

# C/C++: Les tableaux & chaînes de caractères.



Vincent Gaudillière

Polytech Nancy

2017 / 2018

## Les tableaux.

---

Intérêt: stocker une série de valeurs.

Un tableau en C:

- est une suite de **variables de même type**, situées dans un **espace contigu en mémoire**,
- possède une certaine **taille**.

## Les tableaux (1D).

---

```
/* Déclaration */  
int mon_tableau[4];  
  
/* Affectation */  
mon_tableau[0] = 10; // premier élément = indice 0  
mon_tableau[1] = 3;  
mon_tableau[2] = 6;  
mon_tableau[3] = 4;
```

```
/* Initialisation */  
int mon_tableau[4] = {10,3,6,4};
```

## Les tableaux 2D (tableaux de tableaux).

---

```
/* Déclaration */
```

```
int tab[3][3];
```

```
/* Affectation */
```

```
tab[0][0]=1; tab[0][1]=2; tab[0][2]=3; // 1 2 3
```

```
tab[1][0]=4; tab[1][1]=5; tab[1][2]=6; // 4 5 6
```

```
tab[2][0]=7; tab[2][1]=8; tab[2][2]=9; // 7 8 9
```

```
/* Initialisation */
```

```
int mon_tableau[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
```

## Les tableaux 3D (tableaux de tableaux de tableaux).

---

```
/* Déclaration */
```

```
int T[2][2][2];
```

```
/* Affectation */
```

```
T[0][0][0]=1; T[0][0][1]=2; // 1 2
```

```
T[0][1][0]=3; T[0][1][1]=4; // 3 4
```

```
T[1][0][0]=5; T[1][0][1]=6; // 5 6
```

```
T[1][1][0]=7; T[1][1][1]=8; // 7 8
```

```
/* Initialisation */
```

```
int T[2][2][2] = {{{1,2},{3,4}},{5,6},{7,8}}};
```

## Allocation statique.

---

Déclaration de la taille d'un tableau:

```
int size = 10;  
int mon_tableau[size]; // INTERDIT (en C89) !
```

```
int mon_tableau[10]; // OK
```

```
#define HEIGHT_IMAGE 1080 // directive de préprocesseur  
#define WIDTH_IMAGE 1920 // directive de préprocesseur  
  
unsigned char mon_image1 [HEIGHT_IMAGE] [WIDTH_IMAGE]; //OK  
unsigned char mon_image2 [HEIGHT_IMAGE] [WIDTH_IMAGE]; //OK
```

## Les chaînes de caractères.

---

Une chaîne de caractères:

- sert à représenter le texte,
- est un tableau de variables de type char (caractère),
- possède un caractère de fin de chaîne ('\0').

Le type char peut représenter un entier (entre -128 et 127):

```
char c1 = 5; // entier entre -128 et 127 (256 valeurs)
```

Le type char peut aussi représenter un caractère (codage ASCII):

```
char c2 = 'a'; // lettre minuscule  
char c3 = 'B'; // lettre majuscule  
char c4 = ':'; // caractère spécial  
char c5 = '5'; // chiffre
```

## Les chaînes de caractères.

---

```
/* Déclaration */  
char maChaine[40];  
  
/* Affectation */  
maChaine[0] = 'H';  
maChaine[1] = 'e';  
maChaine[2] = 'l';  
maChaine[3] = 'l';  
maChaine[4] = 'o';  
maChaine[5] = '\\0'; // caractère de fin de chaîne!
```

```
/* Initialisation */  
char maChaine[40] = {'H', 'e', 'l', 'l', 'o', '\\0'};  
char maChaine[40] = "Hello"; // à utiliser.
```



## Affichage des (chaînes de) caractères.

---

```
char monChar1 = 'A';  
printf("%c\n", monChar1); // A  
printf("%d\n", monChar1); // 65
```

```
char monChar1 = 65;  
printf("%d\n", monChar2); // 65  
printf("%c\n", monChar2); // A
```

```
char maPhrase[30] = "Bonjour à tous!";  
printf("%s\n", maPhrase); // Bonjour à tous!
```