

Dizionari

In italiano **dizionari**. I dizionari contengono elementi organizzati in coppie **chiave:valore**.

Caratteristiche:

- Ordinati: a partire da Python 3.7, i dizionari sono ordinati, il che significa che mantengono l'ordine in cui gli elementi sono stati aggiunti.

Nelle versioni precedenti (Python 3.6 e precedenti), i dizionari erano non ordinati.

- Modificabili: i dizionari sono modificabili, quindi puoi cambiare, aggiungere o rimuovere elementi dopo averli creati.
- Duplicati non consentiti: non è possibile avere due elementi con la stessa chiave in un dizionario, ma i valori possono essere duplicati.
- Tipi di dato: i valori possono essere di tutti i tipi di dato che abbiamo visto, possono anche contenere altri dizionari.

```
In [ ]: !python --version

Python 3.10.16

In [30]: # Creazione di un dizionario
film = {
    "titolo": "Bastardi senza gloria",
    "regista": "Quentin Tarantino",
    "anno": 2009,
    "genere": ["Guerra", "Drammatico", "Azione"],
}
print(film)

{'titolo': 'Bastardi senza gloria', 'regista': 'Quentin Tarantino', 'anno': 2009, 'genere': ['Guerra', 'Drammatico', 'Azione']}
```

```
In [ ]: # Duplicati non consentiti: le chiavi devono essere uniche
film2 = {
    "titolo": "Oppenheimer",
    "titolo": "Interstellar" # Sovrascrive il valore precedente
}
print(film2)

{'titolo': 'Interstellar'}
```

Dizionari innestati

Un dizionario può contenere dizionari: cosa figa e molto potente!

```
In [ ]: # Creazione di un dizionario annidato
studenti = {
    "studente1": {
        "nome": "Alice",
        "età": 18,
        "corsi": ["Matematica", "Fisica"]
    },
    "studente2": {
        "nome": "Marco",
        "età": 17,
        "corsi": ["Storia", "Inglese"]
    },
    "studente3": {
        "nome": "Sara",
        "età": 19,
        "corsi": ["Chimica", "Biologia"]
    }
}
```

Metodi

Accesso

Per accedere ai vari elementi di un dizionari ci sono vari modi e metodi. Si può utilizzare una **chiave** accedendo al valore riferito ad essa, oppure con il metodo **get()**.

Il metodo **get()** è importante e molto utile perché permette di specificare un valore di ritorno se la chiave specificata non esiste.

dictionary.get(keyname, value)

```
In [ ]: # Riferendosi al nome della chiave
film['titolo']

'Bastardi senza gloria'

In [16]: # Utilizzando il metodo get()
film.get('titolo')

Out[16]: 'Bastardi senza gloria'

In [32]: # Get() con valore di ritorno non default
film.get('valutazione', 0)

Out[32]: 0

In [ ]: # Accedere ad elementi innestati
studenti['studente3']['nome']

'Sara'
```

Modifica

I dizionari si possono modificare grazie all'utilizzo delle loro **chiavi** o con il metodo **update()**.

```
In [34]: # Riferendosi al nome della chiave
print(film)
film['valutazione'] = 9.0 # Aggiunta la chiave 'valutazione'
print(film)
film['valutazione'] = 8.9 # Modifica chiave 'valutazione'
print(film)

{'titolo': 'Bastardi senza gloria', 'regista': 'Quentin Tarantino', 'anno': 2009, 'genere': ['Guerra', 'Drammatico', 'Azione']}
{'titolo': 'Bastardi senza gloria', 'regista': 'Quentin Tarantino', 'anno': 2009, 'genere': ['Guerra', 'Drammatico', 'Azione'], 'valutazione': 9.0}
{'titolo': 'Bastardi senza gloria', 'regista': 'Quentin Tarantino', 'anno': 2009, 'genere': ['Guerra', 'Drammatico', 'Azione'], 'valutazione': 8.9}

In [ ]: # Utilizzando il metodo update()
macchina = {
    "marca": "Ford",
    "modello": "Mustang",
    "anno": 1964
}
macchina.update({"anno": 2020})
print(macchina)

{'marca': 'Ford', 'modello': 'Mustang', 'anno': 2020}
```

Aggiungere un elemento

Sempre grazie all'utilizzo della chiave e del metodo **update()** possiamo aggiungere una coppia **chiave:valore** al dizionario.

```
In [ ]: # Utilizzando una chiave
macchina['targa'] = "FE587GB"
print(macchina)

{'marca': 'Ford', 'modello': 'Mustang', 'anno': 2020, 'colore': 'nero', 'targa': 'FE587GB'}

In [ ]: # Utilizzando il metodo update()
macchina.update({"colore": "nero"})
print(macchina)

{'marca': 'Ford', 'modello': 'Mustang', 'anno': 2020, 'colore': 'nero', 'targa': 'FE587GB'}
```

Rimuovere un elemento

Si possono rimuovere elementi (coppie chiave:valore) dai dizionari attraverso i metodi.

