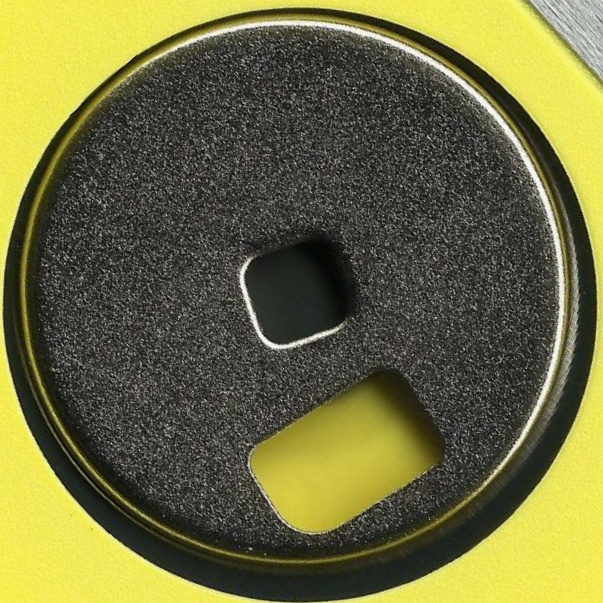


Gestion des variables



# Gestion des variables : Automatique

- Par défaut, une variable locale est automatique.
- Elle est déclarée entre les accolades { } d'une fonction.
- Son existence est liée à l'exécution de la fonction, elle n'existe que pendant l'exécution de la fonction.
- Son contenu est perdu entre 2 appels de la fonction.
- Lors de l'entrée dans la fonction, la variable est créée sur la pile (stack) et elle est détruite lors de la sortie de la fonction.
- On dit encore que sa portée ou visibilité est limitée à la fonction qui l'a déclarée.



Ce type de variable n'est pas initialisée à sa déclaration : elle contient alors une valeur indéterminée.



# Gestion des variables : Globale

- Elle est définie avant le programme principal (main).
- Ce type de variable est accessible et modifiable par le programme principal et par n'importe quelle fonction.
- Par défaut, elle est initialisée à 0.
- Les variables globales sont à éviter sauf cas particuliers.
- Une variable globale utilise de la mémoire pendant tout le programme alors qu'une variable locale n'occupe de la mémoire que pendant l'exécution de la fonction.

# Gestion des variables : Externe

- Le langage C prévoit une déclaration permettant de spécifier qu'une variable globale a déjà été définie dans un autre fichier source.
- Celle-ci se fait à l'aide du mot-clé `extern`.

test.h

```
int x;  
  
int test() {  
    ...  
}
```

programme.c

```
#include "test.h"  
  
extern int x;  
  
int main() {  
    ...  
}
```

- La déclaration `extern` n'effectue pas de réservation d'emplacement de variable. Elle ne fait que préciser que la variable globale `x` est définie par ailleurs et elle en précise le type

# Gestion des variables : **Externe**

- C'est le rôle de l'éditeur de liens que de retrouver dans le premier module l'adresse effective de la variable x et de la reporter dans le second module.

# Gestion des variables : **Register**

- L'usage de ce mot clé est utile dans un contexte de logiciel embarqué.
- Il indique que la variable devrait être stockée dans un registre du processeur.
- Cela permet de gagner en performance par rapport à des variables qui seraient stockées dans un espace mémoire beaucoup moins rapide, comme une pile placée en mémoire vive.
- Il fonctionne ni avec les variables globales ni les pointeurs.

# Gestion des variables : Statique

- C'est une variable locale à une fonction. Au même titre qu'une variable automatique, son existence n'a lieu que pendant l'exécution de la fonction.
- Par contre, ce type de variable **garde sa valeur d'un appel à l'autre** de la fonction.
- Par défaut, elle est initialisée à 0. Là aussi, sa portée est limitée à la fonction qui l'a déclarée. Ce type de variable est précédé du mot clé **static** lors de la déclaration

# Gestion des variables : Statique

```
#include <stdio.h>

void displayHello();

int main() {
    for (int i = 0; i < 5; i++) {
        displayHello();
    }
}

void displayHello() {
    static int count = 0;
    count++;
    printf("Hello %d !\n", count);
}
```

```
> hello.exe
Hello 1 !
Hello 2 !
Hello 3 !
Hello 4 !
Hello 5 !
```