

# **Chapitre 1 LES STRUCTURES**

## ✓ Définition

Une structure est un bloc permet de regrouper des informations de type différent.

Une structure en C est une collection de données de types différents regroupées sous une entité logique dite structure.

## ✓ Déclaration

<pre>struct nom_structure {     type_champ1 nom_champ1;     type_champ2 nom_champ2;     .     .     .     type_champN nom_champN; };</pre>	<pre><b>Typedef</b> struct nom_structure {     type_champ1 nom_champ1;     type_champ2 nom_champ2;     .     .     .     type_champN nom_champN; }Stuct_Type;</pre>
--	---

## Exemple

Typedef struct livre

```
{
    char    auteur[12];
    char titre[20];    float
    prix;
} livres;
```

### Déclarer une variable structure

La déclaration d'une variable de type structure peut introduire le mot struct ou non selon la méthode de déclaration.

- **Syntaxe 1**

```
struct Nom_Structure Nom_variable ;
```

**Exemple :** struct Livres lv ;

- **Syntaxe 2**

```
Nom_Structure Nom_variable ;
```

**Exemple :** Livres lv ;

### Accès aux champs d'une structure

Un champ d'une structure est référencé par son nom précédé du nom de la variable structure.

### **Syntaxe**

Nom\_Variable\_Structure.Nom\_Champ

### **Exemple**

Les champs de la variable lv sont accessibles via les noms :

**lv.auteur, lv.titre et lv.prix**

On pourrait avoir :

```
lv.prix=109;  
strcpy(lv.auteur,"Allali");
```

### **Remarque :**

On peut mettre : **lv2=lv1**

Par contre, on ne peut pas comparer globalement des variables structurées. Il faut comparer chacun de leurs champs. **if(lv1==lv2)** n'est pas autorisé.

### **✓ Tableaux de structures**

Lorsque le nombre d'enregistrements à gérer est important, on peut alors créer un tableau dont les éléments sont des variables structurées.

### **Syntaxe**

Nom\_Structure Nom\_Tab[Taille] ;

### **Exemple**

livre T[50] ;

Cette déclaration, permet de créer un tableau de 50 éléments dont chacun est une variable structurée de type " livre". Comme d'habitude, les éléments seront ici désignés par le nom du tableau avec un index, soit T[0] à T[49] dans notre exemple.

Pour accéder à un champ d'un élément du tableau, il faut le désigner selon la syntaxe suivante:

```
nom_tableau[index].nom_champ
```

On pourrait écrire :

```
T[36].an=1997;  
scanf("%s", T[4].auteur);
```