

Les catégories de collections

La bibliothèque standard propose de nombreux conteneurs de données. Pour le moment, vous avez vu principalement `std::vector` et `'std::string'`. Ce chapitre présente les autres collections de la bibliothèque standard et leurs fonctionnalités de base.

Conteneur séquentiel et associatif

main.cpp

```
#include <iostream>
#include <vector>
#include <map>

int main() {
    using data_t = std::pair<int, std::string>;
    std::vector<data_t> v {
        { 0x8cc6, "toto" },
        { 0x8b49, "titi" },
        { 0x4b2e, "tata" }
    };
    for (auto & p: v)
        std::cout << p.first << ' ' << p.second << std::endl;

    std::cout << std::endl;

    std::map<int, std::string> m {
        { 0x8cc6, "toto" },
        { 0x8b49, "titi" },
        { 0x4b2e, "tata" }
    };
    for (auto & p: m)
        std::cout << p.first << ' ' << p.second << std::endl;
}
```

affiche :

```
36038 toto
35657 titi
19246 tata

19246 tata
35657 titi
36038 toto
```

Conteneurs séquentiel (vector, array, list, etc) et associatif (map, set, unordered, etc). Mémoire contiguë, liste chaînée.

Créer une collection

Construction (par défaut, par copie, reserve/resize). Size, capacité, empty, shrink_to_fit. swap

Ajouter et supprimer des éléments

Ajout et suppression d'éléments. L'idiome remove-erase. emplace vs push/insert

Complexité algo

Accéder aux éléments

Accès aux éléments (random access, front, back)

Les autres fonctions membres

Autres fonctions membre (find, count, etc)

Itérateurs

allocator, data()

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