

## I DIZIONARI

Le liste sono un esempio di tipo di dati composti che possono usare solo numeri interi come indici. I dizionari sono simili agli altri tipi composti ma si differenziano per il fatto di poter usare qualsiasi tipo di dato immutabile come indice. Se desideriamo creare un dizionario per la traduzione di parole dall'inglese all'italiano è utile poter usare la parola inglese come indice di ricerca della corrispondente italiana.

Gli indici usati sono in questo caso delle stringhe.

Un modo per creare un dizionario è partire con un dizionario vuoto e aggiungere via via gli elementi. Il dizionario vuoto è indicato da {}.

Apri python in modalità interattiva e scrivi le tre righe seguenti (alla fine di ogni riga premi <invio>):

```
>>> Eng2Ita = {}
>>> Eng2Ita['one'] = 'uno'
>>> Eng2Ita['two'] = 'due'
```

La prima assegnazione crea un dizionario chiamato Eng2Ita; le altre istruzioni aggiungono elementi al dizionario. Possiamo stampare il valore del dizionario nel solito modo:

```
>>> print Eng2Ita
{'one': 'uno', 'two': 'due'}
```

Gli elementi di un dizionario appaiono in una sequenza separata da virgole. Ogni **voce** contiene un **indice** ed il corrispondente **valore** separati da due punti.

In un dizionario gli indici sono chiamati **chiavi** e un elemento è detto **coppia chiave-valore**.

Un altro modo di creare un dizionario è quello di fornire direttamente una serie di coppie chiave-valore:

```
>>> Eng2Ita = {'one': 'uno', 'two': 'due', 'three': 'tre'}
```

Se stampiamo ancora una volta il valore di Eng2Ita abbiamo una sorpresa:

```
>>> print Eng2Ita
{'one': 'uno', 'three': 'tre', 'two': 'due'}
```

Le coppie chiave-valore non sono in ordine! Per fortuna non c'è ragione di conservare l'ordine di inserimento dato che il dizionario non fa uso di indici numerici. Per cercare un valore usiamo infatti una chiave:

```
>>> print Eng2Ita['two']
'due'
```

La chiave 'two' produce correttamente 'due' anche se appare in terza posizione nella stampa del dizionario.

## OPERAZIONI SUI DIZIONARI

L'istruzione **del** rimuove una coppia chiave-valore da un dizionario. Vediamo di fare un esempio pratico creando un dizionario che contiene il nome di vari tipi di frutta (la chiave) ed il numero di frutti corrispondenti in magazzino (il valore):

```
>>> Magazzino = {'mele': 430, 'banane': 312, 'arance': 525, 'pere': 217}
>>> print Magazzino
{'banane': 312, 'arance': 525, 'pere': 217, 'mele': 430}
```

Dovessimo togliere la scorta di pere dal magazzino possiamo direttamente rimuovere la voce dal dizionario:

```
>>> del Magazzino['pere']
>>> print Magazzino
{'banane': 312, 'arance': 525, 'mele': 430}
```

o se intendiamo solo cambiare il numero di pere senza rimuoverne la voce dal dizionario possiamo cambiare il valore associato:

```
>>> Magazzino['pere'] = 0
>>> print Magazzino
{'banane': 312, 'arance': 525, 'pere': 0, 'mele': 430}
```

La funzione **len** opera anche sui dizionari ritornando il numero di coppie chiave-valore, cioè il numero di voci del dizionario:

```
>>> len(Magazzino)
4
```

## METODI DEI DIZIONARI

Un **metodo** è simile ad una funzione, visto che prende parametri e ritorna valori, ma la sintassi di chiamata è diversa. Il metodo **keys** prende un dizionario e ritorna la lista delle sue chiavi: invece di invocarlo con la sintassi delle funzioni

```
keys(Eng2Ita)
usiamo la sintassi dei metodi
Eng2Ita.keys()
```

Scrivi quanto segue (dopo la prima riga premi <invio>):

```
>>> Eng2Ita.keys()
['one', 'three', 'two']
```

Questa forma di **notazione punto** specifica il nome della **funzione** **keys** ed il nome dell'**oggetto** cui applicare la funzione **Eng2Ita**. Le parentesi vuote indicano che questo metodo non prende parametri.

