

Pour manipuler des données, les langages C/C++ nous offrent le concept de **variable**.

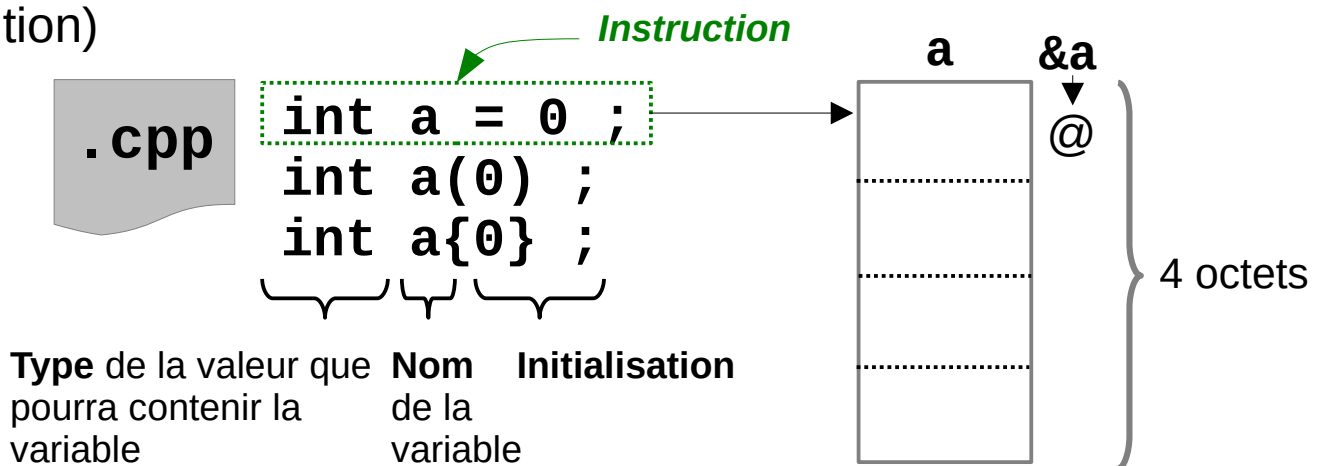
Une variable est un **espace de stockage pour une donnée**.

Une variable est associée à symbole (un **nom** qui sert d'identificateur) qui renvoie à une position de la mémoire (une **adresse**) dont le contenu peut prendre successivement différentes valeurs pendant l'exécution d'un programme.

La variable doit avoir un **type** qui est la convention d'interprétation de la séquence de bits (codage binaire). Le type de la variable spécifie aussi sa **taille** mémoire (la longueur de cette séquence) soit habituellement 8 bits, 32 bits, 64 bits, ...

La valeur d'une variable est la **séquence de bits** elle même. Cette séquence peut être codée de différentes façons suivant son **type**.

Instanciation : (définition)



8 bits	char	octet	<i>octet, byte</i>
16 bits	short	mot	<i>word</i>
32 bits	<u>int</u>	double mot	<i>dword</i>
64 bits	long	long mot	<i>lword</i>

Exemples :

```
bool reussie = true; // true (ou false) est une valeur booléenne
char unite   = 'g';  // un caractère, ne pas oublier les simples quotes '
int  nb0eufs = 3;    // 3 est une valeur entière
unsigned long int a = 12345678UL; // U pour unsigned et L pour long
int *p = &nb0eufs;  // p est un pointeur et &nb0eufs est une adresse

float qte = 350.0f; // f pour float (simple précision)
double poids = 0. ; // "0." rend la valeur réelle pour un double
                  // (double précision)
```

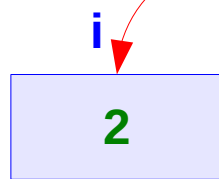
$\text{uneVariable} \leftarrow \text{uneValeur}$
 $\text{uneVariable} = \text{uneValeur};$

Une instruction en C/C++

lvalue = **rvalue**;

Une **left-value** doit donc être un emplacement de stockage en mémoire possédant un type précis, c'est-à-dire la référence à quelque chose de modifiable. Exemple : une variable, une case de tableau, ...

i = **2** ;

