



La **portée** (scope) d'un identifiant est l'étendue au sein de laquelle cet identifiant est lié. En général, on distingue :

- La portée locale (ou variable locale) à l'intérieur d'un bloc `{}`
- La portée globale (ou variable globale) à l'extérieur d'un bloc `{}`

Attention :

Un identifiant à portée locale masque un identifiant de même nom, mais de plus grande portée (« résolution au plus proche »).

Déclarer deux fois le même identifiant dans la même portée peut être considéré comme une erreur.

Les variables globales sont généralement déconseillées (i.e. interdites) en raison des "effets de bord" (sauf exception comme les verrous dans la programmation multi-tâches).

```
#include <iostream>

namespace toto {
    int a = 1;
}

int a = 0;

int main() {
    int a = 2;

    std::cout << "[global] a = " << ::a << std::endl; // affiche 0
    std::cout << "[toto] a = " << toto::a << std::endl; // affiche 1
    std::cout << "[local] a = " << a << std::endl; // affiche 2

    {
        int a = 42;
        std::cout << "[local] a = " << a << std::endl; // affiche 42 !
    }

    std::cout << "[global] a = " << ::a << std::endl; // affiche 0
    std::cout << "[toto] a = " << toto::a << std::endl; // affiche 1
    std::cout << "[local] a = " << a << std::endl; // affiche 2

    return 0;
}
```

.cpp

✦ Le mot clé **static** peut modifier le comportement :

- Dans une portée locale, la variable n'est pas redéclarée et conserve sa valeur à chaque passage dans le bloc
- Dans une portée globale, l'accès à la variable est limitée au fichier.