

ES10-CreazioneLista

March 20, 2019

0.0.1 ESERCIZIO 10 - Creazione, accesso e aggiornamento di una lista

Creare una lista di 6 elementi in cui sia presente almeno un valore intero, un valore decimale, una stringa e un valore booleano. Accedere al terzo elemento e in seguito aggiornarlo con la stringa **gatto**. Accedere poi alla sottolista che va dal secondo al quarto elemento e aggiornarla con la lista vuota.

Verificare poi: - se il secondo elemento della lista è diverso dall'intero 345 - se l'intero 345 è presente nella lista - se due elementi a scelta sono presenti contemporaneamente nella lista - se almeno uno tra due elementi a scelta è presente nella lista

0.0.2 Soluzione

Alla variabile `lista` vengono assegnati i 6 elementi usando la sintassi con le parentesi quadre []

```
In [2]: lista = [345, 'ciao', 10.24, 3.14, 90, False]
```

La dimensione della lista viene ottenuta chiamando la funzione `len()` e passando come argomento la variabile `lista` che contiene la lista. Il valore restituito dalla funzione viene assegnato alla variabile `lunghezza`

```
In [3]: lunghezza = len(lista)
```

```
In [4]: lunghezza
```

```
Out[4]: 6
```

Il terzo elemento della lista è quello nella posizione di indice (positivo) 2

```
In [5]: lista[2]
```

```
Out[5]: 10.24
```

In alternativa, si può accedere al terzo elemento della lista tramite l'indice negativo pari a `-lunghezza+2`

```
In [6]: lista[-lunghezza+2]
```

```
Out[6]: 10.24
```

Con la stessa sintassi di accesso a un elemento si aggiorna il terzo elemento con la stringa **gatto**

```
In [7]: lista[2] = 'gatto'
```

Il contenuto della variabile `lista` è ora

```
In [8]: lista
```

```
Out[8]: [345, 'ciao', 'gatto', 3.14, 90, False]
```

Per accedere alla sottolista che va dal secondo al quarto elemento basta utilizzare la sintassi di *slicing* specificando all'interno delle parentesi quadre [] l'indice del secondo elemento (cioè 1) e l'indice dell'elemento successivo al quarto (cioè il quinto a cui corrisponde l'indice positivo 4) separati dal simbolo :

```
In [9]: lista[1:4]
```

```
Out[9]: ['ciao', 'gatto', 3.14]
```

A questo punto si può sostituire in `lista` questa sottolista con la lista vuota

```
In [10]: lista[1:4] = []
```

Il contenuto della lista è ora

```
In [11]: lista
```

```
Out[11]: [345, 90, False]
```

Per verificare se il secondo elemento della lista è diverso dall'intero 345 basta usare l'operatore `!=` con a destra l'intero 345 e a sinistra la sintassi di accesso al secondo elemento in `lista`. Il risultato della valutazione è un valore booleano (True o False)

```
In [12]: lista[1] != 345
```

```
Out[12]: True
```

Per verificare la presenza di un elemento all'interno della lista basta usare l'operatore `in` con a destra l'elemento di cui verificare la presenza e a sinistra la variabile `lista` che contiene la lista. Il risultato della valutazione è un valore booleano (True o False)

```
In [13]: 345 in lista
```

```
Out[13]: True
```

Per verificare la presenza contemporanea di due elementi basta usare l'operatore di congiunzione logica `and`

```
In [14]: 345 in lista and 10.42 in lista
```

```
Out[14]: False
```

Per verificare la presenza di almeno uno tra due elementi basta usare l'operatore di disgiunzione logica `or`

```
In [15]: 345 in lista or 10.42 in lista
```

```
Out[15]: True
```