

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Branchements conditionnels (partie 3)

Concepts (extraits des sous-titres générés automatiquement) :

Bloc d'instructions. X égal. Instruction if. Est égal. Bloc de conditions vraies. Premières valeurs. Instructions c. Exécution de ce bloc. Conditions fausses. Haut niveau. Deuxième if. Sous bloc. Fait générales. Condition vraie. Valeurs différentes.



[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/>

Branchements conditionnels

(Partie 3)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

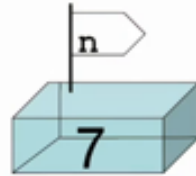


```
int n;

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

if (n < 5) {
    cout << "Votre nombre est plus petit que 5." << endl;
}

→ cout << "Au revoir" << endl;
```



Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre:
7
Au revoir
█
```



Pour résumer, le schéma général d'un branchement conditionnel en C++ est

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

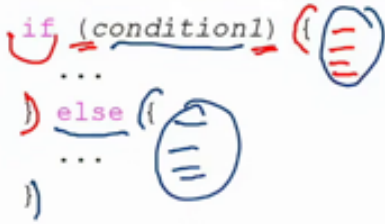
résumé

0m 1s



Les choix imbriqués

L'instruction `if` suit donc le schéma:



Les instructions figurant dans les blocs sont absolument quelconques. *Il peut donc s'agir d'autres instructions `if`.*

donc le suivant : on a une instruction `if` suivie d'une condition entre parenthèses rondes puis un bloc d'instructions ici donc entre accolades qui sont les instructions qui vont s'exécuter si la condition est vraie et éventuellement de façon optionnelle, le mot réservé `else` suivi d'un bloc d'instructions qui vont s'exécuter si la condition est fausse. Et ces instructions ici qui sont dans les blocs sont des instructions C++ tout à fait

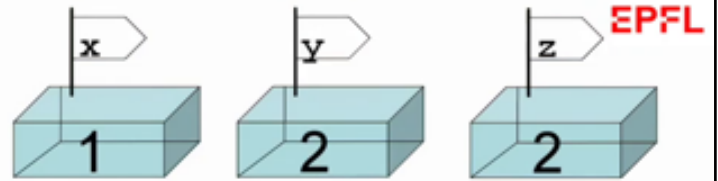
notes

résumé

0m 5s



Supposons:



```
if (x == y) {  
    if (y == z) {  
        cout << "Les trois valeurs sont egales." << endl;  
    } else {  
        cout << "Seules les deux premieres valeurs sont egales." << endl;  
    }  
} else {  
    if (x == z) {  
        cout << "Seules la premiere et la troisieme valeurs sont egales." << endl;  
    } else {  
        if (y == z) {  
            cout << "Seules les deux dernieres valeurs sont egales." << endl;  
        } else {  
            cout << "Les trois valeurs sont differentes." << endl;  
        }  
    }  
}
```

générales, elles peuvent en particulier elles-mêmes contenir d'autres if, c'est ce que l'on appelle des choix imbriqués. Prenons un exemple, supposons par exemple que si on ait trois valeurs donc x, y et z et que on veuille tester comment elles se positionnent par rapport à leur égalité, savoir si x est égal à y, savoir si x est égal à z, ou savoir si les trois variables sont égales. On va commencer donc par exemple, on va faire un test si x est égal à y, et là vous voyez, on va avoir tout un bloc si x est égal y qui va aller jusqu'ici, donc ça, ça sera vrai si x est égal y et si x est différent de y alors à ce moment-là ça sera tout cet autre bloc ici qui sera exécuté. Si x est égal à y alors à ce moment-là on peut rentrer dans un bloc d'instructions et avoir de nouveau une autre instruction ici qui contient elle-même son propre sous-bloc de conditions vraies, ici, et puis éventuellement son else pour un sous bloc ici de conditions fausses. Donc ce qui fait que si x est égal à y et ensuite, on rentre dans le bloc, si y est égal à z, alors dans ce cas-là on affichera que les trois valeurs sont égales. Sinon si x est égal à y mais que y n'est pas égal à z, alors à ce moment-là on exécutera cet affichage, seules les deux premières valeurs sont égales. Et on pourrait procéder de la sorte aussi dans le cas où x est différent de y. Evidemment il ne faut pas abuser de ce genre de solutions. Pratiquement, si vous avez plus que trois niveaux alors à ce moment là le code devient vite illisible. Illustrons maintenant très concrètement l'exemple précédent avec par exemple le cas où on a trois variables x, y et z qui ont toutes les trois la

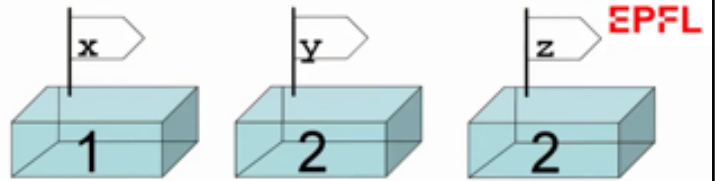
notes

résumé

0m 37s



Supposons:

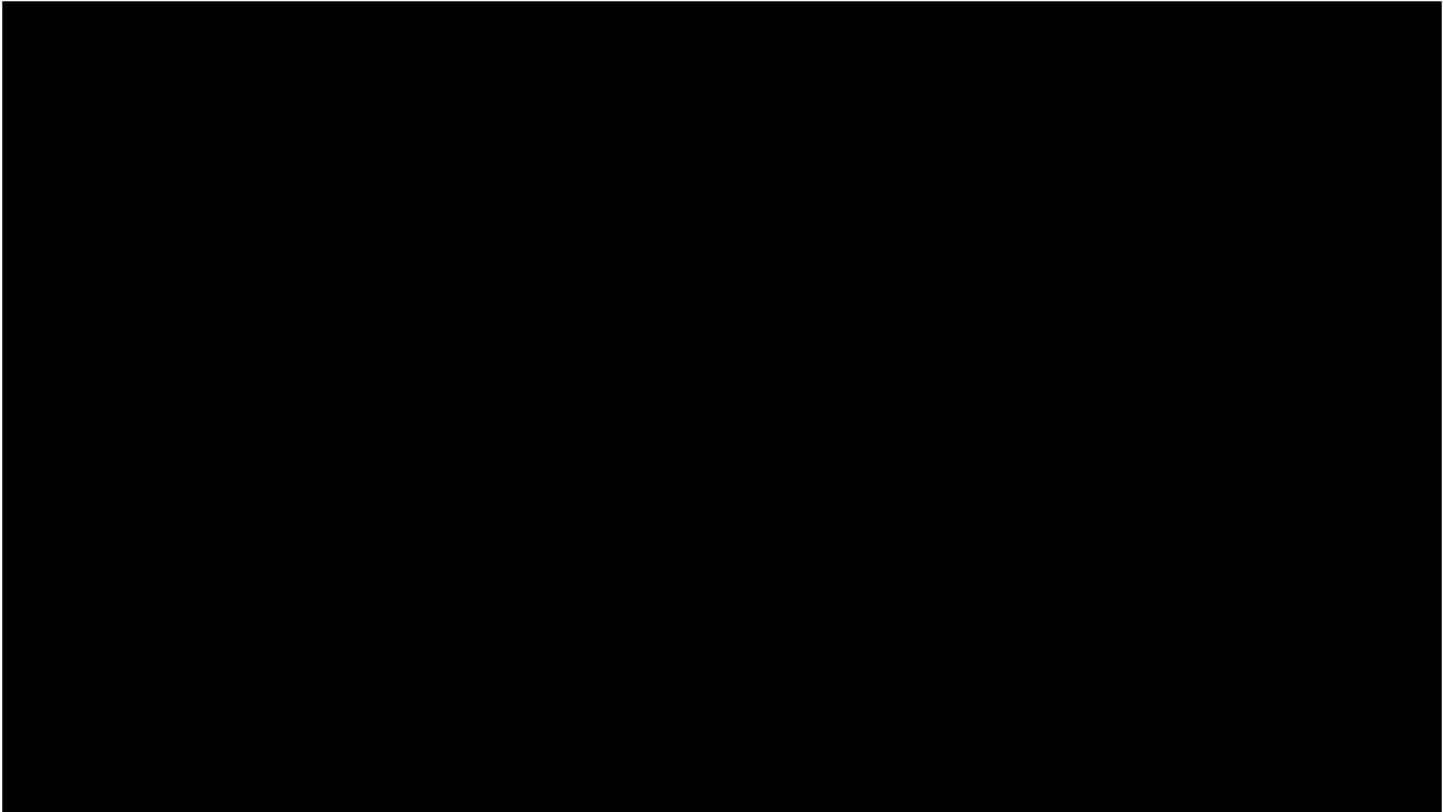


