

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Branchements conditionnels (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

Bloc d'instructions. Certaine condition. Branchements conditionnels. Séquence d'instructions. Parties du programme. Mot-clé if. Nombre de traitements. Séquence précédente. Accolade ouvrante. Valeurs de certaines données. Exemple concret. Mot-clé else. Boucles conditionnelles. Mot clé if. Type d'instructions.



[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/>
page 1/10

Branchements conditionnels

(Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Dans une séquence précédente nous avons vu que programmer c'est décrire une

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



.....

.....

.....

.....

.....

Jusqu'ici, toutes les instructions des programmes étaient exécutées, et les unes après les autres.



tâche que l'on veut faire exécuter à l'ordinateur et que cette tâche nous la décomposons d'un côté en des données qui vont représenter les grandeurs réelles du monde et un certain nombre de traitements, de manipulations, d'opérations sur ces données. On a vu comment certains traitements pouvaient opérer sur les données. Mais pour que ce soit vraiment un programme il faut que les données puissent influencer les traitements et c'est ce qui s'appelle les structures de contrôle, c'est ce que nous allons donc maintenant aborder dans cette séquence. En effet, jusqu'à présent nos programmes étaient simplement constitués d'instructions que l'on

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

0m 5s



Jusqu'ici, toutes les instructions des programmes étaient exécutées, et les unes après les autres.

Les **structures de contrôle** permettent de changer ce comportement.

Il y a 3 structures de contrôle:

- les branchements conditionnels,
- les itérations, et
- les boucles conditionnelles.

Nous allons commencer par les **branchements conditionnels**, qui permet de sauter certaines parties du programme si certaines conditions sont remplies, et qui utilisent le mot-clé `if`.

exécutait les unes après les autres et les données n'avaient aucune influence sur ces traitements. Pour avoir de l'influence, faire que les valeurs de certaines données, de certaines grandeurs, permettent de changer ce comportement strictement linéaire de l'exécution, on a ce que l'on appelle donc des structures de contrôle. Il existe fondamentalement ce qu'on appelle trois types de contrôles. Tout d'abord les branchements conditionnels que nous allons voir dans cette vidéo et qui permettent de choisir une séquence ou une autre en fonction de certaines valeurs. Ensuite on a les itérations qui permettent de répéter toute une séquence d'instructions et enfin on a les boucles conditionnelles qui permettent elles aussi de répéter toute une séquence d'instructions mais à une certaine condition. Commençons donc maintenant par les branchements conditionnels qui comme je viens donc de le dire, permettent de sauter certaines

notes

résumé

0m 49s



i

```
→ if (n < 5) {  
    cout << "Votre nombre est plus petit que 5." << endl;  
} else {  
    cout << "Votre nombre est plus grand ou egal a 5." << endl;  
}
```

parties du programme ou de choisir certaines parties du programme à exécuter. Prenons un exemple concret. Supposons par exemple que vous vouliez faire l'une ou l'autre chose en fonction d'une valeur entrée, saisie au clavier par l'utilisateur. Supposons par exemple que vous décidiez que si n est plus petit que cinq, vous voulez afficher, faire exécuter un certain type d'instructions, afficher par exemple que le nombre est plus petit que cinq et que par contre si n est supérieur ou égal à cinq alors vous voulez faire une autre séquence d'instructions. Et bien c'est ce que l'on appelle ici un branchement conditionnel que nous allons maintenant donc détailler. Pour cela donc vous avez d'abord une partie ici qui va être introduite par le mot-clé `if` qui va faire ce branchement.

notes

résumé

1m 49s




```

if ( COND ) {
    // COND est vraie
} else {
    // COND est fausse
}

```

```

if (n < 5) {
    cout << "Votre nombre est plus petit que 5." << endl;
} else {
    cout << "Votre nombre est plus grand ou egal a 5." << endl;
}

