

Support de cours

Cours:

## Initiation à la programmation (en C++)

Vidéo:

### Utilisation des vector (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

**Éléments d'un tableau. Déclaration de la variable. Boucle for standard. Nom de la variable. Nouvel âge variable. Dernier élément du tableau. Nom du tableau. Nouvelle norme c. Première façon. Question est. Taille du tableau. Âge du prochain. Fonction f. Troisième voie. Symbole de modification.**



[vers la recherche de séquences vidéo](#)  
(dans Initiation à la programmation (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

# Tableaux : utilisation des `vector` (Partie 2)

## Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Une manière courante d'accéder aux éléments d'un tableau,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



- Si l'on ne veut pas modifier les éléments du tableau :

```
for(auto nom_de_variable : tableau)
```

- Si l'on veut modifier les éléments du tableau :

```
for(auto& nom_de_variable : tableau)
```

*int f(int x);*

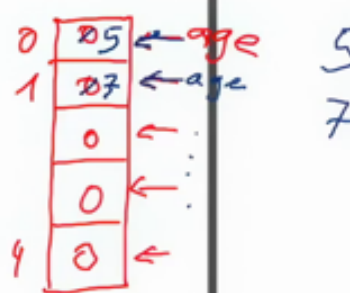
où *type* est le type des éléments contenus dans le tableau.

Exemples :

```
vector<int> ages(5);

for(auto& age : ages) {
    cout << "Age de l'employé suivant? ";
    cin >> age;
}

cout << "Age des employés : " << endl;
for(auto age : ages) {
    cout << " " << age << endl;
}
```



consiste à parcourir le tableau. En C++, nous avons trois façons d'itérer sur un tableau. Depuis la nouvelle norme C++ 2011, nous avons un moyen qui nous permet d'itérer simplement sur toutes les valeurs. Nous le détaillerons dans un instant. Vous pourriez aussi, bien sûr, avec l'itération vous avez vu jusqu'à présent, la boucle for standard, itérer sur les différents index explicites du tableau, donc si je désigne ces index en utilisant i, je vais faire une boucle itérant sur différentes valeurs pour i, je vais commencer avec la valeur 0, et ira par exemple un par un, jusqu'au dernier élément du tableau, le dernier élément, laissez-moi vous rappeler, si t est la taille du tableau, a l'indice t - 1. Alors ici, nous allons faire une boucle qui va de 0 à strictement inférieur à la taille du tableau. La question est de savoir comment nous allons représenter cette taille, comment allons-nous connaître cette taille, Je vous en parlerai en détail dans un instant également. Je vais également expliquer ce que représente ce nouveau type size\_t. Enfin, troisième voie, beaucoup plus avancée, qui ne sera pas présenté dans ce cours, consiste à utiliser ce qu'on appelle des itérateurs. Une fois que vous souhaitez parcourir un tableau, la question que tu te poseras, est laquelle de ces deux manières précédentes choisir. Dans la plupart des cas, je vous recommande de choisir la première façon de procéder, d'utiliser les itérations C++ 2011, que nous allons maintenant détailler. Si vous ne pouvez pas, dans ce cas vous utilisez une itération standard. Passons maintenant aux itérations introduites dans la nouvelle norme C++ 2011. Ces itérations utilisent également le mot-clé pour, ainsi que des parenthèses (). Mais entre parenthèses, nous utiliserons cette fois le séparateur deux points (:) au lieu du point-virgule, et nous aurons deux parties, au lieu de trois parties. Avant les deux points, nous aurons la déclaration de la variable que

notes

résumé

0m 5s



- Si l'on ne veut pas modifier les éléments du tableau :

```
for(auto nom_de_variable : tableau)
```

- Si l'on veut modifier les éléments du tableau :

```
for(auto& nom_de_variable : tableau)
```

*int f(int& x);*

où *type* est le type des éléments contenus dans le tableau.

