

# 5 - Array

## Array Monodimensionali

Un array è una struttura dati che memorizza una collezione di elementi dello stesso tipo. La dichiarazione di un array richiede due informazioni fondamentali:

1. Il **tipo di dato** base (ad esempio `int`, `float`, `char`).
2. Il **numero di elementi** che si desidera memorizzare.

## Sintassi

Per dichiarare un array monodimensionale, si utilizza la seguente sintassi:

```
tipo nome_array[dimensione];
```

Ad esempio:

```
int numeri[10];
```

Questo dichiara un array di tipo `int` capace di contenere 10 numeri interi.

## Accesso agli Elementi

Gli elementi dell'array sono accessibili tramite un **indice**, che parte sempre da `0` (zero-based indexing). Per accedere o modificare un elemento si utilizza la notazione con le parentesi quadre:

```
nome_array[indice];
```

Ad esempio:

```
numeri[0] = 5; // Assegna il valore 5 al primo elemento dell'array.  
int valore = numeri[3]; // Legge il quarto elemento dell'array.
```

## Array Multidimensionali

Gli array multidimensionali permettono di organizzare i dati in più dimensioni, ad esempio matrici (bidimensionali), tabelle tridimensionali e così via.

## Definizione

La dichiarazione segue una sintassi simile agli array monodimensionali, ma con più indici:

```
tipo nome_array[dim1][dim2]...[dimN];
```

Ad esempio:

```
int matrice[3][4];
```

Questo crea un array bidimensionale di 3 righe e 4 colonne.

## Concetti Importanti

1. Un array multidimensionale può essere visto come un array di array. Ad esempio, una matrice `int matrice[3][4]` è un array di 3 elementi, ognuno dei quali è a sua volta un array di 4 interi.
2. Il numero massimo di dimensioni è teoricamente infinito, ma nella pratica si usano al massimo 2 o 3 dimensioni, sia per semplicità che per limitazioni di memoria.

## Accesso agli Elementi

Per accedere a un elemento specifico, si usano indici separati da parentesi quadre:

```
matrice[2][3] = 10; // Assegna 10 all'elemento nella terza riga e quarta colonna.  
int valore = matrice[1][1]; // Legge l'elemento nella seconda riga e seconda colonna.
```

## Operazioni Sugli Array

Alcune operazioni comuni includono:

- **Assegnazione di valori:**

```
matrice[3][3] = s[1] + x; // Somma un valore e lo assegna a un elemento specifico.
```

- **Incremento:**

```
matrice[i][j] += 1; // Incrementa il valore corrente di 1.
```

- **Spostamento di valori:**

```
matrice[i][j] = matrice[i+1][j]; // Copia un valore dalla riga successiva.
```

## Inizializzazione

Gli array multidimensionali possono essere inizializzati come gli array monodimensionali:

```
int matrice[2][2] = {{1, 2}, {3, 4}};
```