```

Ici if, qui veut dire si en anglais, qui va vous permettre de faire un choix. Si la condition qui est derrière est vérifiée alors le if va exécuter les instructions qui lui sont indiquées et éventuellement vous pourriez avoir une partie sinon, qui en anglais s'écrit avec le mot-clé else, qui se trouve ici, qui si la condition n'est pas vérifiée alors va faire que c'est cette deuxième partie qui va s'exécuter. Le schéma général est donc le suivant. Nous avons le mot clé if qui va être suivi d'une condition entre parenthèses, nous reviendrons plus tard sur les conditions, puis une accolade ouvrante, un bloc d'instructions que l'on veut voir exécutées si la condition est vraie, lorsque la condition sera vraie, c'est ce bloc d'instructions qui va s'exécuter, et puis une accolade fermante. On peut ensuite avoir éventuellement le mot clé-réservé else, de nouveau une accolade ouvrante, un autre bloc d'instructions ici que je mets en bleu, et qui s'exécutera

notes

résumé

2m 37s




```

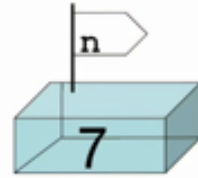
int n;

cout << "Entrez votre nombre:" << endl;
cin >> n;

→ if (n < 5) {
    cout << "Votre nombre est plus petit que 5." << endl;
} else {
    cout << "Votre nombre est plus grand ou egal a 5." << endl;
}

cout << "Au revoir" << endl;

```



Ce qui s'affiche dans la fenêtre Terminal:

```

Entrez votre nombre:
7
|

```

si la condition est fausse. Voilà donc pour le schéma général mais regardons maintenant pas à pas un exemple particulier. Dans cet exemple, la structure de contrôle, le branchement conditionnel qui nous intéresse, va se trouver ici, mais on commence par donc déclarer une variable `n` de type entier que l'on n'a pas initialisée donc ici on ne connaît pas sa valeur puis on affiche sur le terminal ici le message "entrez un nombre" suivi du saut à la ligne puis on attend la saisie au clavier d'une valeur ici de type entier par l'utilisateur. Supposons que l'utilisateur entre la valeur trois donc maintenant `n` a ici la valeur trois. On arrive donc à notre branchement conditionnel qui a été introduit par le mot-clé `if` ici, puis qui a donc sa condition entre parenthèses `n inférieur à cinq`. On commence par se poser la question, évaluer est ce que `n` est strictement inférieur à cinq? Trois est strictement inférieur à cinq, ici la condition est donc vraie et donc on va rentrer dans le code, le bloc ici qui est directement sous le `if`. Donc ce qui se passe c'est que l'on arrive ici à l'exécution du `cout`, "votre nombre est plus petit que cinq", c'est le message qui s'affiche. L'exécution continue ensuite après le bloc, c'est-à-dire ici qu'on va sauter directement à l'instruction ici qui affiche le message "Au revoir" et le programme se termine. Quand est-il maintenant si à cette étape

notes

résumé

3m 50s



Les accolades délimitent un bloc d'instructions

```
if (n < 5) {
    cout << "Votre nombre est plus petit que 5." << endl;
} else {
    cout << "Votre nombre est plus grand ou egal a 5." << endl;
}
```

ici, l'utilisateur avait entré la valeur sept? Alors à ce stade, le programme va de nouveau se poser la question, est-ce que sept est strictement inférieur à cinq? Mais ici la condition va être fausse. Sept n'est pas strictement inférieur à cinq. Comme la condition est fausse, le if va cette fois-ci sauter dans la partie else, va complètement ignorer la partie précédente pour se brancher directement et d'où le nom **branchement conditionnel**, va pour se brancher directement donc ici sur cette ligne d'affichage qui va dire que le nombre n est supérieur ou égal à cinq. Puis on va continuer l'exécution du programme après le bloc, la suite du programme qui suit le bloc de branchement conditionnel et donc afficher le dernier message "Au revoir". Les accolades servent donc à définir le bloc d'instructions qui va être contrôlé,

notes

résumé

5m 25s



soit le bloc d'instructions qui s'exécute lorsque la condition est vraie, soit le bloc d'instructions qui s'exécute lorsque la condition est fausse. Ce bloc d'instructions c'est l'ensemble des instructions que l'on veut contrôler, sur lesquelles on veut justement effectuer un branchement et on peut donc regrouper plusieurs instructions, autant d'instructions que l'on veut, dans un bloc, Supposons par exemple que l'on veuille, non seulement afficher que le nombre est plus petit que cinq mais que l'on veuille aussi dire quelle est la valeur du nombre, alors on va simplement ici dans le cas où la condition est vraie, dans le bloc contrôlé par la condition, lorsque la condition est vraie, ici rajouter une ligne dans ce bloc. Ceci est donc tout à fait possible et ça permet de contrôler plusieurs instructions par notre branchement conditionnel. branchement conditionnel.

6m 25s

