

Programmazione in Python

strutture dati: dizionari

Dario Pescini - Mirko Cesarini

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi

`nome.cognome@unimib.it`

Strutture dati complesse: dizionari



dizionario

possiede un nome **d** ed aggrega più oggetti, identificabili tramite una *chiave*, in una *mappa*:

```
d = a 'pippo' 6 d ... z
```

Strutture dati complesse: dizionari



dizionario

possiede un nome **d** ed aggrega più oggetti, identificabili tramite una **chiave**, in una **mappa**:

d = **d**[**chiave**₁] **d**[**chiave**₂] **d**[**chiave**₃] ... **d**[**chiave**_n]

Strutture dati complesse: dizionari



dizionario

possiede un nome **d** ed aggrega più oggetti, identificabili tramite una **chiave**, in una **mappa**:

$$d = \{ (\text{chiave}_1, \text{valore}_1), (\text{chiave}_2, \text{valore}_2), (\text{chiave}_3, \text{valore}_3), \dots, (\text{chiave}_n, \text{valore}_n) \}$$

Strutture dati complesse: dizionari

Il dizionario è una struttura dati **complessa** di tipo **mappa**, **dinamica** ed **eterogenea** i cui elementi vengono identificati tramite una **chiave**.

```
diz = { 'a': 7, 'b': 3.0 + 5, 'c': 'pippo', 'd': 2 + 1j }
```

dichiarazione dizionario: { }

- **diz** nome del dizionario
- { } delimitatori del dizionario
- 'a': 7, 'b': 3.0 + 5, ... coppie chiave : valore contenute nel dizionario
- , separatore delle coppie

Strutture dati complesse: dizionari

diz

7	8.0	pippo	2+1j
---	-----	-------	------

a b c d

insieme di valori:

```
>>> diz = {'a': 7, 'b': 8.0, 'c': 'pippo', 'd': 2+1j}
>>> print diz
{'a': 7, 'c': 'pippo', 'b': 8.0, 'd': (2+1j)}
>>> print diz[2]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 2
>>> print diz['b']
8.0
>>> _
```

Strutture dati complesse: dizionari

diz

7	8.0	pippo	2+1j
---	-----	-------	------

a b c d

lunghezza:

```
>>> len(diz)
4
>>> _
```

Strutture dati complesse: dizionari

diz

7	8.0	pippo	2+1j
a	b	c	d

!! mutabile: !!

```
>>> diz['c'] = 'nuovo'  
>>> print diz  
{'a': 7, 'c': 'nuovo', 'b': 8.0, 'd': (2+1j)}  
>>> _
```


Metodi per i dizionari

Esistono due metodi principali che potete utilizzare per i dizionari:

- `keys()` restituisce la lista delle chiavi contenute nel dizionario
- `values()` restituisce la lista dei valori contenuti nel dizionario

Metodi: esempi

```
diz = { 'a': 7, 'b': 3.0 + 5, 'c': 'pippo', 'd': 2 +1j }
```

- `diz.keys()` → ['a', 'c', 'b', 'd']

Metodi: esempi

```
diz = { 'a': 7, 'b': 3.0 + 5, 'c': 'pippo', 'd': 2 +1j }
```

- `diz.keys()` → ['a', 'c', 'b', 'd']
- `diz.values()` → [7, 'pippo', 8.0, (2+1j)]