

LES TABLEAUX EN C

Un tableau est une collection d'éléments stockés dans des emplacements de mémoire continus.

Eléments	3	12	10	4	2	1	15	30
indices	0	1	2	3	4	5	6	7

Taille du tableau : 8

Premier indice : 0

Dernier Indice : 7

Déclaration de tableaux en C :

Nous pouvons déclarer un tableau en spécifiant son type et sa taille ou en l'initialisant, ou les deux.

Déclaration de tableau en spécifiant la taille

Exemple 1 :

```
1
2   int main(void){
3       // Déclaration de tableau en spécifiant la taille
4       int tab[10];
5
6       // nous pouvons également déclarer un tableau de taille spécifiée par l'util
7       int n=10;
8       int T[n];
9
10      return 0;
11  }
```

Déclaration de tableau en initialisant des éléments

Exemple 2 :

```
1   int main(void){
2       int tab[] = { 10, 20, 30, 40 }
3
4       // Le compilateur crée un tableau de taille 4.
5       // ci-dessus est identique à "int tab[4] = {10, 20, 30, 40}"
6
7       return 0;
8   }
```

Déclaration de tableau en spécifiant la taille et en initialisant les éléments

Exemple 2 :

```
1   int main(void){
```

```

2      int tab[6] = { 10, 20, 30, 40 }
3      // Le compilateur crée un tableau de taille 6, initialise d'abord
4      // 4 éléments tels que spécifiés par l'utilisateur et deux éléments par 0.
5      // ci-dessus est identique à "int tab[] = {10, 20, 30, 40, 0, 0}"
6      return 0;
7  }
8

```

Propriétés d'un tableau en C :

Accéder aux éléments du tableau :

Les éléments de tableau sont accessibles à l'aide d'un index entier. L'indice de tableau commence par 0 et va jusqu'à la taille du tableau moins 1.

Exemple 3 :

```

1
2      #include < stdio.h>
3      int main(void){
4          int tab[] = { 10, 20, 30, 40 }
5          tab[2]=5;
6          tab[0]=2;
7          tab[2]=tab[1];
8
9          printf("%d - %d - %d - %d", tab[0], tab[1], tab[2], tab[3]);
10         return 0;
11     }

```

2 - 20 - 20 - 40

Attention : Il n'y a pas de contrôle d'indice hors limite :

En effet, il n'y a pas de contrôle d'indice hors limite en C, par exemple, le programme suivant compile bien mais peut produire une sortie inattendue lors de l'exécution.

Exemple 4 :

```

1
2      #include < stdio.h>
3      int main(void){
4          int tab[2];
5          printf("%d \n", arr[3]);
6          printf("%d ", arr[-2]);
7
8          return 0;
9      }

```

-434996632
32766