica["Mario"])

# Aggiungere un nuovo contatto? Basta assegnare!
rubrica["Toad"] = "333-0000000"
print("Rubrica:", rubrica)

```

Accesso sicuro con `get()`

!!! danger "Attenzione alle chiavi inesistenti!"

Se cerchi una chiave che non esiste con `rubrica["Bowser"]`, Python ti l

```

rubrica = {"Mario": "333-1234567", "Luigi": "333-7654321"}

# Con [] se la chiave non esiste → BOOM! KeyError!
# print(rubrica["Peach"]) # Non farlo!

# Con get() → niente drammi, restituisce None
print(rubrica.get("Peach"))

# Oppure un valore di default a tua scelta
print(rubrica.get("Peach", "Non trovato"))
print(rubrica.get("Mario", "Non trovato"))

```

Rimuovere elementi

Due modi principali per fare pulizia nel dizionario:

```
voti = {"matematica": 8, "italiano": 7, "inglese": 9, "storia": 6}

# del – cancella senza pietà
del voti["storia"]
print("Dopo del:", voti)

# pop() – cancella e ti dice cosa ha tolto (tipo un ricevuta!)
voto_inglese = voti.pop("inglese")
print(f"inglese rimosso: {voto_inglese}")
print("Dopo pop:", voti)
```

Iterare su un dizionario

Scorrere un dizionario è comodissimo. Hai tre opzioni a seconda di cosa ti serve:

```
voti = {"matematica": 8, "italiano": 7, "inglese": 9, "storia": 6}

# Solo le chiavi (il default)
print("Materie:")
for materia in voti:
    print(f"  {materia}")

# Solo i valori
print(
Voti:")
for voto in voti.values():
    print(f"  {voto}")

# Chiavi E valori insieme (il più utile!)
print(
Pagella:")
for materia, voto in voti.items():
    print(f"  {materia}: {voto}")
```

Metodi utili

Il dizionario ha un sacco di metodi comodi. Ecco i più usati:

```
persona = {"nome": "Mario", "eta": 16, "citta": "Roma"}

# Chiavi, valori, coppie
print("Chiavi:", list(persona.keys()))
print("Valori:", list(persona.values()))
print("Coppie:", list(persona.items()))

# Verificare se una chiave esiste (utilissimo!)
print("'nome' esiste?", "nome" in persona)
print("'peso' esiste?", "peso" in persona)

# Numero di coppie
print("Elementi:", len(persona))
```

Dictionary comprehension

Come per le liste, puoi creare dizionari in modo compatto con una sola riga. Stessa magia, diverso contenitore:

```
# Tabella dei quadrati
quadrati = {n: n**2 for n in range(1, 6)}
print(quadrati)

# Filtrare un dizionario (solo i sufficienti!)
voti = {"matematica": 8, "italiano": 5, "inglese": 9, "storia": 4}
sufficienti = {m: v for m, v in voti.items() if v >= 6}
print("Sufficienti:", sufficienti)
```

I Set

Un set è una **collezione di elementi unici senza ordine**. Tipo un club esclusivo: niente duplicati ammessi! Si crea con le parentesi graffe `{}` :

```
# Set: elimina automaticamente i duplicati
numeri = {1, 3, 5, 3, 7, 1, 9, 5}
print(numeri) # I duplicati? Spariti! Puf!

# Creare un set da una lista (modo veloce per rimuovere duplicati)
lista = [1, 2, 2, 3, 3, 3, 4]
insieme = set(lista)
print(insieme)
```

Operazioni tra set

I set supportano le **operazioni insiemistiche** che hai studiato in matematica. Finalmente servono a qualcosa!

```

classe_a = {"Mario", "Luigi", "Peach", "Toad"}
classe_b = {"Luigi", "Yoshi", "Peach", "Daisy"}

# Unione: tutti gli studenti (senza duplicati)
print("Unione:", classe_a | classe_b)

# Intersezione: chi c'è in entrambe le classi?
print("In comune:", classe_a & classe_b)

# Differenza: solo in classe A
print("Solo A:", classe_a - classe_b)

# Differenza simmetrica: in una o nell'altra, ma NON in entrambe
print("Esclusivi:", classe_a ^ classe_b)

```

Quando usare cosa?

Hai un sacco di strutture dati, ecco una guida rapida per scegliere quella giusta:

Struttura	Uso tipico	Esempio
Lista	Collezione ordinata, con duplicati	Voti di uno studente
Tupla	Dati fissi, non modificabili	Coordinate (x, y)
Dizionario	Associare chiavi a valori	Rubrica, pagella
Set	Elementi unici, operazioni insiemistiche	Lista presenze senza duplicati

Esercizi

Esercizio 1: Conta parole

Conta quante volte appare ogni parola in una frase. Praticamente stai costruendo un mini motore di ricerca!

```
frase = "il gatto e il cane e il pesce"  
parole = frase.split()  
  
# Crea un dizionario con il conteggio di ogni parola
```

Esercizio 2: Rubrica

Crea una rubrica dove l'utente può aggiungere, cercare e visualizzare contatti. La tua prima app utile!

```
rubrica = {}  
  
# Aggiungi 3 contatti  
for i in range(3):  
    nome = input(f"Nome contatto {i+1}: ")  
    tel = input(f"Telefono di {nome}: ")  
    rubrica[nome] = tel  
  
# Stampa la rubrica  
print()  
Rubrica completa:  
for nome, tel in rubrica.items():  
    print(f"  {nome}: {tel}")
```