



# Les variables



- Les variables permettent de représenter les **données d'un programme**.
- Chaque variable est matérialisée par un emplacement **mémoire identifié** par un **nom**.
- Chaque variable possède :
  1. un **nom**
  2. un **type**
  3. une **valeur**

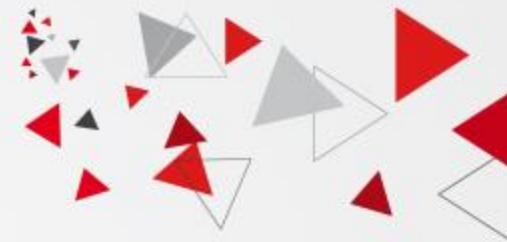


A blue arrow points from the word "nom" in the third list item to the "Variable" column header. A blue arrow points from the word "type" to the "adresse" column header. A blue arrow points from the word "valeur" to the "Valeur" column header. A purple curly brace groups the three columns.

| <i>Variable</i> | <i>adresse</i>          | <i>Valeur</i> |
|-----------------|-------------------------|---------------|
| c               | 0xbffff4ac (3221222572) | 0             |
| b               | 0xbffff4b0 (3221222576) | 2             |
| a               | 0xbffff4b4 (3221222580) | 1             |



# Les variables



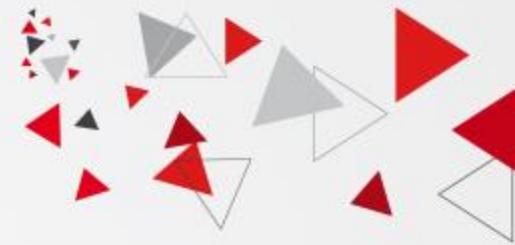
- Qu'est-ce qu'une variable ?
  1. Le nom d'un emplacement mémoire
  2. On en crée presque autant qu'on veut
  3. Son contenu peut changer dans le temps
  4. Elle contient toujours quelque chose



Une variable sera toujours définie avant utilisation : elle doit être associée à un type de données



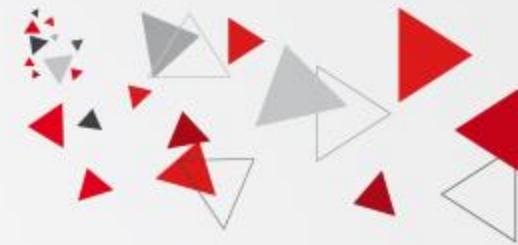
# Les variables ( Nom )



- Un **nom de variables**:
  - Contient **32 caractères max**
  - Commence par une **lettre** ou le **symbole \_**ou **\$**
  - C distingue les **majuscules** et les **minuscules**.  
**compteur** ' et ' **Compteur**' sont **deux** variables différentes
  - Ne peut pas être un des **mots réservés du C**: Int, char, float, if ....
  - Le choix des noms est important pour la **lisibilité** du programme



# Les variables ( Type )



C est un langage typé:

## Types de base en C

### Types Entiers

- ◆ **int** : entier standard
- ◆ **short** : entier court
- ◆ **long** : entier long
- ◆ **char** : caractère

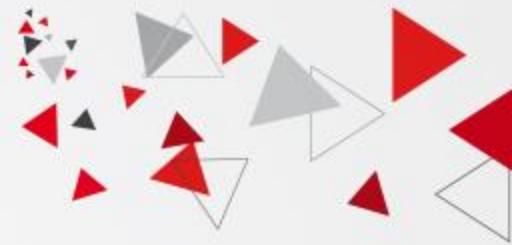
### Types Réels

- ◆ **float** : réel
- ◆ **double** : réel double
- ◆ **long double** : réel long double

Le langage C implémente le type char qui est en fait numérique.



# Les variables ( Type : Entiers )



Entiers

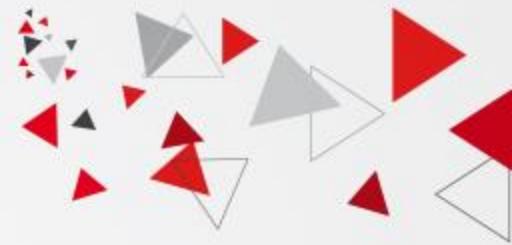
| Type           | Nombre d'octets | Domaine               |
|----------------|-----------------|-----------------------|
| int            | 4               | $[-2^{31}, 2^{31}-1]$ |
| unsigned int   | 4               | $[0, 2^{32}-1]$       |
| short          | 2               | $[-32768, 32767]$     |
| unsigned short | 2               | $[0, 65535]$          |
| long           | 4               | $[-2^{63}, 2^{63}-1]$ |
| unsigned long  | 4               | $[0, 2^{63}-1]$       |
| char           | 1               | $[-128, 127]$         |
| Unsigned char  | 1               | $[0, 255]$            |

Réels

|             |    |                                   |
|-------------|----|-----------------------------------|
| Float       | 4  | $[-3.4*10^{38}, 3.4*10^{38}]$     |
| double      | 8  | $[-1.7*10^{308}, 1.7*10^{308}]$   |
| long double | 10 | $[-3.4*10^{4932}, 1.7*10^{4932}]$ |



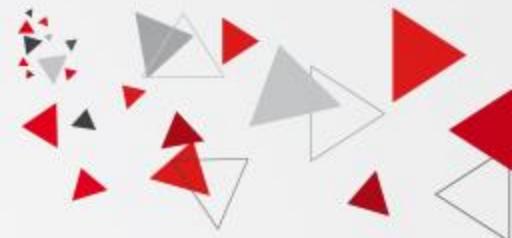
# La portée des variables



- Un bloc est délimité par des accolades
- Toute variable a une **durée de vie** bornée au bloc où elle est déclarée
- Ce bloc définit la **portée de la variable**
- L'espace mémoire pour stocker la variable est alloué lors de sa déclaration et libéré à la fin du bloc où elle a été allouée



# Les variables ( Exemple )



**main** est toujours la fonction principale d'un programme C

Début de bloc d'instructions

```
#include <stdio.h> /* Bibliothèque d'entrées-sorties standard */  
int main()  
{  
    int a;  
    // déclaration  
    a=127;  
    // affectation  
    printf("la valeur de a = %d",a); // affichage de la valeur de a  
    return 0;  
}
```

Fin de bloc d'instructions