

LISTE

Le liste in sono collezioni di oggetti, di qualsiasi tipo di dato essi siano.

A differenza di molti altri linguaggi, Python permette di creare liste composte da oggetti di diversi tipi di dato tra loro.

Per inizializzare una lista è sufficiente racchiudere i suoi elementi tra parentesi quadre, e separarli con delle virgole:

```
l = [1, 8, "paperino"]
```

Ora la lista l contiene gli elementi: 1, 8, paperino.

N.B. il primo elemento di una lista ha indice 0, l'ultimo ha indice -1.

Per scandire gli elementi di una lista, è sufficiente passarli a rassegna tramite un ciclo(in questo caso meglio for):

```
l = [1, 8, "paperino"]
for el in l:
    if el.isdecimal():
        print("è un numero")
    if el.isalpha():
        print("è una parola")
    else:
        print("è una stringa di caratteri misti")
```

Ad ogni iterazione, la variabile di appoggio el conterrà l'elemento corrente della lista.

METODI SULLE LISTE

Python offre una vasta gamma di metodi da applicare alle liste, tra cui i più utili:

-append(elemento): permette di aggiungere un elemento in coda alla lista

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

-clear(): svuota la lista

```
>>> l = [1,2,3]
>>> l
[1, 2, 3]
>>> l.clear()
>>> l
[]
```

-insert(indice, elemento): inserisce l'elemento richiesto all'indice specificato.

Attenzione: non invertire l'ordine dei parametri:

```
>>> l = ["paperino", "paperoga", "paperone"]
>>> l.insert("gastone", 2)
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    l.insert("gastone", 2)
TypeError: 'str' object cannot be interpreted as an integer
>>> l.insert(2, "gastone")
>>> l
['paperino', 'paperoga', 'gastone', 'paperone']
```

-pop(): rimuove l'elemento in coda:

```
>>> l = ["paperino", "paperoga", "paperone"]
>>> l
['paperino', 'paperoga', 'paperone']
>>> l.pop()
'paperone'
>>> l
['paperino', 'paperoga']
```

-sort(): ordina gli elementi della lista secondo l'ordinamento del tipo(es: numeri in ordine crescente, stringhe secondo la tabella ASCII):

```
>>> l = [8, 129, 42]
>>> l.sort()
>>> l
[8, 42, 129]
>>> l = ["topolino", "paperino", "archimede"]
>>> l.sort()
>>> l
['archimede', 'paperino', 'topolino']
```

ESEMPIO:

trovare il massimo elemento di una lista:

```
max_n = l[0]
for el in l:
    if el > max_n:
        max_n = el

#più semplicemente, sfruttando l'ordinamento
l.sort()
max_n = l[-1]
```