Una chiamata ad un metodo è detta **invocazione**; in questo caso diciamo che stiamo invocando **keys** sull'oggetto **Eng2Ita**.

Il metodo **values** funziona in modo simile: ritorna la lista dei valori in un dizionario:

```
>>> Eng2Ita.values()
['uno', 'tre', 'due']
```

Se un metodo prende un argomento usa la stessa sintassi delle chiamate di funzioni. Il metodo **has\_key** prende come argomento una chiave e ritorna vero (1) se la chiave è presente nel dizionario, falso (0) in caso contrario:

```
>>> Eng2Ita.has_key('one')
true
>>> Eng2Ita.has_key('deux')
false
```

Se provi a invocare un metodo senza specificare l'oggetto cui si fa riferimento ottieni un errore:

```
>>> has_key('one')
NameError: has_key
```

Purtroppo il messaggio d'errore a volte, come in questo caso, non è del tutto chiaro: Python cerca di dirci che la funzione **has\_key** non esiste, dato che con questa sintassi abbiamo chiamato la funzione **has\_key** e non invocato il metodo **has\_key** dell'oggetto.

Esempio (scrivi ed esegui il seguente programma)

```
eta = {
    'Adelaide' : 35,
    'Antonio' : 35,
    'Virginia' : 2.3,
    'Renato' : 15
}
print eta['Renato']
print eta['Adelaide']
for nome,anni in eta.iteritems():
    print "nome:",nome,"anni", " ", eta["", anni
```

## ALIAS E COPIA NEI DIZIONARI

Visto che i dizionari sono mutabili devi stare molto attento agli alias: quando due variabili si riferiscono allo stesso oggetto un cambio effettuato su una influenza immediatamente il contenuto dell'altra. Se desideri poter modificare un dizionario e mantenere una copia dell'originale usa il metodo `copy`.

Per fare un esempio costruiamo un dizionario *Opposti* che contiene coppie di parole dal significato opposto:

```
>>> Opposti = {'alto': 'basso', 'giusto': 'sbagliato',
'vero': 'falso'}
>>> Alias = Opposti
>>> Copia = Opposti.copy()
```

*Alias* e *Opposti* si riferiscono allo stesso oggetto; *Copia* si riferisce ad una copia del dizionario nuova di zecca. Se modifichiamo *Alias*, *Opposti* viene modificato:

```
>>> Alias['giusto'] = 'errato'           <invio>
>>> Opposti['giusto']                   <invio>
'errato'
```

*Opposti* resta immutato se modifichiamo *Copia*:

```
>>> Copia['vero'] = 'false'              <invio>
>>> Opposti['vero']                     <invio>
'falso'
```

**Esercizio:** copia il seguente programma e salvalo con il nome `rubrica.py`

```
def print_menu():
    print '1. Print Phone Numbers'
    print '2. Add a Phone Number'
    print '3. Remove a Phone Number'
    print '4. Lookup a Phone Number'
    print '5. Quit'
    print

numbers = {}
menu_choice = 0
print_menu()
while menu_choice != 5:
    menu_choice = input("Type in a number (1-5):")
    if menu_choice == 1:
        print "Telephone Numbers:"
        for x in numbers.keys():
            print "Name: ",x," \tNumber: ",numbers[x]
        print
    elif menu_choice == 2:
        print "Add Name and Number"
        name = raw_input("Name:")
        phone = raw_input("Number:")
        numbers[name] = phone
    elif menu_choice == 3:
        print "Remove Name and Number"
        name = raw_input("Name:")
        if numbers.has_key(name):
            del numbers[name]
        else:
            print name," was not found"
    elif menu_choice == 4:
        print "Lookup Number"
        name = raw_input("Name:")
        if numbers.has_key(name): #cerca un valore che ha una chiave in un dizionario
            print "The number is",numbers[name]
        else:
            print name," was not found"
    elif menu_choice != 5:
        print_menu()
```