Exemples :

```
vector<int> ages(5);

for(auto& age : ages) {
    cout << "Age de l'employé suivant? ";
    cin >> age;
}

cout << "Age des employés : " << endl;
for(auto age : ages) {
    cout << " " << age << endl;
}
```

nous voulons utiliser pour parcourir le tableau, et après les deux points, simplement le nom du tableau sur lequel nous voulons itérer. Si l'itération sur ce tableau doit modifier les éléments, vous devrez ajouter entre le mot-clé réservé ici, auto et le nom de la variable, ajoutez le symbole de modification, comme nous l'indiquons à une fonction *f* que ses paramètres pourraient être modifiés par la fonction.

Permettez-moi de vous rappeler que si par exemple nous avons une fonction *f*, qui prend un paramètre *x*, que nous voulons modifier, ce que vous devez faire, c'est de le transmettre par référence, donc ici nous avons utilisé le symbole de référence pour indiquer un passage par référence. C'est exactement la même chose ici dans cette boucle *for*, nous utiliserons le symbole de référence pour indiquer que le tableau itéré peut être modifié via la variable utilisée pour l'itérer. Prenons un exemple. Supposons que nous voulions manipuler un tableau d'entiers, que nous nommerons ici âges, au pluriel, qui est un tableau dynamique contenant des entiers, et que nous initialisons au départ avec cinq entiers tous de valeur nulle. Au début, nous voudrions remplir ce tableau avec les valeurs demandées à l'utilisateur. Pour y parvenir, nous parcourons le tableau avec une boucle *for*. Mais comme nous allons modifier le tableau, il est important d'itérer sur la boucle *for* ici, avec le symbole de modification, avec une référence au tableau. La syntaxe est donc la suivante, nous aurons le mot-clé *pour*, suivi d'une parenthèse, suivi du nom du tableau, *ages*, c'est-à-dire après les deux points, et avant de retrouver l'*auto*, et le nom d'une variable, donc ici on déclare un nouvel âge variable au singulier, qui parcourra le tableau. Ensuite, dans le bloc associé au *for*, vous pouvez simplement utiliser cette variable *age*, c'est le nom que nous avons donné à la variable qui va stocker les différentes valeurs du tableau. Comme toute variable, elle

notes

résumé

- Si l'on ne veut pas modifier les éléments du tableau :

```
for(auto nom_de_variable : tableau)
```

- Si l'on veut modifier les éléments du tableau :

```
for(auto& nom_de_variable : tableau)
```

*int f(int& x);*

où *type* est le type des éléments contenus dans le tableau.

