

mettre ce chapitre aussi tôt ?

Variables statiques et globales

Dans les chapitres précédents, les variables sont locales, elles sont détruites en sortant du bloc :

```
{  
    int const i {};  
    // utilisation de i  
} // i est détruit ici
```

Avec une portée local, plus facile de suivre et comprendre le rôle et l'évolution des valeurs.

Possible d'avoir une portée globale ou une valeur statique, mais rend plus difficile la lecture du code, surtout sur de gros projets (on ne peut savoir quand une variable est modifiée)

L'utilisation de variables non locales doit toujours se faire après avoir vérifié que c'était la seule solution

Variables globales

portée dans toutes l'application, pas dans un bloc.

```
int i {};  
  
void f() {  
    cout << i << endl;  
}  
  
int main() {  
    ++i;  
    f();  
    ++i;
```

```
f();  
}
```

Variables statiques

valeur est conservée

```
void f() {  
    static int i {};  
    ++i;  
    cout << i << endl;  
}  
  
int main() {  
    f();  
    f();  
    f();  
}
```

affiche :

```
1  
2  
3
```

[Chapitre précédent](#) [Sommaire principal](#) [Chapitre suivant](#)

[Cours, C++](#)