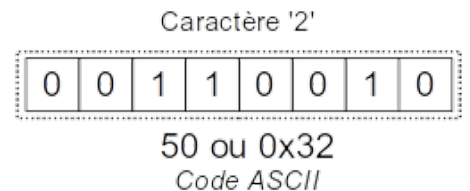


0xbfa56938
adresse mémoire

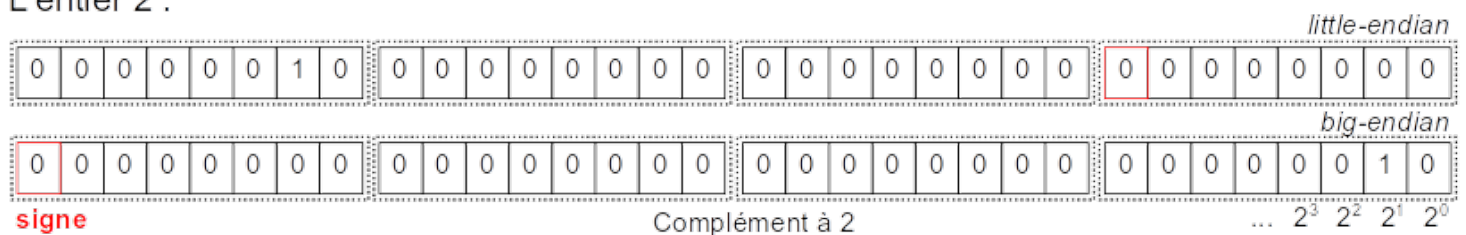
Une **right-value** doit être une valeur d'un type précis mais n'a pas forcément de zone de stockage en mémoire. Exemple : une valeur, une constante, une variable, une expression, ...

Notion de type

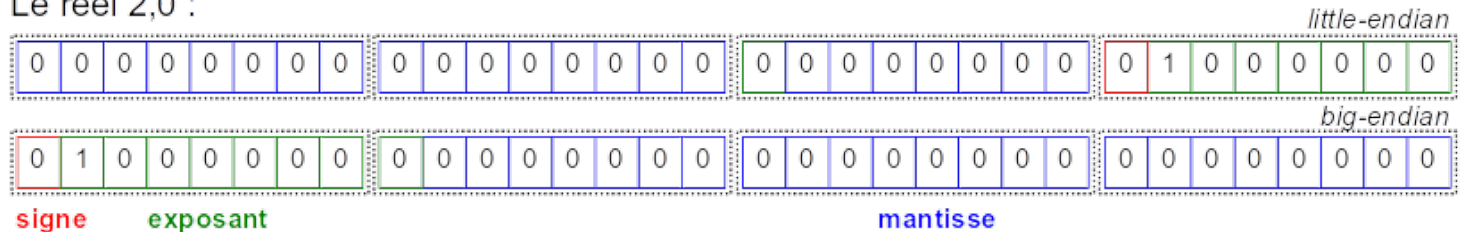
Le caractère '2' :



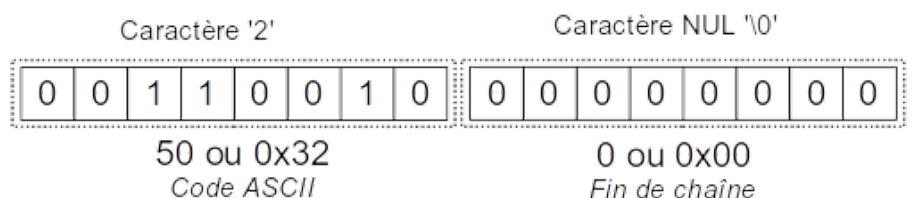
L'entier 2 :



Le réel 2,0 :



La chaîne de caractères "2" :



Integral types



Type specifier	Equivalent type	Width in bits
		C++ standard
signed char	signed char	at least 8
unsigned char	unsigned char	
short	short int	at least 16
short int		
signed short		
signed short int		
unsigned short	unsigned short int	
unsigned short int		
int	int	at least 16
signed		
signed int		
unsigned	unsigned int	
unsigned int		
long	long int	at least 32
long int		
signed long		
signed long int		
unsigned long	unsigned long int	
unsigned long int		
long long	long long int (C++11)	at least 64
long long int		
signed long long		
signed long long int		
unsigned long long	unsigned long long int (C++11)	
unsigned long long int		

Boolean type

`bool` — integer type, capable of holding one of the two values: `true` or `false`. The value of `sizeof(bool)` is implementation defined and might differ from `1`.

Floating-point types

Standard floating-point types

`float` — single precision floating-point type.
`double` — double precision floating-point type.
`long double` — extended precision floating-point type.

Type	Size in bits	Format	Value range
character	char 8	signed	−128 to 127
		unsigned	0 to 255
integer	short 16	signed	−32768 to 32767
		unsigned	0 to 65535
	int 32	signed	−2,147,483,648 to 2,147,483,647
		unsigned	0 to 4,294,967,295
	long 64	signed	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
		unsigned	0 to 18,446,744,073,709,551,615
binary floating-point	float 32	IEEE-754 🔗	min normal: $\pm 1.175,494,3 \cdot 10^{-38}$ max: $\pm 3.402,823,4 \cdot 10^{38}$
	double 64	IEEE-754 🔗	min normal: $\pm 2.225,073,858,507,201,4 \cdot 10^{-308}$ max: $\pm 1.797,693,134,862,315,7 \cdot 10^{308}$
	long double 128	IEEE-754 🔗	min normal: $\pm 3.362,103,143,112,093,506,262,677,817,321,752,602,6 \cdot 10^{-4932}$ max: $\pm 1.189,731,495,357,231,765,085,759,326,628,007,016,2 \cdot 10^{4932}$