Programmazione in Python

strutture dati: tuple

Dario Pescini - Mirko Cesarini

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi

dario.pescini@unimib.it

Strutture dati complesse: tuple

tupla

possiede un nome `t` ed aggrega più oggetti organizzati sequenzialmente:

```
t = a 'pippo' 6 d … z
```

Strutture dati complesse: tuple

tupla

possiede un nome `t` ed aggrega più oggetti organizzati sequenzialmente:

```
t = t[0] t[1] t[2] t[3] … t[n-1]
```

Strutture dati complesse: tuple

La tupla è una struttura dati complessa di tipo sequenza, statica ed eterogenea.

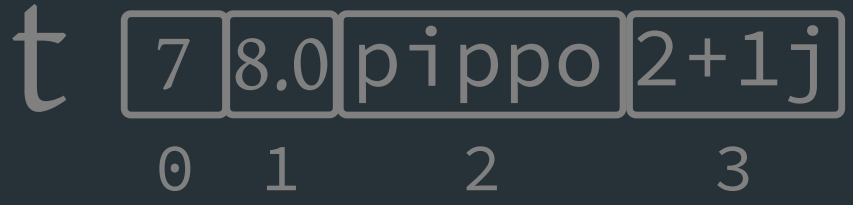
```
t = (7, 3.0 + 5, 'pippo', 2 +1j)
```

dichiarazione tupla: (,)

- `t` nome della tupla
- `( )` delimitatori della tupla - facoltativi
- `7 3.0 + 5 'pippo' 2 +1j` elementi della tupla
- `,` separatore degli elementi ed identificatore della una t-upla

oppure `t = 7, 3.0 + 5, 'pippo', 2 +1j`

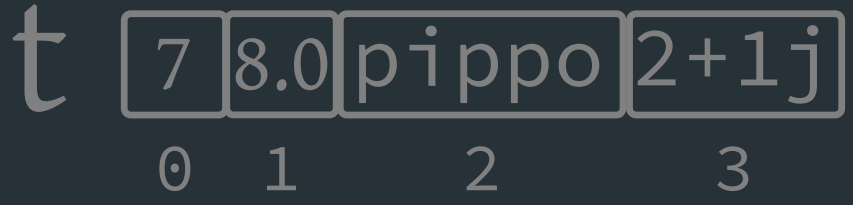
Strutture dati complesse: tuple



sequenza:

```
>>> print t
(7, 8.0, 'pippo', (2+1j))
>>> print t[2]
pippo
>>> _
```

Strutture dati complesse: tuple



sequenza:

```
>>> print t
(7, 8.0, 'pippo', (2+1j))
>>> print t[2]
pippo
>>> _
```

indici negativi:

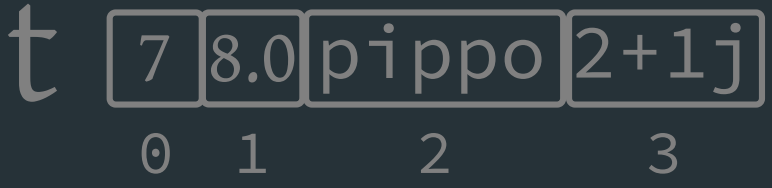
```
>>> t[-1]
(2+1j)
>>>
```

Strutture dati complesse: tuple

lunghezza:

```
>>> len(t)
4
>>> _
```

Strutture dati complesse: tuple



immutabile:

```
>>> t[2] = 'nuova'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>> _
```

Strutture dati complesse: tuple

Slicing:

```
>>> t[:2]
(7, 8.0)
>>> t[2:]
('pippo', (2+1j))
>>> t[1:3]
(8.0, 'pippo')
>>> _
```

Strutture dati complesse: tuple


```
tupla = (7, 3.0 + 5, 'pippo', 2 +1j)

i = 0
while (i < len(tupla)):
    print tupla[i]
    i = i + 1
```

```
dario@vulcano: python tuplaWhile.py
7
8.0
pippo
(2+1j)
dario@vulcano: _
```

Strutture dati complesse: tuple

É possibile estendere una tupla tramite concatenazione:

```
>>> t =  (7, 8.0, 'pippo', 2+1j)
>>> id(t)
4455446760
>>> t = t + (12345,)
>>> print t
(7, 8.0, 'pippo', (2+1j), 12345)
>>> id(t)
4453274064
>>> _
```

A che cosa servono le tuple?

- Permettono di raggruppare informazioni eterogenee e gestirle unitariamente, es.

```
studente = (629435, 'Rossi', 'Mario')
auto = ('BA555AB', 'Fiat', 'Punto')
```

  La non modificabilità è un vantaggio in questi casi

- Utili per definire degli insiemi di costanti

```
giorniSettimana=('lunedi', 'martedi', 'mercoledi',
      'giovedi', 'venerdi', 'sabato', 'domenica')
```

- Le tuple permettono di memorizzare i dati e accedervi più velocemente delle liste

Attenzione!

- Per creare una tupla basta inizializzarla

```
a=(2, 5, 'ciao')
print( a )
```

```
(2, 5, 'ciao')
```

- Se inizializzate una tupla con un singolo valore, ricordatevi di inserire una virgola alla fine, altrimenti il singolo valore verrebbe interpretato non come una tupla ma come un'espressione matematica
- è la virgola prima ancora delle () a definire una tupla

```
b=2, # b=(2,) #equivalente
print( b )
print( type(b) )
```

```
(2)
<type 'tuple'>
```

```
b=(2)
print( b )
print(type(b))
```

```
2
<type 'int'>
```

Avrete notato ...

- Qualche volta è stata utilizzata la print con una sintassi strana

```
variabile = 10
print("testo", 5, "altro testo", variabile)
```

```
('testo', 5, 'altro testo', 10)
```

- L'insieme degli argomenti della print() viene interpretato come una t-upla

```
tu="testo", 5, "altro testo"
print(tu)
print(type(tu))
```

```
('testo', 5, 'altro testo')
<type 'tuple'>
```

Metodi per le tuple

Esistono due metodi che potete utilizzare per le tuple:

- `index(element)` restituisce l'indice più piccolo di element nella tupla
- `count(element)` restituisce il conteggio delle occorrenze di element nella tupla

Metodi: esempi

```
tupla = (0, 0, 0, 1, 0, 2, 4, 5)
```

- `tupla.index(0)` $\longrightarrow$ `0`
  `tupla.index(1)` $\longrightarrow$ `3`

Metodi: esempi

```
tupla = (0, 0, 0, 1, 0, 2, 4, 5)
```

- `tupla.index(0)` $\longrightarrow$ `0`
  `tupla.index(1)` $\longrightarrow$ `3`
- `tupla.count(0)` $\longrightarrow$ `4`