Gli Array in Python

Prof. Francesco Gobbi

I.I.S.S. Galileo Galilei, Ostiglia

2 febbraio 2026

Cos'è un Array

Un **array** è una struttura dati che consente di memorizzare **più valori all'interno di un'unica variabile**.

Questo significa che, invece di creare tante variabili diverse, possiamo raccogliere i dati in un'unica struttura ordinata.

Caratteristiche fondamentali:

- Gli elementi sono memorizzati in **posizioni ordinate**
- Ogni posizione è identificata da un **indice numerico**
- Gli elementi devono essere **tutti dello stesso tipo** (omogeneità)

Gli array sono molto utilizzati quando dobbiamo gestire **insiemi di dati dello stesso tipo**.

Indice dell'Array

L'**indice** indica la posizione di ogni elemento all'interno dell'array.

In Python la numerazione parte sempre da **0**.

Questo permette di accedere in modo diretto e veloce ad ogni elemento.

Esempio:

```
[10, 20, 30, 40]
  0   1   2   3
```

Importare la Libreria

Gli array non sono una struttura nativa di Python.

Per poterli utilizzare dobbiamo importare la libreria **array** assegnandole un alias.

```
1 import array as arr
```

Da questo momento potremo usare `arr.array()` per creare i nostri array.

Creare un Array Vuoto

Quando creiamo un array dobbiamo sempre specificare il **tipo degli elementi** che conterrà.

Questo perché gli array devono contenere solo dati **omogenei**.

```
1 import array as arr
2
3 numeri = arr.array('i')
```

Abbiamo creato un array vuoto che conterrà solo numeri interi.

Creare un Array con Elementi

Possiamo anche inizializzare un array inserendo subito alcuni valori.

In questo modo l'array viene creato già popolato.

```
1 import array as arr
2
3 numeri = arr.array('i', [10, 20, 30, 40])
```

Codifiche Principali dei Tipi di Array

Il tipo degli elementi viene indicato tramite una **codifica**.

Questa codifica dice a Python quale tipo di dato potrà essere inserito nell'array.

Codice	**Tipo di dato**
`'i'`	Intero con segno
`'I'`	Intero senza segno
`'f'`	Numero reale (float)
`'d'`	Numero reale doppia precisione
`'u'`	Caratteri Unicode

Aggiungere Elementi all'Array

Per inserire nuovi valori utilizziamo il metodo **append()**.

Il nuovo elemento viene sempre inserito **in fondo all'array**.

```
1 numeri.append(50)
```

Scorrere l'Array con un for sugli Elementi

Possiamo leggere tutti gli elementi dell'array usando un ciclo `for`.

In questo caso il ciclo lavora direttamente sugli oggetti contenuti nell'array.

```
1 for numero in numeri:
2     print(numero)
```

Scorrere l'Array con un for usando l'Indice

Possiamo anche scorrere l'array utilizzando l'indice.

Questo è utile quando dobbiamo conoscere la posizione dell'elemento.

```
1 for i in range(len(numeri)):
2     print(numeri[i])
```

Modificare un Elemento in un Indice Specifico

Possiamo modificare direttamente il valore in una certa posizione.

Basta indicare l'indice tra parentesi quadre.

```
1 numeri[2] = 99
```

Eliminare un Elemento in un Indice Specifico

Per eliminare un elemento utilizziamo il metodo **pop()** indicando l'indice.

Quando un elemento viene eliminato:

- L'array si accorcia di una posizione
- Tutti gli elementi successivi si spostano a sinistra
- Gli indici vengono aggiornati automaticamente

Esempio pratico:

```
numeri = arr.array('i', [10, 20, 30, 40])

numeri.pop(1)

print(numeri)
```

Risultato del vettore:

```
array('i', [10, 30, 40])
```

L'elemento 20 è stato eliminato e gli altri si sono spostati.

Eliminare un Elemento con `del vettore[i]`

Oltre al metodo `pop(i)`, esiste un altro modo per eliminare un elemento da un array: l'istruzione **del**.

Scrivendo:

```
del numeri[1]
```

l'elemento in posizione 1 viene eliminato.

Esempio completo:

```
import array as arr

numeri = arr.array('i', [10, 20, 30, 40])

del numeri[1]

print(numeri)
```

Risultato del vettore:

```
array('i', [10, 30, 40])
```