

Variables simples

→ Variables et types de variables

Une variable est une zone de la mémoire vive de l'ordinateur, à laquelle on donne un nom. Cette zone est destinée à accueillir un contenu (nombre, caractère(s), ...). Ce contenu étant plus ou moins volumineux, et devant être manipulé de telle ou telle manière, il est nécessaire d'informer l'ordinateur de son type.

Le type d'une variable permet de préciser l'encombrement que prendra la variable dans la mémoire, et comment elle sera manipulée dans le programme.

Remarque : en C « décimal » signifie « entier en base 10 », alors que « nombre à virgule flottante » signifie « nombre décimal »

Type	Contenu	Taille	Plage de valeurs
char	caractère «signé»	1 octet	-128 → 127 (codes ASCII signés)
unsigned char	caractère	1 octet	0 → 255 (codes ASCII)
int	entier décimal : élément de Z	2 ou 4 octets <i>suivant les compilateurs</i>	↕
unsigned int	entier décimal : élément de Z ⁺	2 ou 4 octets <i>suivant les compilateurs</i>	↕
short	entier décimal «court»	2 octets	-32768 → 32767
unsigned short	entier décimal «court» positif	2 octets	0 → 65535
long	entier décimal «long»	4 octets	-2147483647 → 2147483648
unsigned long	entier décimal «long» positif	4 octets	0 → 4294967295
float	nombre à virgule flottante : élément de D	4 octets	$\pm 3,4 \cdot 10^{-38} \rightarrow \pm 3,4 \cdot 10^{38}$
double	élément de D «long»	8 octets	$\pm 1,7 \cdot 10^{-308} \rightarrow \pm 1,7 \cdot 10^{308}$
long double	élément de D «très long»	10 octets	$\pm 3,4 \cdot 10^{-4932} \rightarrow \pm 3,4 \cdot 10^{4932}$

→ Déclaration d'une variable

Toutes les variables utilisées dans un programme doivent être déclarées au début du programme, de la manière suivante :

type_de_variable nom_de_variable1, nom_de_variable2, ... ;

Les variables peuvent être initialisées (c'est-à-dire que l'on peut leur affecter une valeur) lors de leur déclaration :

type_de_variable nom_de_variable1, nom_de_variable2=valeur, ... ;

(Dans l'exemple ci-dessus, **nom_de_variable2** a été initialisée à **valeur**, alors que **nom_de_variable1** n'a pas été initialisée.)

Attention : la zone-mémoire désignée par une variable, peut contenir n'importe quoi, tant que la variable n'a pas été initialisée, c'est-à-dire que l'on n'a pas mis quelque chose dans cette zone-mémoire.

Exemples de déclarations

```
int i, j=17, k; /* i et k n'ont pas été initialisées */  
char c='W', d='@';  
float volume=0.0; /* 0.0 est la "version float" de 0 */  
unsigned short numero=1; /* numero ne pourra être négatif */
```

→ Affectation d'une valeur à une variable

Pour affecter une valeur à une variable, c'est-à-dire modifier le contenu de la zone-mémoire correspondante, on utilise principalement l' opérateur d'affectation = suivant le modèle :

variable = valeur ;

Exemple **indice = 7 ; /* On affecte 7 à la variable « indice » */**

Attention au type de la variable : veiller à affecter une valeur *d'un certain type* à une variable *de même type*. Si ce n'est pas le cas, l'affectation se fera quand même, mais l'ordinateur effectuera une conversion entre les types, ... et des surprises peuvent survenir !

Exemple **int i=2 ;
float r=45,842517 ;

i=r; /* On essaie d'affecter le contenu de r à i */**
Si on affiche alors la valeur de **i** , on obtient : **45**
(comme **i** est un entier, les décimales de **r** ont été supprimées)