Exemples :

```
vector<int> ages(5);

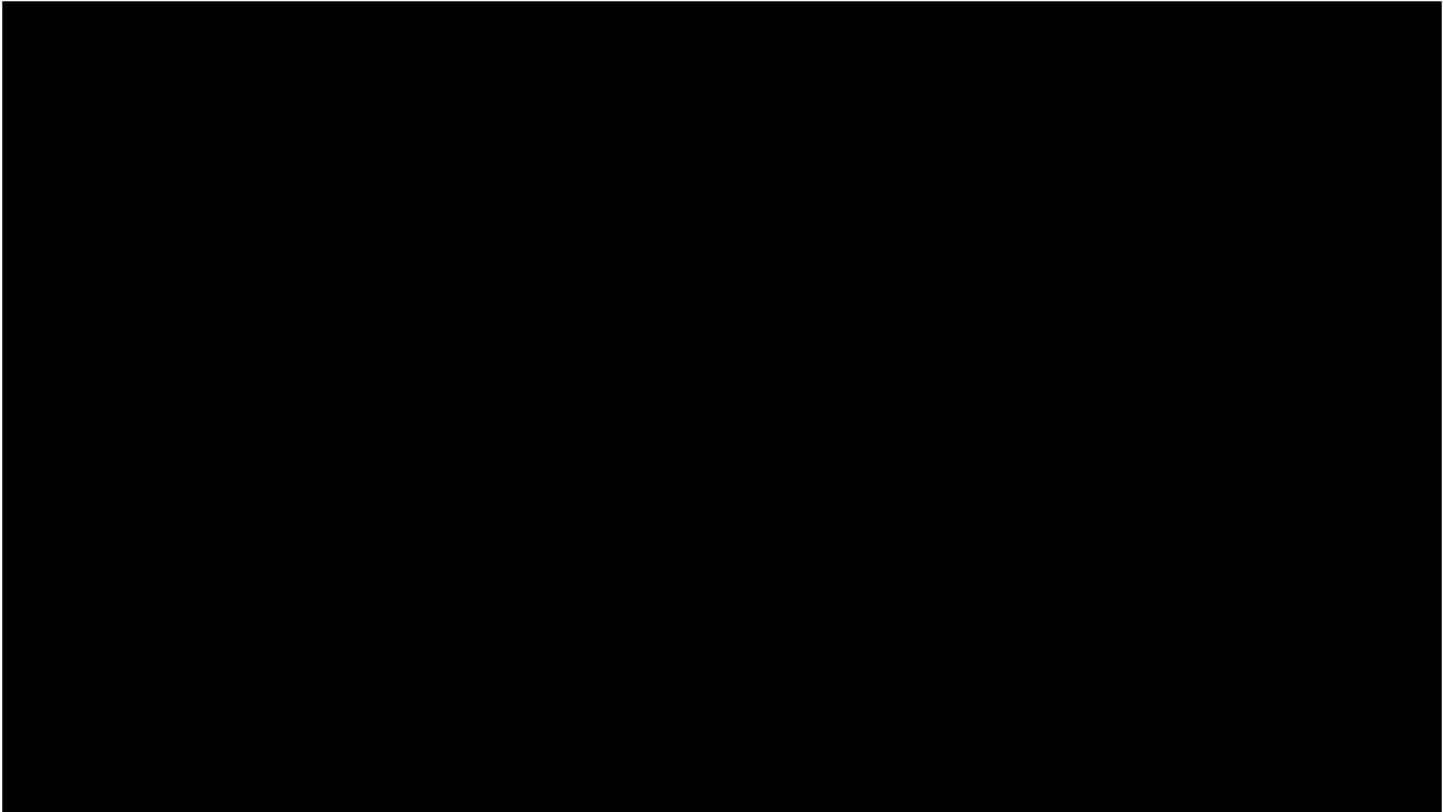
for(auto& age : ages) {
    cout << "Age de l'employé suivant? ";
    cin >> age;
}

cout << "Age des employés : " << endl;
for(auto age : ages) {
    cout << " " << age << endl;
}
```

va alternativement parcourir les différents éléments du tableau, il va parcourir le tableau étape par étape. Chaque étape imprimera le message prévu, l'âge du prochain employé, demander l'âge, et stocké dans cet âge variable ici, qui, alternativement, passera en revue chaque élément du tableau. Ce qui va se passer, c'est que la première fois la boucle est exécutée, la variable *age* indiquera le premier élément du tableau, et nous demanderons une valeur pour cette variable. Supposons que l'utilisateur entre la valeur 5, à ce moment-là, nous mettrons 5 dans cette cellule du tableau. On refait ensuite l'itération, c'est le rôle de la boucle *for* d'itérer, et à ce moment-là, l'âge sera déplacé vers la deuxième entrée du tableau. Supposons qu'ici, l'utilisateur entre 7, à ce moment-là, nous en aurons 7 qui iront ici, et ainsi de suite, jusqu'à ce que l'âge ait parcouru l'ensemble du tableau, jusqu'à ce que nous remplissions tout le tableau. Nous pourrions ensuite l'imprimer en utilisant une autre itération, alors nous commençons ici en affichant un message indiquant que nous allons imprimer le tableau des âges, puis on refait une boucle itérative avec le mot-clé *pour*, la parenthèse, le deux points qui sépare d'un côté le nom du tableau, les âges au pluriel, et de l'autre côté,

notes

résumé



une variable. que je peux appeler à nouveau âge, c'est un autre âge variable, il a une autre portée, il aura la portée de cette boucle for ici, tandis que la variable d'âge précédente avait la portée de cette première pour, ici. Et nous utilisons cette variable d'âge pour parcourir le tableau, comme toute autre variable, simplement, que va-t-il se passer c'est qu'au début, cette variable d'âge indiquera le premier élément du tableau, puis en alternance, prendra la valeur des différents éléments du tableau, nous allons donc commencer par imprimer 5 ici, alors nous imprimerons 7... jusqu'à ce que nous ayons parcouru l'ensemble du tableau. jusqu'à ce que nous ayons parcouru l'ensemble du tableau.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

