



Struct

Informatica B

A cosa servono?

```
int anno;  
int pagine;  
char autore[30];  
char titolo[100];
```

❑ Se ho bisogno di rappresentare un altro libro?

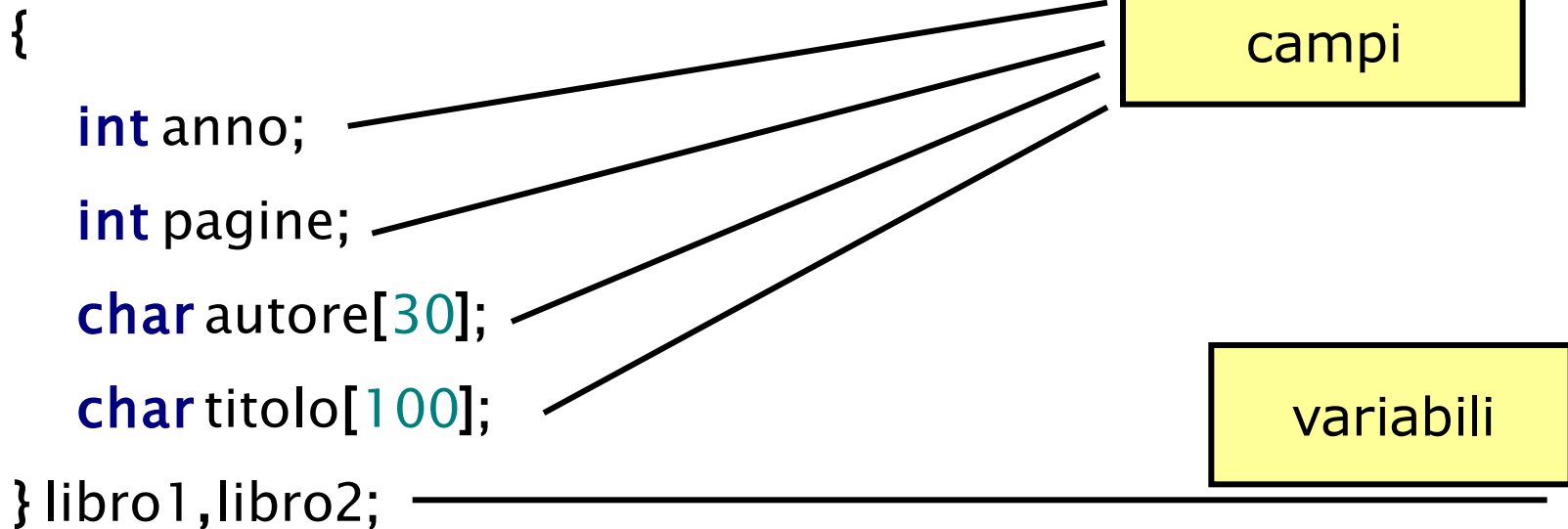
```
int anno2;  
int pagine2;  
char autore2[30];  
char titolo2[100];
```

- Non è compatto
- Poco leggibile
- Non consente di incapsulare l'informazione

Struct

- ❑ La **struct** permette di rappresentare in maniera compatta ed incapsulata tipi di dati con una struttura complessa:

struct



- ❑ Rispetto agli array, gli elementi non sono numerati ma hanno un nome e possono essere di tipo diverso

Accesso ai campi

- ❑ Per accedere ad un campo di una struct si usa la seguente sintassi:

`<nome_variabile> . <nome_campo>`

- ❑ Esempio:

```
typedef struct {  
    int anno;  
    int pagine;  
    char autore[30];  
    char titolo[100];  
} libro;
```

```
libro l;
```

```
l.anno = 1998;
```

- ❑ Come per gli array la lettura avviene elemento per elemento

```
libro l;  
printf("Inserire anno: ");  
scanf("%d",&l.anno);  
printf("Inserire num pagine: ");  
scanf("%d",&l.pagine);  
printf("Autore: ");  
gets(l.autore);  
printf("Titolo: ");  
gets(l.titolo);
```

Array di struct

- ❑ È possibile (e spesso utile) dichiarare un array di struct
- ❑ In questo modo è possibile gestire una sequenza di elementi ognuno dei quali rappresenta un dato strutturato
- ❑ Esempio

typedef struct

```
{  
    int anno;  
    int pagine;  
    char autore[30];  
    char titolo[100];  
} libro;
```

```
libro biblioteca[N];
```

Esempio

- ❑ Leggere i dati di una biblioteca e trovare il libro più vecchio

```
#include <stdio.h>
#define N 3
typedef struct {
    int anno;
    int pagine;
    char autore[30];
    char titolo[100];
} libro;
int main() {
    libro biblioteca[N];    int i,vecchio;
    /*Lettura*/
    for (i=0; i<N; i++) {
        scanf("%d",&biblioteca[i].anno);
        scanf("%d",&biblioteca[i].pagine);
        gets(biblioteca[i].autore);
        gets(biblioteca[i].titolo); }

    /*Ricerca del libro piu' vecchio*/
    vecchio=0;
    for (i=1; i<N; i++)
        if (biblioteca[i].anno < biblioteca[vecchio].anno)
            vecchio = i;
    /* Stampa il libro piu' vecchio*/
    printf("Dati libro piu' vecchio\n");
    printf("Anno: %d\n",biblioteca[vecchio].anno);
    printf("Num pagine: %d\n",biblioteca[vecchio].pagine);
    printf("Autore: %s\n",biblioteca[vecchio].autore);
    printf("Titolo: %s\n",biblioteca[vecchio].titolo);

    system("PAUSE");
    return 0;
}
```

Assegnamento e confronto

- ❑ Il C permette di effettuare assegnamento fra struct senza dover considerare un elemento alla volta:

```
libro l1, l2;
```

```
l1 = l2;
```

- ❑ Non è invece possibile effettuare confronti fra struct ma occorre considerare un campo per volta!

```
if(l1 == l2)
```

```
    printf("Sono lo stesso libro");
```