Metodo	Descrizione
pop()	Rimuove l'elemento con la chiave passata come parametro, restituisce errore se non esiste la chiave
popitem()	Rimuove l'elemento inserito per ultimo
clear()	Rimuove tutti gli elementi

```
In [ ]: macchina.pop("colore")

'nero'

In [ ]: macchina.popitem()

('targa', 'FE587GB')

In [ ]: macchina.clear()
print(macchina)

{}
```

Miscellanea

Metodo	Descrizione
items()	Restituisce una lista contenente tutti gli elementi (chiave:valore) del dizionario
keys()	Restituisce una lista contenente tutte le chiavi del dizionario
values()	Restituisce una lista contenente tutti i valori all'interno del dizionario

Funzione len(): non è un metodo ma è una funzione; permette di stampare il numero di elementi nel dizionario.

```
In [ ]: print(film)

{'titolo': 'Bastardi senza gloria', 'regista': 'Quentin Tarantino', 'anno': 2009, 'genere': ['Guerra', 'Drammatico', 'Azione'], 'valutazione': 9.0}

In [ ]: film.items()

dict_items([('titolo', 'Bastardi senza gloria'), ('regista', 'Quentin Tarantino'), ('anno', 2009), ('genere', ['Guerra', 'Drammatico', 'Azione']), ('valutazione', 9.0)])

In [ ]: film.keys()

dict_keys(['titolo', 'regista', 'anno', 'genere', 'valutazione'])

In [ ]: film.values()

dict_values(['Bastardi senza gloria', 'Quentin Tarantino', 2009, ['Guerra', 'Drammatico', 'Azione'], 9.0])

In [ ]: len(film)

5
```

Esercizio

Data una serie di amici sottoforma di dizionario in cui un amico ha nome e età:

- Restituire l'età dell'amico1
- Restituire il nome del terzo amico (usando le chiavi)
- Restituire il nome del terzo amico (usando get())
- Cancellare amico6 e amico7 con popitem()
- Cancellare amico6 e amico7 con pop()
- Controllare se amico4 è maggiorenne
- Chiedere all'utente di quale amico vuole sapere il nome (aspettarsi un intero) e rispondere all'utente
- Contare quanti amici hanno più di 20 anni

```
In [20]: amici = {
    "amico1": {
        "nome": "Beatrice",
        "età": 16,
    },
    "amico2": {
        "nome": "Marco",
        "età": 21,
    },
    "amico3": {
        "nome": "Mattia",
        "età": 19,
    },
    "amico4": {
        "nome": "Stefano",
        "età": 21,
    },
    "amico5": {
        "nome": "Sara",
        "età": 24,
    },
    "amico6": {
        "nome": "Gaia",
        "età": 23,
    },
    "amico7": {
        "nome": "Elena",
        "età": 20,
    }
}

In [ ]: # 1
amici['amico1']['età']

16

In [ ]: # 2a
amici['amico3']['nome']

'Mattia'

In [ ]: # 2b
amici['amico3'].get('nome')

'Mattia'

In [12]: # 3a
print(amici.popitem())
print(amici.popitem())

('amico7', {'nome': 'Elena', 'età': 20})
('amico6', {'nome': 'Gaia', 'età': 23})

In [16]: # 3b
print(amici.pop("amico6"))
print(amici.pop("amico7"))

{'nome': 'Gaia', 'età': 23}
{'nome': 'Elena', 'età': 20}

In [ ]: # 4
if amici['amico4']['età'] >= 18:
    print(amici['amico4']['nome'] + " è maggiorenne!")
else:
    print(amici['amico4']['nome'] + " è minorenne!")

Stefano è maggiorenne!

In [ ]: # 5
numero_amico = input("Di quale amico vuoi sapere il nome?")
amici[numero_amico+ numero_amico]['nome']
# Oppure
amici[f'amico{numero_amico}']['nome']

Di quale amico vuoi sapere il nome? 3
'Mattia'

In [45]: # 6a (senza ciclo for)
maggiorenni = 0

if amici["amico1"]["età"] >= 18:
    maggiorenni += 1
if amici["amico2"]["età"] >= 18:
    maggiorenni += 1
if amici["amico3"]["età"] >= 18:
    maggiorenni += 1
if amici["amico4"]["età"] >= 18:
    maggiorenni += 1
if amici["amico5"]["età"] >= 18:
    maggiorenni += 1
if amici["amico6"]["età"] >= 18:
    maggiorenni += 1
if amici["amico7"]["età"] >= 18:
    maggiorenni += 1

print(f"Il numero di amici maggiorenni è: {maggiorenni}")

Il numero di amici maggiorenni è: 6
```

Dizionari e cicli for

```
In [47]: # 6b (con ciclo for)
maggiorenni = 0

for amico in amici:
    if amici[amico]['età'] >= 18:
        maggiorenni += 1

print(f"Il numero di amici maggiorenni è: {maggiorenni}")

Il numero di amici maggiorenni è: 6

In [51]: # 6c (con ciclo for)
maggiorenni = 0

for numero, amico in amici.items():
    if amico['età'] >= 18:
        maggiorenni += 1

print(f"Il numero di amici maggiorenni è: {maggiorenni}")

Il numero di amici maggiorenni è: 6
```