

# Uso avanzato dei puntatori

Matteo Spanio

10 aprile 2024

## Strutture

### Strutture in C

Il C non permette di definire Classi e oggetti, si possono comunque definire dei tipi aggiuntivi:

- Strutture: insieme di dati
- Unioni: alternative tra tipi diversi
- Enumerazioni: insieme di costanti

#### **struct**

Una struttura è un insieme di dati, si differenzia dagli array perché:

- gli elementi possono essere di tipo diverso
- gli elementi sono identificati da un nome

```
struct studente {  
    char nome[20];  
    char cognome[20];  
    int matricola;  
};
```

### Inizializzazione

Si possono inizializzare le strutture quando vengono dichiarate:

```
struct studente s = {"Mario", "Rossi", 12345};
```

Si possono inizializzare meno elementi rispetto a quelli dichiarati, in questo caso gli elementi non inizializzati vengono impostati a 0.

Un modo alternativo di inizializzare:

```
struct studente s = {  
    .nome = "Mario",  
    .cognome = "Rossi",  
    .matricole = 12345,  
};
```

## Accedere agli elementi

Per accedere ai singoli elementi di una struttura (detti anche membri) si usa l'operatore . (punto) dopo il nome della variabile:

```
struct studente s = {"Mario", "Rossi", 12345};  
  
printf("Nome: %s\n", s.nome);  
printf("Cognome: %s\n", s.cognome);  
s.matricola = 54321;
```

## Copie di strutture

Si può copiare interamente una struttura usando una variabile dello stesso tipo:

```
struct studente a, b = {"Matteo", "Spanio", 56789};  
a = b;
```

### ! Importante

Solo l'operatore = è valido tra 2 struct, gli operatori == e != NON si possono usare per vedere se due strutture sono uguali.

## Dare i nomi alle strutture

Le strutture possono avere un nome, gli si può associare un tipo, oppure possono essere anonime.

Le strutture col nome, detto *structure tag*, si dichiarano come abbiamo già visto:

```
struct nome {  
    type member_name;  
};
```

### Avviso

Il tipo di una variabile `struct` è `struct nome_struct`, omettere la parola `struct` è un errore.

## Strutture anonime

Nel caso in cui un si voglia usare una `struct` solo in un punto specifico del codice non è necessario associarvi un nome, si può dichiarare e associare direttamente:

```
struct { int x; int y; } punto;  
punto.x = 12;  
punto.y = 18;
```

## typedef

`typedef` è un operatore che permette di definire alias per i tipi:

```
typedef int Bool;  
typedef float Euro;  
typedef char[] String;
```

Queste dichiarazioni permettono scrivere codice più chiaro.

## Strutture con typedef

Dal momento che le strutture sono usate moltissimo e i programmatori sono pigri, solitamente, si preferisce omettere la parola chiave `struct` per riferirsi al tipo delle strutture grazie a `typedef`:

```
typedef struct {  
    int x;  
    int y;  
} Point;  
Point punto = {  
    .x = 1,  
    .y = 2,  
};
```