```
if (x == y) {  
    if (y == z) {  
        cout << "Les trois valeurs sont egales." << endl;  
    } else {  
        cout << "Seules les deux premieres valeurs sont egales." << endl;  
    }  
} else {  
    if (x == z) {  
        cout << "Seules la premiere et la troisieme valeurs sont egales." << endl;  
    } else {  
        if (y == z) {  
            cout << "Seules les deux dernieres valeurs sont egales." << endl;  
        } else {  
            cout << "Les trois valeurs sont differentes." << endl;  
        }  
    }  
}
```

valeur un. On commence ici par le premier if, est-ce que x est égal à y? Oui, un est égal à un donc à ce moment là on va rentrer dans ce bloc ici, on va pouvoir donc exécuter notre deuxième if ici, est-ce que y est égal à z? Donc on se pose la question est-ce que les deux valeurs sont égales? Ici oui, un est égal à un donc on va rentrer ici dans cette ligne et le message qui va s'afficher c'est "les trois valeurs sont égales". Puis le programme va continuer à exécuter la suite. Ici il n'y a pas de suite à exécuter. Exécuter la suite du premier bloc ici de plus haut niveau, il n'y a toujours pas de suite et donc ici le programme se termine. Supposons maintenant que l'on ait des valeurs différentes, par exemple que l'on ait les valeurs deux dans z et dans y et qu'on ait la valeur un dans x. Alors la première condition ici va commencer par s'exécuter, est-ce que un est égal à deux? Bien sûr que non et donc la condition ici est fausse ce qui fait qu'on va se brancher ici sur le bloc else correspondant et donc on va rentrer dans l'exécution de ce bloc ici par donc le test de ici une deuxième condition, est-ce que x est égal à z? Est-ce que un est égal à deux? C'est aussi un faux et donc ce deuxième if va lui-même ici se brancher sur un bloc qui va s'exécuter, qui est le bloc qui va s'exécuter ici, lequel bloc commence lui-même par une troisième instruction

notes

résumé



if qui va évaluer sa condition. Est-e que y est égal à z? Deux est égal à deux, oui. Et donc à ce moment-là on va exécuter le bloc qui est contrôlé par la condition vraie et donc ici on va afficher que les deux dernières valeurs sont égales puis on va terminer l'exécution du programme, c'est-à-dire qu'on va terminer l'exécution des instructions ici qui suivaient if. Il n'y a rien, puis ensuite la fin des instructions de ce bloc ici, il n'y a rien non plus, puis enfin les instructions du bloc de plus haut niveau, il n'y a rien non plus dans ce cas particulier, mais voilà comment s'exécuterait le programme de l'exemple. l'exemple.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

4m 13s

