

## Descrizione

- Una lista è una **sequenza** di oggetti qualunque (anche di tipo diverso, anche altre liste)

```
>>> l = ["abc", 2, [1,2]]
```

- Essendo una sequenza, condivide le operazioni su sequenza viste per le stringhe

```
>>> len(l)
3
```

- La lista è un tipo **mutabile**: può essere allungata ed accorciata, e si possono modificare i suoi elementi

```
>>> l[2] = 10
>>> l
['abc', 2, 10]
```

## Operatori

```
>>> l = [1,2,3]
```

```
>>> l + ["a","b"]
```

```
[1, 2, 3, 'a', 'b']
```

```
>>> l + []          # [] indica una lista vuota
```

```
[1, 2, 3]
```

```
>>> l * 3
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

```
>>> 3 in l
```

```
True
```

```
>>> [3,1] in l # vale solo per singoli elementi
```

```
False
```

## Indicizzazione e sottolista

```
>>> l = [1, 2, 3, 4, 5, 6]
```

```
>>> l[2]
```

```
3
```

```
>>> l[-1]
```

```
6
```

```
>>> l[:]
```

```
[1, 2, 3, 4, 5, 6]
```

```
>>> l[3:]
```

```
[4, 5, 6]
```

```
>>> l[:-2]
```

```
[1, 2, 3, 4]
```

## Matrici

- Una lista i cui elementi siano tutte liste implementa una *matrice*
- Si possono recuperare tramite uno o due indici righe o singoli elementi della matrice

```
>>> matrix = [[1,2,3],[4,5,6],[7,8,9]]
```

```
>>> matrix[2]
```

```
[7, 8, 9]
```

```
>>> matrix[2][0]
```

```
7
```

```
>>> matrix[2][0:2]
```

```
[7, 8]
```

## Modifica con assegnazione ad indici

- Essendo oggetti mutevoli, è possibile modificare il contenuto di una lista

```
>>> l = [1, 2, 3, 4, 5, 6]
```

```
>>> p = l
```

```
>>> l[2] = "a"
```

```
>>> l
```

```
[1, 2, 'a', 4, 5, 6]
```

```
>>> p
```

```
[1, 2, 'a', 4, 5, 6]
```

- Poiché l'oggetto cambia, tutte le variabili che vi si riferiscono “vedono” il cambiamento

## Modifica con assegnazione a sottoliste

- E' possibile sostituire intere sottoliste di una certa lista

```
>>> l = [1, 2, 3, 4, 5, 6]
```

```
>>> l[2:5] = ["a", "b"]
```

```
>>> l
```

```
[1, 2, 'a', 'b', 6]
```

- Non è necessario sostituire la sottolista con una della stessa lunghezza, può essere più lunga o corta
- E' possibile eliminare intere sottoliste sostituendole con la lista vuota

```
>>> l[2:4] = []
```

```
>>> l
```

```
[1, 2, 6]
```

## Modifica tramite il comando `del`

- Il comando `del` permette di cancellare elementi o sottoliste specificando posizione o gamma di posizioni

```
>>> l = [1, 2, 3, 4, 5, 6]
```

```
>>> del l[2]
```

```
>>> l
```

```
[1, 2, 4, 5, 6]
```

```
>>> del l[-1]
```

```
>>> l
```

```
[1, 2, 4, 5]
```

```
>>> del l[2:4]
```

```
>>> l
```

```
[1, 2]
```

```
>>> del l[:]
```

```
>>> l
```

```
[]
```

## Metodi di modifica

- Come i tipi stringa, anche le liste hanno una serie di metodi da eseguire
- Essendo oggetti mutabili, molti di questi metodi modificano direttamente l'oggetto

**append** aggiunge un elemento in fondo alla lista

```
>>> l = [1, 2, 3, 4, 5, 6]
>>> l.append(7)
>>> l
[1, 2, 3, 4, 5, 6, 7]
```

**extend** estende la lista attaccando in fondo tutti gli elementi di una lista

```
>>> l.extend([8, 9])
>>> l
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```



## Metodi di modifica

**insert** Inserisce un elemento prima di una certa posizione

```
>>> l = [1,2,3,4,5,6]
>>> l.insert(3,3.5)
>>> l
[1, 2, 3, 3.5, 4, 5, 6]
```

## Metodi di modifica

**sort** Ordina la lista (assume l'esistenza di una funzione di ordinamento tra elementi, si usa per liste di elementi dello stesso tipo)

```
>>> l = ["basso", "medio", "alto"]
>>> l.sort()
>>> l
['alto', 'basso', 'medio']
```

**reverse** Riordina la lista dall'ultimo al primo elemento

```
>>> l.reverse()
>>> l
['medio', 'basso', 'alto']
```