Questo è l'output del programma `rubrica.py`:

```

1. Print Phone Numbers
2. Add a Phone Number
3. Remove a Phone Number
4. Lookup a Phone Number
5. Quit

Type in a number (1-5):2
Add Name and Number
Name:Joe
Number:545-4464

Type in a number (1-5):2
Add Name and Number
Name:Jill
Number:979-4654

Type in a number (1-5):2
Add Name and Number
Name:Fred
Number:132-9874

Type in a number (1-5):1
Telephone Numbers:
Name: Jill Number: 979-4654
Name: Joe Number: 545-4464
Name: Fred Number: 132-9874

Type in a number (1-5):4
Lookup Number
Name:Joe
The number is 545-4464

Type in a number (1-5):3
Remove Name and Number
Name:Fred

Type in a number (1-5):1
Telephone Numbers:
Name: Jill Number: 979-4654
Name: Joe Number: 545-4464

Type in a number (1-5):5

```

Vediamo come funziona il programma.

Innanzitutto viene definita la funzione `print_menu` che visualizza sullo schermo un menu più volte usato nel programma. A questo punto compare la linea `numbers = {}` che dichiara `numbers` come un dizionario.

Le linee seguenti fanno funzionare il menu:

```

for x in numbers.keys():
print "Name: ",x," \tNumber: ",numbers[x]

```

Tramite questo ciclo si possono visualizzare le informazioni contenute nel dizionario.

La funzione `numbers.keys()` restituisce una lista che viene poi utilizzata dal ciclo `for`. Questa lista non ha un ordine particolare, quindi se la volete in ordine alfabetico la dovrete ordinare. Con la notazione `numbers[x]` potete accedere ai singoli membri del dizionario. Ovviamente in questo caso `x` è una stringa. Successivamente la linea `numbers[name] = phone` aggiunge un nome ed un numero di telefono al dizionario. Se `name` è stato già inserito nel dizionario `phone`, rimpiazza il valore precedente. Le linee successive:

```

if numbers.has_key(name):
    del numbers[name]

```

controllano se una chiave è già presente nel dizionario, in tal caso la rimuovono. La funzione `numbers.has_key(name)` ritorna vero se `name` è presente in `numbers`, altrimenti ritorna falso. La linea `del numbers[name]`

rimuove la chiave `name` ed il valore ad essa associato. Le linee:

```

if numbers.has_key(name):
    print "The number is",numbers[name]

```

controllano se nel dizionario è presente una determinata chiave, se la trovano, stampano il numero ad essa associato. Infine, se la scelta non è presente nel menu (quindi non è valida) il programma visualizza nuovamente il menu.

## Il programma precedente italianizzato

```
def scrivi_menu():
    print '1. Stampare Numeri telefonici'
    print '2. Inserire Numero'
    print '3. Eliminare Numero'
    print '4. Cercare Numero'
    print '5. Ciao'
    print

# inizia il programma
numeri = {}
menu_scelta = 0
scrivi_menu()

while menu_scelta != 5:
    menu_scelta = input("Scegli un'azione (digita numero da 1 a 5):")
    if menu_scelta == 1:
        print "Numeri telefonici:"
        for x in numeri.keys():
            print "Nome: ",x," \tNumero: ",numeri[x]
        print
    elif menu_scelta == 2:
        print "Inserisci Nome e Numero"
        nome = raw_input("Nome:")
        tele = raw_input("Numero:")
        numeri[nome] = tele
    elif menu_scelta == 3:
        print "Elimina Nome e Numero"
        nome = raw_input("Nome:")
        if numeri.has_key(nome):
            del numeri[nome]
        else:
            print nome," non c'e'!"
    elif menu_scelta == 4:
        print "Cerca Numero"
        nome = raw_input("Nome:")
        if numeri.has_key(nome):
            print "Il numero e'",numeri[nome]
        else:
            print nome," non c'e'!"
    elif menu_scelta != 5:
        scrivi_menu()
```

## Come ordinare un dizionario

```
#!/usr/bin/env python
# coding=latin-1
print
print "Ordinamento di un dizionario"
rubrica = {"giacomo":3493279676,"alessia":45647,"stefano":6456467567,"francesca":56546676}
print
for x in rubrica.keys():
    print "Name: ",x," \tNumero: ",rubrica[x]

print
print"ora provo ad ordinare per chiave"
print
ordinato=rubrica.keys()
ordinato.sort()
for x in ordinato:
    print "Nome: ",x," \tNumero:",rubrica[x]
```