

# Les bases du C/C++

# SOMMAIRE

<u>Les tableaux.....</u>	<u>1-3</u>
<u>Les structures.....</u>	<u>4-6</u>
<u>Les énumérations.....</u>	<u>7</u>

# Les tableaux

Ils permettent de stocker un grand nombre de données du même type.

Déclaration :

Tableau de 4 int :

`int tableau[4];`

Tableau de 20 double :

`double tableau[20];`

Tableau de 500 char :

`char tableau[500];`



**TOUT TABLEAU COMMENCE A L'INDICE 0**



Remplissage:

`tableau[0] = 10;`

`tableau[1] = 20;`

`tableau[2] = 30;`

`tableau[3] = 40;`

# Les tableaux

Lire une valeur du tableau :

```
printf("%d", tableau[0]);
```

OU

```
logDebug(tableau[0]);
```

Parcourir un tableau : **boucle for**

```
for (i = 0 ; i < 4 ; i++)  
{  
    printf("%d \n", tableau[i]);  
}
```

# Les tableaux

Initialiser un tableau :

```
for (i = 0 ; i < 4 ; i++)  
{  
    tableau[i] = 0;  
}
```

OU

```
int tableau[4] = {0};
```

On peut affecter des valeurs aux cases d'un tableau directement avec la syntaxe :

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

Si on veut la case 0 à 10, la case 1 à 15 et les autres à 0 on écrira simplement :

```
int tableau[4] = {10, 15};
```

# Les structures

Ce sont de **nouveaux types** créés par **assemblage de variables**

```
struct NomDeVotreStructure  
{  
    int variable1;  
    int variable2;  
    int autreVariable;  
    double nombreDecimal;  
};
```



Ici on doit mettre un **point virgule** à la fin !

Les structures sont souvent placées dans les **headers** (fichiers.h)

# Les structures

Pour se simplifier la vie : le **typedef**

Il va remplacer '**struct** NomDeVotreStructure' par 'NomDeVotreStructure'

- A mettre AVANT la déclaration de structure !

Exemple :

```
typedef struct Coordonnees Coordonnees;  
struct Coordonnees  
{  
    int x;  
    int y;  
};
```

Déclaration d'une variable de type Coordonnees:

```
Coordonnees point;  
// L'ordinateur comprend qu'il s'agit de "struct Coordonnees" grâce au typedef
```

# Les structures

Accéder aux composantes de la structure :

```
Coordonnees point;  
point.x = 10;  
point.y = 20;
```

En résumé :

```
Variable.nomDeLaComposante
```

Initialiser la structure : pareil que les tableaux

```
Coordonnees point = {0, 0};
```



# Les énumérations

Même déclaration que les structures.

Contient une **liste de valeurs possibles** pour une variable.

```
typedef enum Volume Volume;  
enum Volume  
{  
    FAIBLE, MOYEN, FORT  
};
```

Associer un chiffre aux valeurs :

- Par défaut, l'énumération commence à 0 (FAIBLE) puis 1 (MOYEN) et 2 (FORT)
- On peut forcer les valeurs

```
typedef enum Volume Volume;  
enum Volume  
{  
    FAIBLE = 10, MOYEN = 50, FORT = 100  
};
```